Employee Attrition & Performance

Sonam Bhadauria

6/6/2019

Introduction

We are working on a dataset "HR-Employee-Attrition"" from Kaggle.This is a fictional data set created by IBM data scientists. (Source - https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset)

Our objective here is to understand what factors contributes most to employees turnover and to create a model that can predict if a certain employee will leave the company or not.

Overall, the implementation of this model could allow management to create better decision-making actions.

```r
URL <- tempfile()
download.file("https://raw.githubusercontent.com/sonam-bhadauria/HarvardX-CYO/master/HR-Employee-Attrit

library(data.table)
```

```
## Warning: package 'data.table' was built under R version 3.5.2
```

```r
HR_data <- read.csv(URL,sep = ',',header = TRUE,stringsAsFactors = TRUE,check.names = TRUE)
HR_data <- data.table(HR_data)
```

We can check the data summary as given below

```r
summary(HR_data)
```

```
##       Age        Attrition            BusinessTravel   DailyRate
##  Min.   :18.00   No :1233   Non-Travel       : 150   Min.   : 102.0
##  1st Qu.:30.00   Yes: 237   Travel_Frequently: 277   1st Qu.: 465.0
##  Median :36.00              Travel_Rarely    :1043   Median : 802.0
##  Mean   :36.92                                       Mean   : 802.5
##  3rd Qu.:43.00                                       3rd Qu.:1157.0
##  Max.   :60.00                                       Max.   :1499.0
##
##                   Department  DistanceFromHome   Education
##  Human Resources       : 63   Min.   : 1.000   Min.   :1.000
##  Research & Development:961   1st Qu.: 2.000   1st Qu.:2.000
##  Sales                 :446   Median : 7.000   Median :3.000
##                               Mean   : 9.193   Mean   :2.913
##                               3rd Qu.:14.000   3rd Qu.:4.000
##                               Max.   :29.000   Max.   :5.000
##
##           EducationField EmployeeCount EmployeeNumber
##  Human Resources : 27    Min.   :1     Min.   :   1.0
##  Life Sciences   :606    1st Qu.:1     1st Qu.: 491.2
##  Marketing       :159    Median :1     Median :1020.5
##  Medical         :464    Mean   :1     Mean   :1024.9
##  Other           : 82    3rd Qu.:1     3rd Qu.:1555.8
##  Technical Degree:132    Max.   :1     Max.   :2068.0
##
##  EnvironmentSatisfaction    Gender      HourlyRate      JobInvolvement
##  Min.   :1.000           Female:588   Min.   : 30.00   Min.   :1.00
```

```
##  1st Qu.:2.000          Male  :882   1st Qu.: 48.00   1st Qu.:2.00
##  Median :3.000                       Median : 66.00   Median :3.00
##  Mean   :2.722                       Mean   : 65.89   Mean   :2.73
##  3rd Qu.:4.000                       3rd Qu.: 83.75   3rd Qu.:3.00
##  Max.   :4.000                       Max.   :100.00   Max.   :4.00
##
##     JobLevel                          JobRole    JobSatisfaction
##  Min.   :1.000   Sales Executive         :326   Min.   :1.000
##  1st Qu.:1.000   Research Scientist      :292   1st Qu.:2.000
##  Median :2.000   Laboratory Technician   :259   Median :3.000
##  Mean   :2.064   Manufacturing Director  :145   Mean   :2.729
##  3rd Qu.:3.000   Healthcare Representative:131   3rd Qu.:4.000
##  Max.   :5.000   Manager                 :102   Max.   :4.000
##                  (Other)                 :215
##   MaritalStatus MonthlyIncome    MonthlyRate    NumCompaniesWorked
##  Divorced:327   Min.   : 1009   Min.   : 2094   Min.   :0.000
##  Married :673   1st Qu.: 2911   1st Qu.: 8047   1st Qu.:1.000
##  Single  :470   Median : 4919   Median :14236   Median :2.000
##                 Mean   : 6503   Mean   :14313   Mean   :2.693
##                 3rd Qu.: 8379   3rd Qu.:20462   3rd Qu.:4.000
##                 Max.   :19999   Max.   :26999   Max.   :9.000
##
##  Over18   OverTime   PercentSalaryHike PerformanceRating
##  Y:1470   No :1054   Min.   :11.00     Min.   :3.000
##           Yes: 416   1st Qu.:12.00     1st Qu.:3.000
##                      Median :14.00     Median :3.000
##                      Mean   :15.21     Mean   :3.154
##                      3rd Qu.:18.00     3rd Qu.:3.000
##                      Max.   :25.00     Max.   :4.000
##
##  RelationshipSatisfaction StandardHours StockOptionLevel TotalWorkingYears
##  Min.   :1.000            Min.   :80    Min.   :0.0000   Min.   : 0.00
##  1st Qu.:2.000            1st Qu.:80    1st Qu.:0.0000   1st Qu.: 6.00
##  Median :3.000            Median :80    Median :1.0000   Median :10.00
##  Mean   :2.712            Mean   :80    Mean   :0.7939   Mean   :11.28
##  3rd Qu.:4.000            3rd Qu.:80    3rd Qu.:1.0000   3rd Qu.:15.00
##  Max.   :4.000            Max.   :80    Max.   :3.0000   Max.   :40.00
##
##  TrainingTimesLastYear WorkLifeBalance YearsAtCompany   YearsInCurrentRole
##  Min.   :0.000         Min.   :1.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:2.000         1st Qu.:2.000   1st Qu.: 3.000   1st Qu.: 2.000
##  Median :3.000         Median :3.000   Median : 5.000   Median : 3.000
##  Mean   :2.799         Mean   :2.761   Mean   : 7.008   Mean   : 4.229
##  3rd Qu.:3.000         3rd Qu.:3.000   3rd Qu.: 9.000   3rd Qu.: 7.000
##  Max.   :6.000         Max.   :4.000   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   : 2.188          Mean   : 4.123
##  3rd Qu.: 3.000          3rd Qu.: 7.000
##  Max.   :15.000          Max.   :17.000
##
```

Observations and Data preparation

1) Data has 1,470 rows with 35 columns(variables)

2) Class Label is Attrition with 1232 'NO' and 237 'Yes' that shows the unbalance class label. we have to pay attention to the unbalance class algorithm problems!

3) Some of variables are related to the years of working wich can be a good candidate for feature generation. Some of variable are related to personal issues like WorkLifeBalance, RelationshipSatisfaction, JobSatisfaction,EnvironmentSatisfaction etc.

4) There are some variables that are related to the income like MonthlyIncome, PercentSalaryHike, etc.

5) More and more, we have to envestigate that, how the company objective factors influence in attition employees, and what kind of working enviroment most will cause employees attrition.

6) We checked our data for Missing values. Fortunately, we dont have any missing values as shown below

```r
apply(is.na(HR_data), 2, sum)
```

```
##                      Age                Attrition           BusinessTravel
##                        0                        0                        0
##                DailyRate               Department         DistanceFromHome
##                        0                        0                        0
##                Education           EducationField            EmployeeCount
##                        0                        0                        0
##           EmployeeNumber  EnvironmentSatisfaction                   Gender
##                        0                        0                        0
##               HourlyRate           JobInvolvement                 JobLevel
##                        0                        0                        0
##                  JobRole          JobSatisfaction            MaritalStatus
##                        0                        0                        0
##            MonthlyIncome              MonthlyRate       NumCompaniesWorked
##                        0                        0                        0
##                   Over18                 OverTime        PercentSalaryHike
##                        0                        0                        0
##        PerformanceRating RelationshipSatisfaction            StandardHours
##                        0                        0                        0
##         StockOptionLevel        TotalWorkingYears    TrainingTimesLastYear
##                        0                        0                        0
##          WorkLifeBalance           YearsAtCompany       YearsInCurrentRole
##                        0                        0                        0
##  YearsSinceLastPromotion     YearsWithCurrManager
##                        0                        0
```

7) Also, We have removed non value attributes.These variables can not play significant role because they are same for all records.

8) EmployeeNumber is a variable for identifying the specific employee.If we have more information about employee and the structure of the employee number, then we can extract some new features. But now it is not possible and that is why we have removed it from our data set.

9) Employee Count is equal 1 for all observation which can not generate useful value for this sample data. Maybe for the other sample of data will be with different values that should be considered for builiding the model in the future for other sets of data. In this analysis, we will remove it.

10) Over 18 is equal to 'Y', which means employee is not less than 18 years old. this attribute should be considered for the future, maybe by changing the ruls of emploement, young people under 18 can also working in companies. Here, according to the data set, we will remove it.

11) Standard Hours is equal 80 for all observation. the decision for this attribute is same to Over18 and Employee Count. BusinessTravel, Department, EducationField, Gender, jobRole, MaritalStatus and OverTime are categorical data and other variabels are continues.

```
HR_data$EmployeeNumber <- NULL
HR_data$EmployeeCount <- NULL
HR_data$Over18 <- NULL
HR_data$StandardHours <- NULL
```

12) After removing Non value data attributes, now our Dataset has 1470 Rows and 31 Columns. Also, we checked for any duplicate records

```
sum(is.na(duplicated(HR_data)))
```

```
## [1] 0
```

13) There are some attributes that are categorical, but some are integer. We changed them into categorical. Also, we do not need any dummy variable creation, where some machine learning algorithms like RF, XGBoost etc. can use categorical variables.

14) For other algorithms like NN we have to change categorical variable more than two level to dummy variable Variable with twol level (Binary) can be change to number very easy.

```
HR_data$Education <- as.factor(HR_data$Education)
HR_data$EnvironmentSatisfaction <- as.factor(HR_data$EnvironmentSatisfaction)
HR_data$JobInvolvement <- as.factor(HR_data$JobInvolvement)
HR_data$JobLevel <- as.factor(HR_data$JobLevel)
HR_data$JobSatisfaction <- as.factor(HR_data$JobSatisfaction)
HR_data$PerformanceRating <- as.factor(HR_data$PerformanceRating)
HR_data$RelationshipSatisfaction <- as.factor(HR_data$RelationshipSatisfaction)
HR_data$StockOptionLevel <- as.factor(HR_data$StockOptionLevel)
HR_data$WorkLifeBalance <- as.factor(HR_data$WorkLifeBalance)
```

Data Visualization

We are done with data preparation, now we are going ahead & analyze the variables through visualization. We are going to check what all variables really contributes in the attrition. So, going forward we will focus on those variables only while building our model.

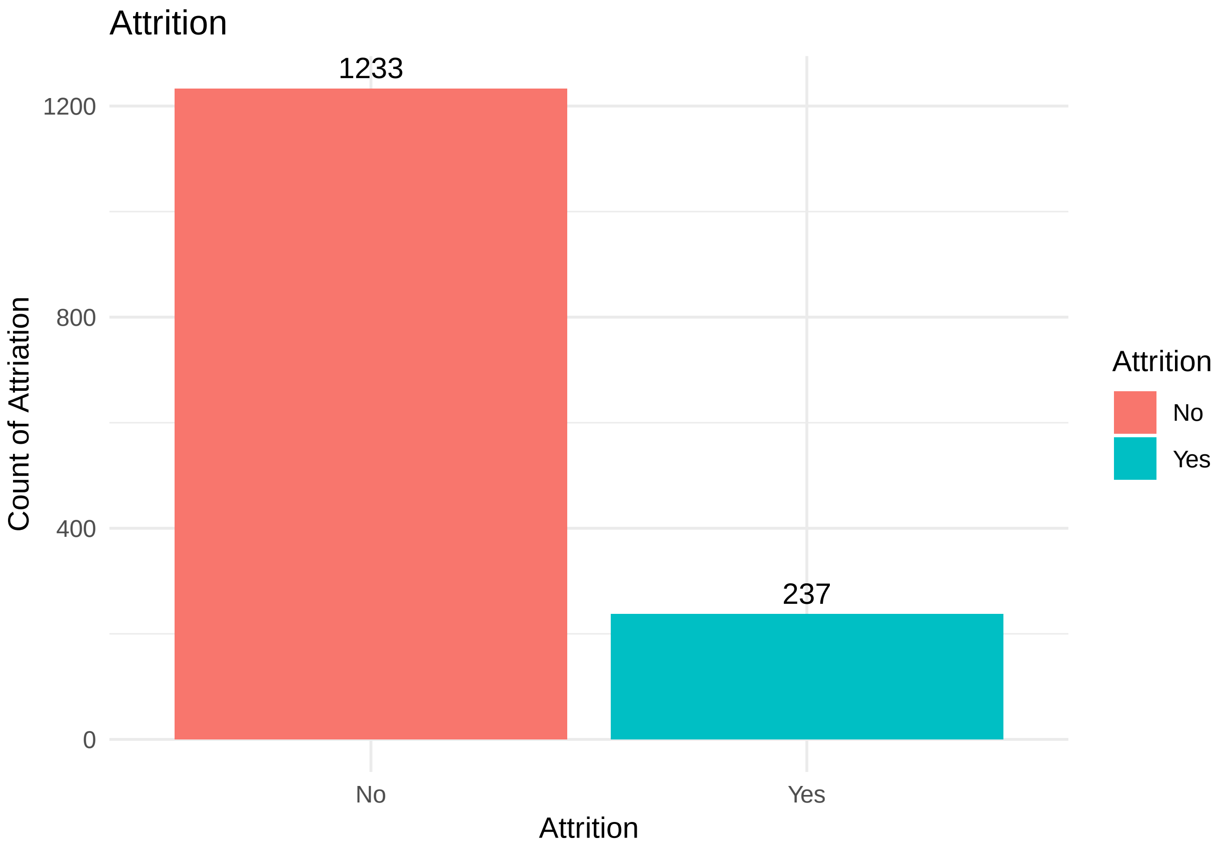
```
summary(HR_data)
```

```
##       Age        Attrition            BusinessTravel   DailyRate
##  Min.   :18.00   No :1233   Non-Travel       : 150   Min.   : 102.0
##  1st Qu.:30.00   Yes: 237   Travel_Frequently: 277   1st Qu.: 465.0
##  Median :36.00              Travel_Rarely    :1043   Median : 802.0
##  Mean   :36.92                                       Mean   : 802.5
##  3rd Qu.:43.00                                       3rd Qu.:1157.0
##  Max.   :60.00                                       Max.   :1499.0
##
##                   Department  DistanceFromHome Education
##  Human Resources       : 63   Min.   : 1.000   1:170
##  Research & Development:961   1st Qu.: 2.000   2:282
##  Sales                 :446   Median : 7.000   3:572
##                               Mean   : 9.193   4:398
##                               3rd Qu.:14.000   5: 48
##                               Max.   :29.000
##
##           EducationField EnvironmentSatisfaction    Gender
```

```
##  Human Resources : 27    1:284                  Female:588
##  Life Sciences   :606    2:287                  Male  :882
##  Marketing       :159    3:453
##  Medical         :464    4:446
##  Other           : 82
##  Technical Degree:132
##
##    HourlyRate     JobInvolvement JobLevel                      JobRole
##  Min.   : 30.00   1: 83          1:543    Sales Executive         :326
##  1st Qu.: 48.00   2:375          2:534    Research Scientist      :292
##  Median : 66.00   3:868          3:218    Laboratory Technician   :259
##  Mean   : 65.89   4:144          4:106    Manufacturing Director  :145
##  3rd Qu.: 83.75                  5: 69    Healthcare Representative:131
##  Max.   :100.00                           Manager                 :102
##                                           (Other)                 :215
##  JobSatisfaction  MaritalStatus MonthlyIncome    MonthlyRate
##  1:289           Divorced:327   Min.   : 1009   Min.   : 2094
##  2:280           Married :673   1st Qu.: 2911   1st Qu.: 8047
##  3:442           Single  :470   Median : 4919   Median :14236
##  4:459                          Mean   : 6503   Mean   :14313
##                                 3rd Qu.: 8379   3rd Qu.:20462
##                                 Max.   :19999   Max.   :26999
##
##  NumCompaniesWorked OverTime   PercentSalaryHike PerformanceRating
##  Min.   :0.000      No :1054   Min.   :11.00     3:1244
##  1st Qu.:1.000      Yes: 416   1st Qu.:12.00     4: 226
##  Median :2.000                 Median :14.00
##  Mean   :2.693                 Mean   :15.21
##  3rd Qu.:4.000                 3rd Qu.:18.00
##  Max.   :9.000                 Max.   :25.00
##
##  RelationshipSatisfaction StockOptionLevel TotalWorkingYears
##  1:276                    0:631            Min.   : 0.00
##  2:303                    1:596            1st Qu.: 6.00
##  3:459                    2:158            Median :10.00
##  4:432                    3: 85            Mean   :11.28
##                                            3rd Qu.:15.00
##                                            Max.   :40.00
##
##  TrainingTimesLastYear WorkLifeBalance YearsAtCompany   YearsInCurrentRole
##  Min.   :0.000         1: 80           Min.   : 0.000   Min.   : 0.000
##  1st Qu.:2.000         2:344           1st Qu.: 3.000   1st Qu.: 2.000
##  Median :3.000         3:893           Median : 5.000   Median : 3.000
##  Mean   :2.799         4:153           Mean   : 7.008   Mean   : 4.229
##  3rd Qu.:3.000                         3rd Qu.: 9.000   3rd Qu.: 7.000
##  Max.   :6.000                         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   : 2.188          Mean   : 4.123
##  3rd Qu.: 3.000          3rd Qu.: 7.000
##  Max.   :15.000          Max.   :17.000
```

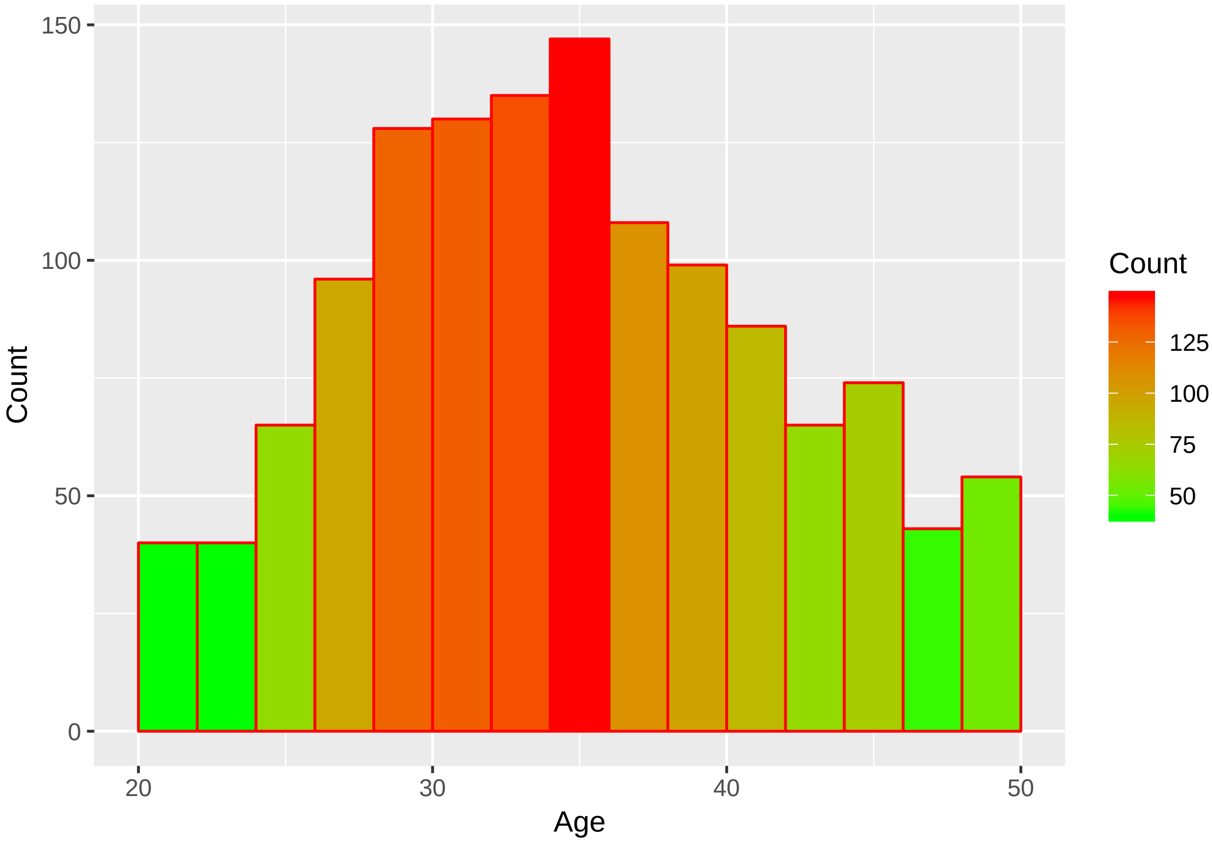
##

Visualization of Attrition



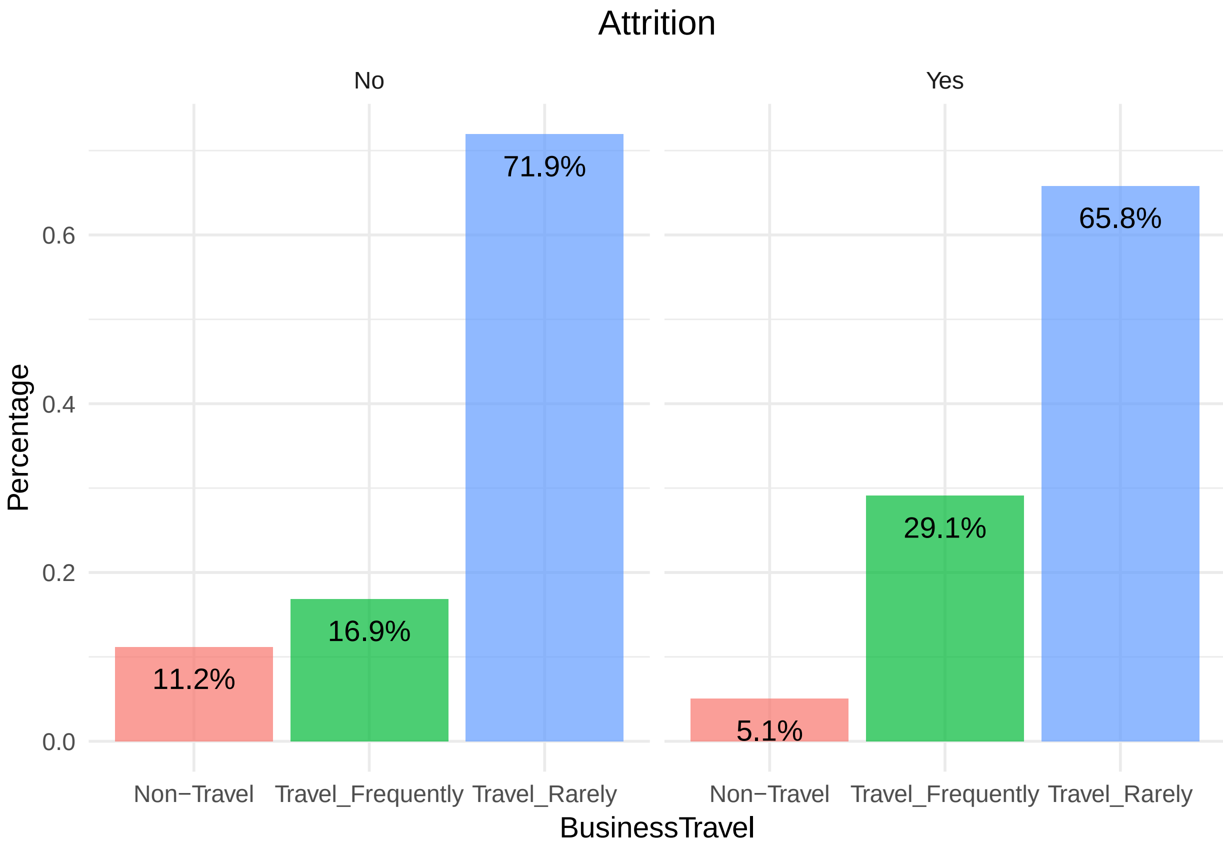
As we can see here, 237/1233=0.19 % of the data label shows the "Yes" in Attrition. This problem should be fixed during the process because unbalanced dataset will bias the prediction model towards the more common class (here is 'NO'). There are different approaches for dealing with unbalanced data in machine learning like using more data (here is not possible), Resampling , changing the machine performance metric, using various algorithms etc.

Visualizing age distribution in a histogram



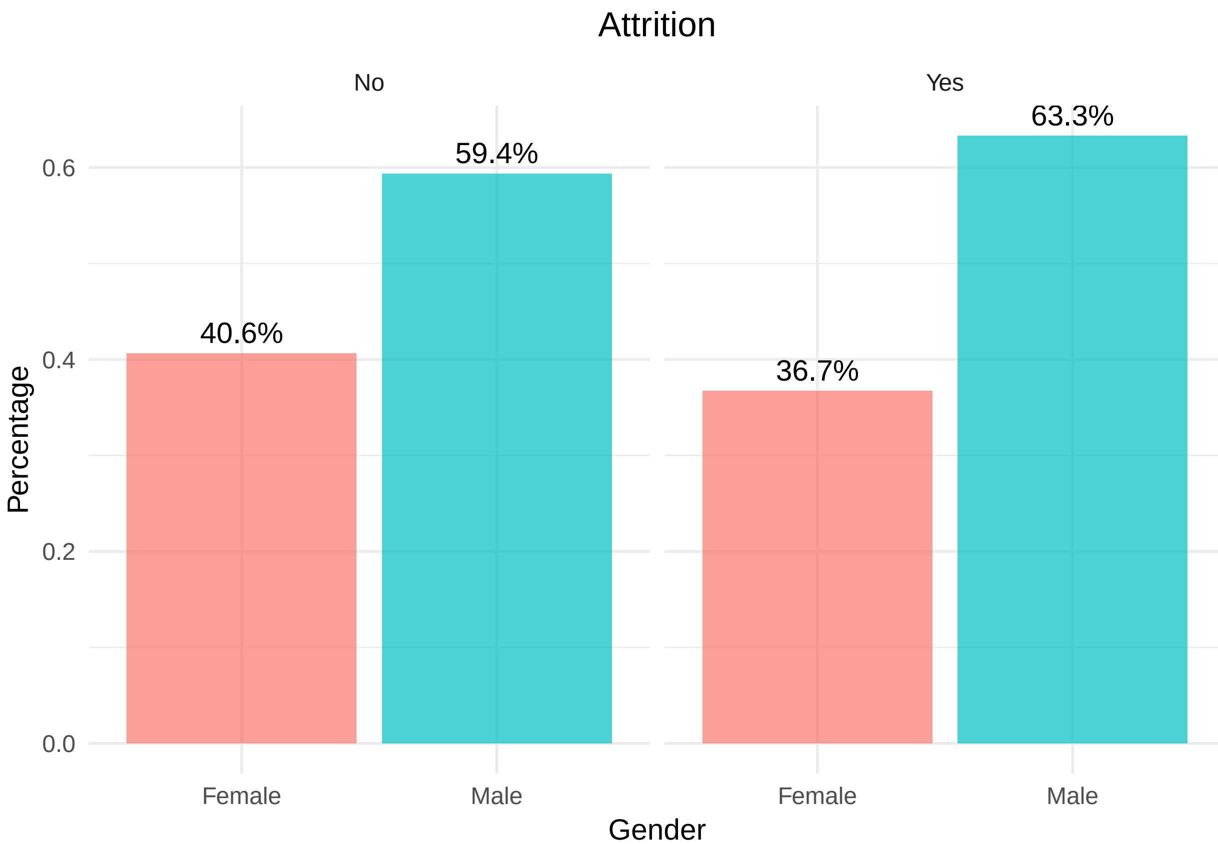
As we can see above, the majority of employees are between 28-36 years. 34-36 years old are very popolar.

Attrition based on business travel



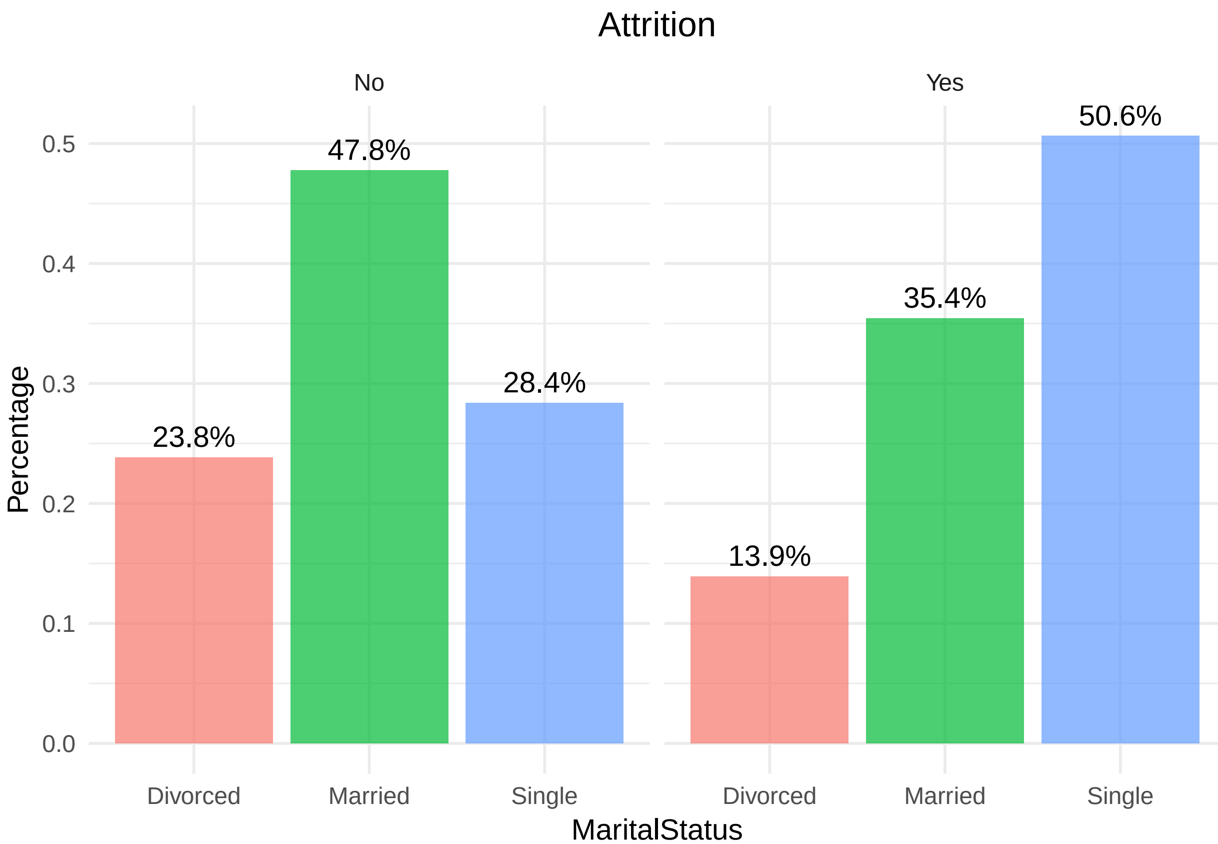
Here is the distribution of the data according to the Business Tralvel situation. More than 70% of employees travel rarely where just 10 % of them has no travel. People who travel frequently tend to have more attrition

Attrition by Gender



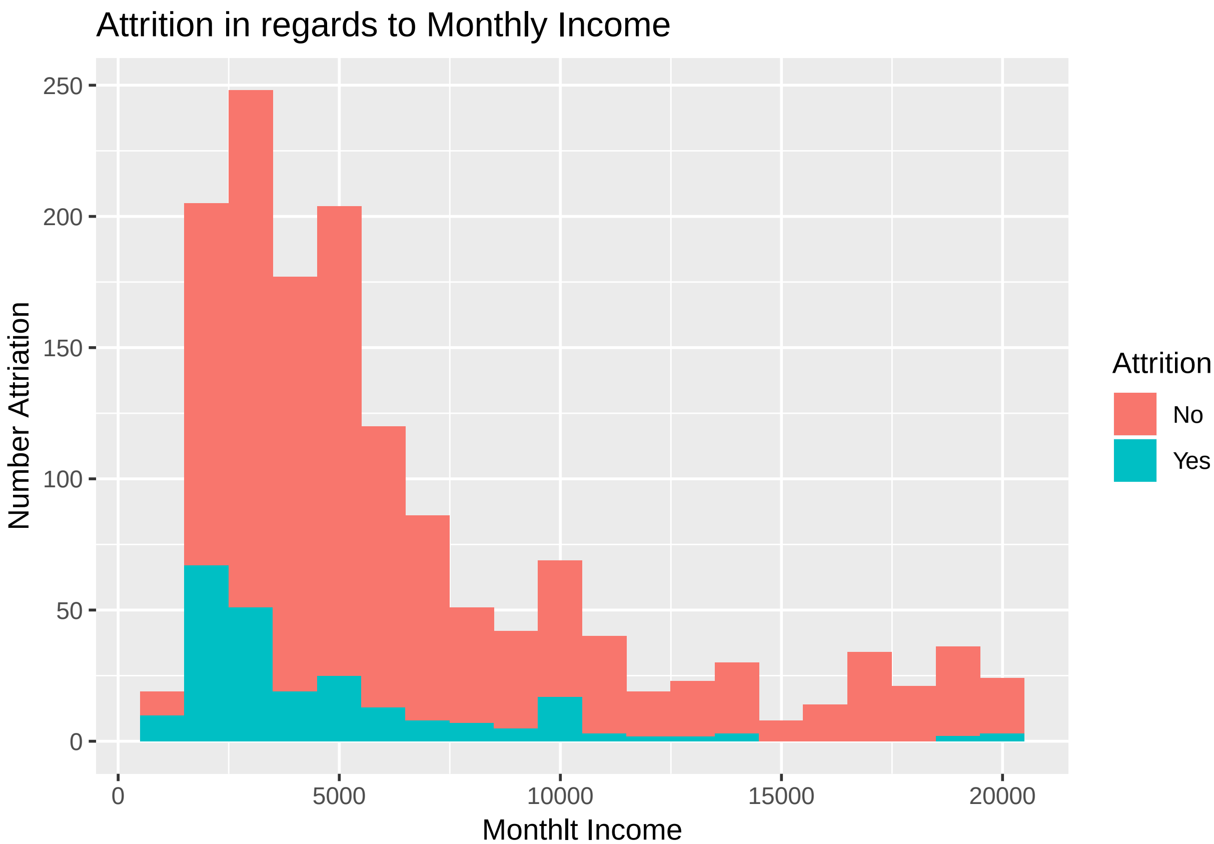
There is no discernible observation. We can not really predict attrition based on this variable.

Attrition by marital status



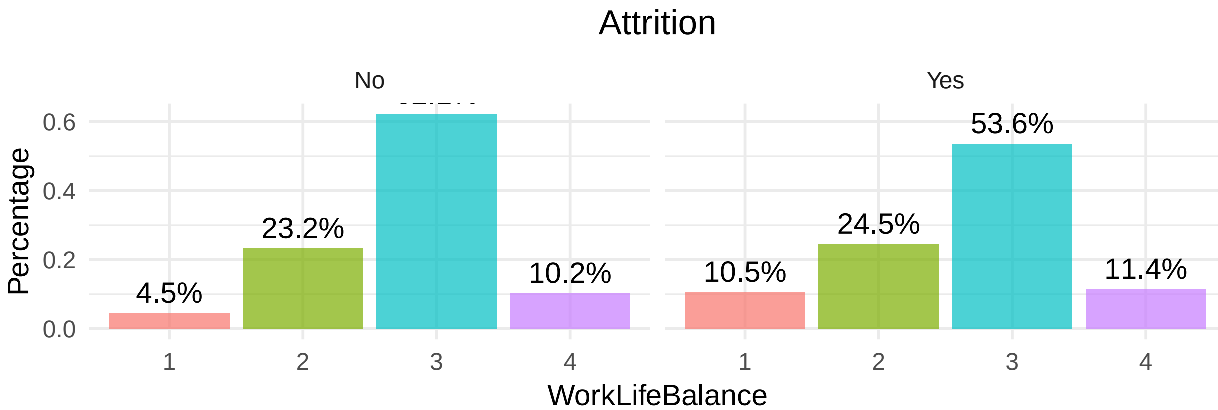
We can infer through above plot that the people who are single tend to have highest attrition and people who are married tend to have the least chances.

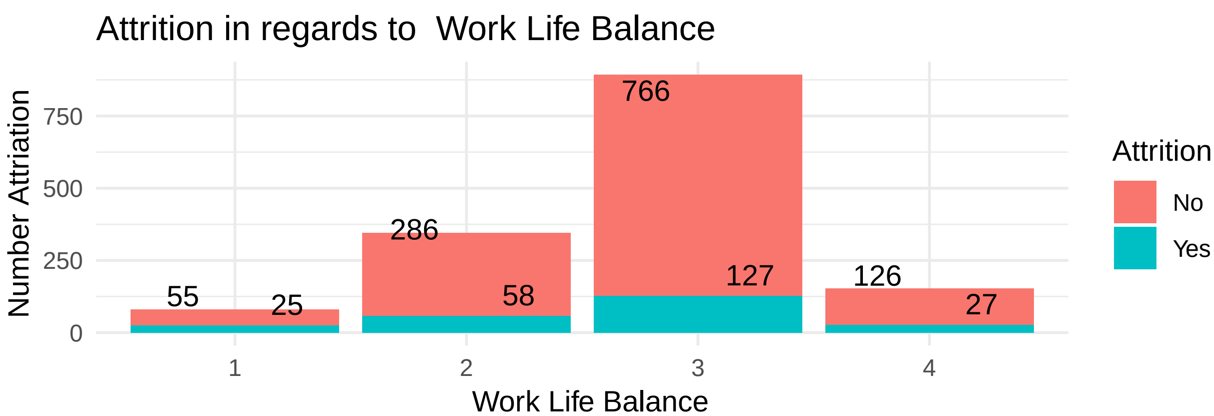
Attrition by Monthly salary



We can notice that the maximum attrition is with the people under 5000.

Attrition based on Work Life Balance





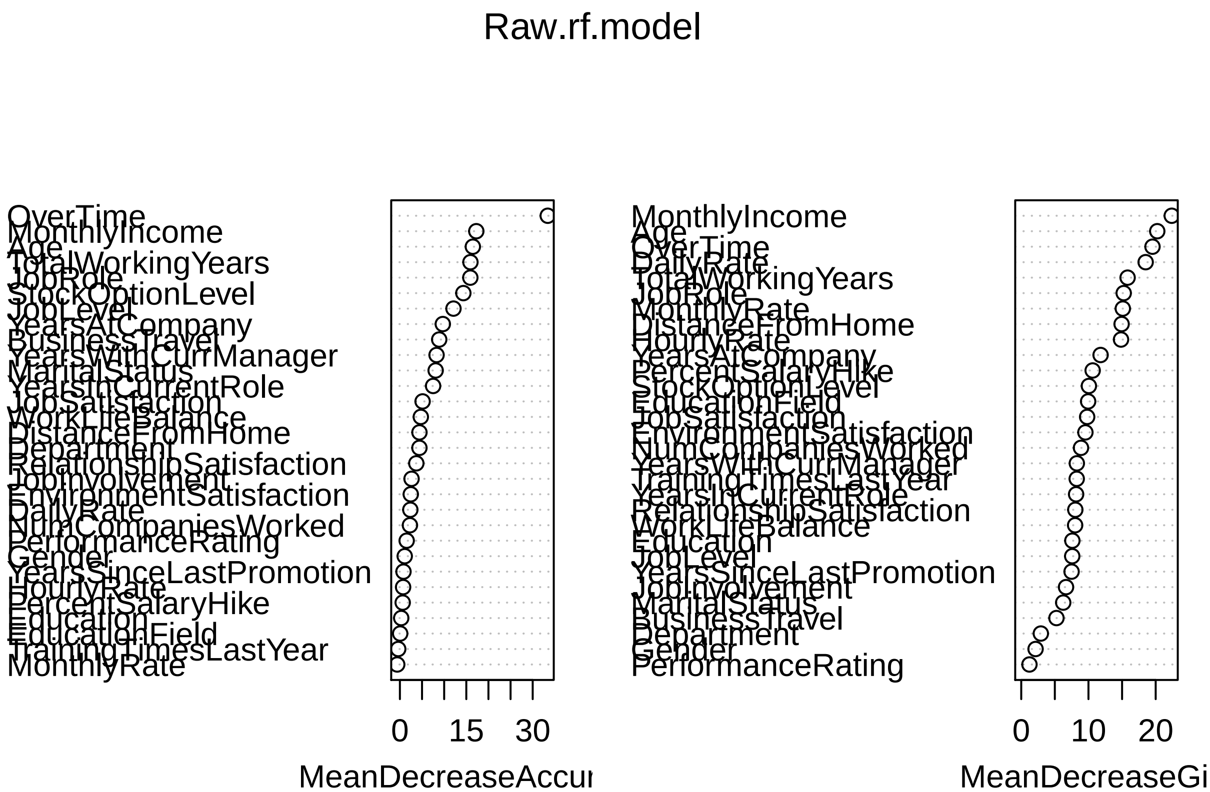
WorkLifeBalance (categorical) - 1 'Bad' , 2 'Good' , 3 'Better' , 4 'Best'

Analysis Process

1 Using Random Forest

Splitting data into train & test sets

Building the model



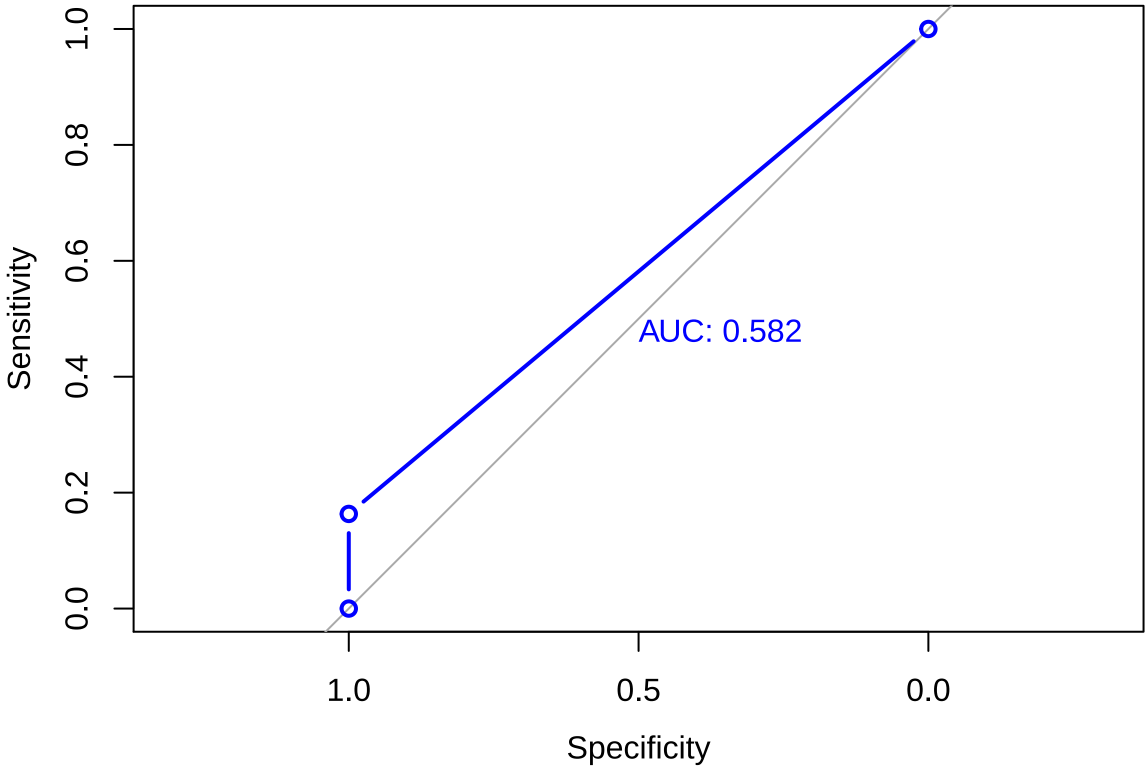
As we can see here OverTime, Monthly Income , Age, Total Working Years and Job Role are the top 5 contributors.

Lets check the accuracy below

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  245   0
##        Yes  41   8
##
##                Accuracy : 0.8605
##                  95% CI : (0.8156, 0.898)
##     No Information Rate : 0.9728
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2454
##
##  Mcnemar's Test P-Value : 4.185e-10
##
##             Sensitivity : 0.8566
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.1633
##              Prevalence : 0.9728
##          Detection Rate : 0.8333
##    Detection Prevalence : 0.8333
##       Balanced Accuracy : 0.9283
##
```

```
##        'Positive' Class : No
##
```

AUC (Area under Curve)



Our results shows good accuracy however the AUC is not very good.

Lets try & find methods to improve the accuracy & AUC.

Feature Engineering

Now we will do some data wrapping here to make our results better:

1) Making age group 18-24 as Young, 25-54 as Middle and > 54 as Senior

```
##
## Middle Senior  Young
##   1304     69     97
```

As we can see here,the majority of employees are in Middle age group

2) Creating a column "Total Satisfaction" which encompasses: Environment Satisfacton, Job Involvement, Job Satisfaction, Relationship and WorkLife Balance

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    6.00   12.00   14.00   13.65   15.00   20.00
```

3) Total years of education

There are five Education level. From high School to PhD (HighSchool=10 years, College=12 years, Bachelor=16 years, Master=18 years, PhD= 22 years)

```
##
##  10  12  16  18  22
## 170 282 572 398  48
```

We can notice that the majority of employees are 16 years education (i.e. Bachelors)

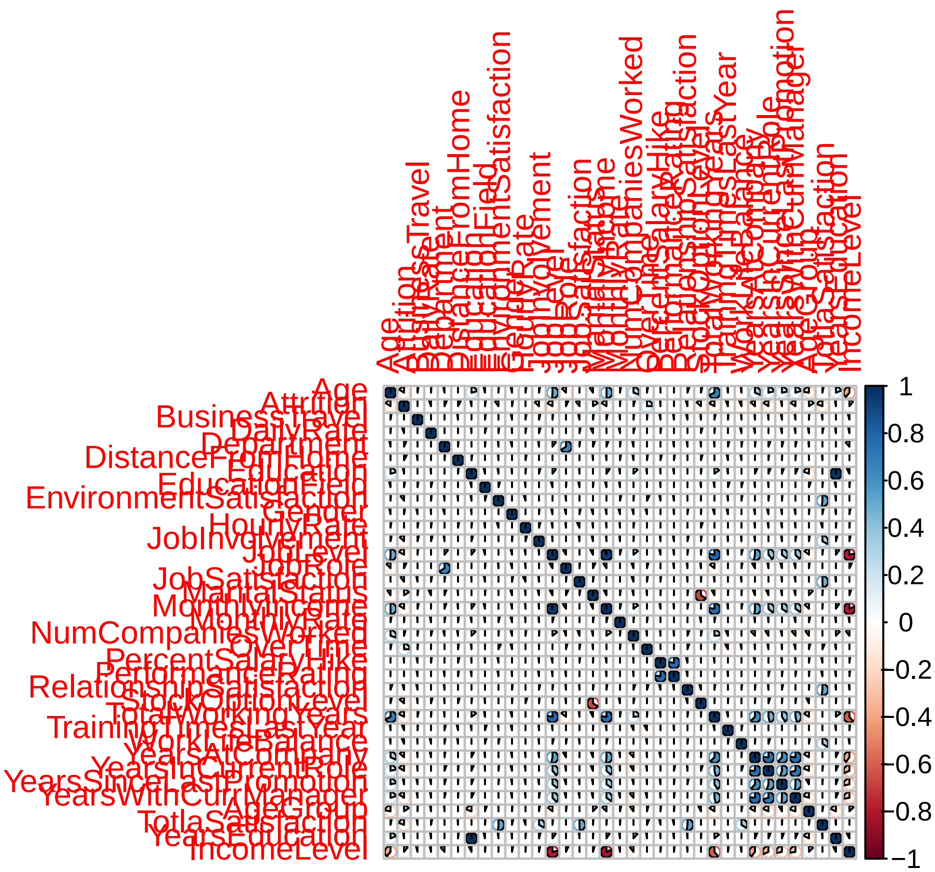
4) Categorising monthly income into low-high groups, based on average income.

```
##
## High  Low
##  493  977
```

Correlation Matrix

Let us see the Correlation Matrix of Data in order to find out the correlation between variables.

```
## corrplot 0.84 loaded
```



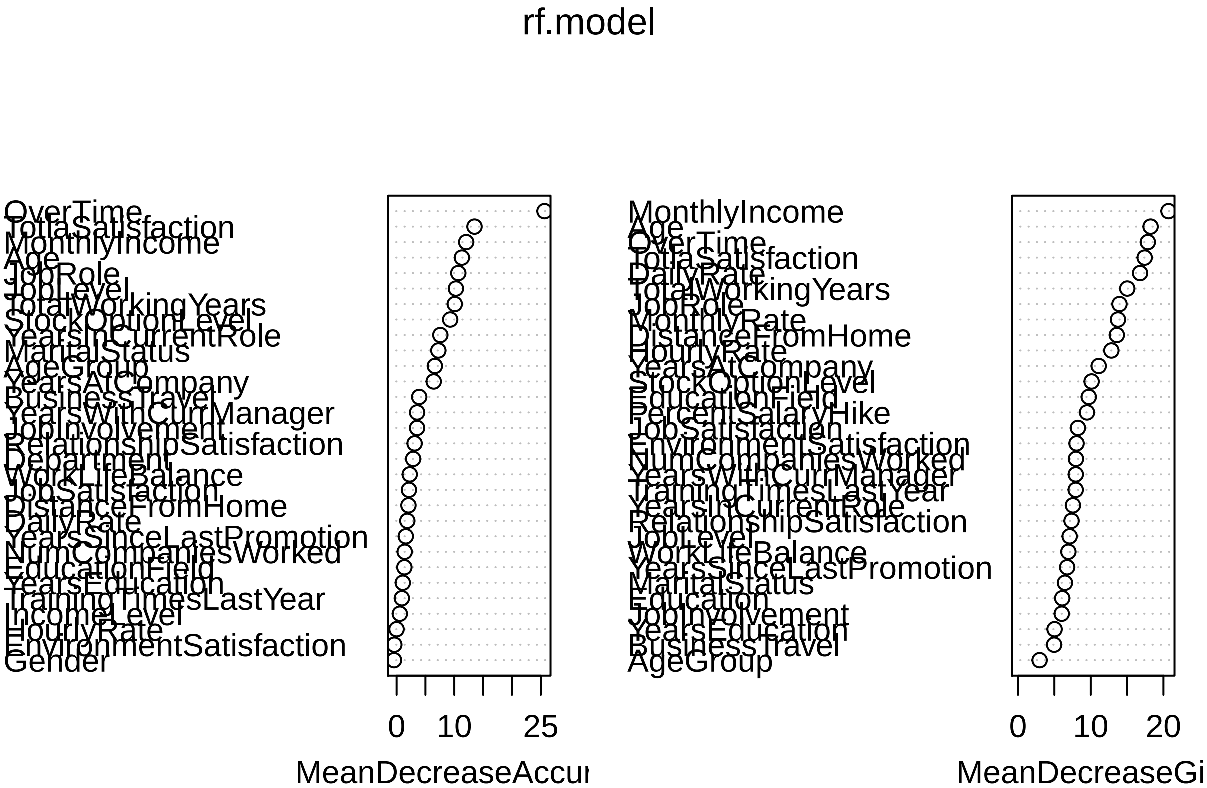
We can see some of the variables are highly correlated. For example -

1) Joblevel and monthly income
2) Education and YearsEducation

They might cause multicollinearity problem in our data set. we have to decide to remove one of them from any group Now we will try again our dataset with new attributes using Random Forest again.

New Random Forest

We are using random forest using the data with updated attributes now.



Here we can see OverTime, TotalSatisfaction , MonthlyIncome, Age and JobRole are top five variables.

Confusion Matrix

Lets check the accuracy using confusion Matrix

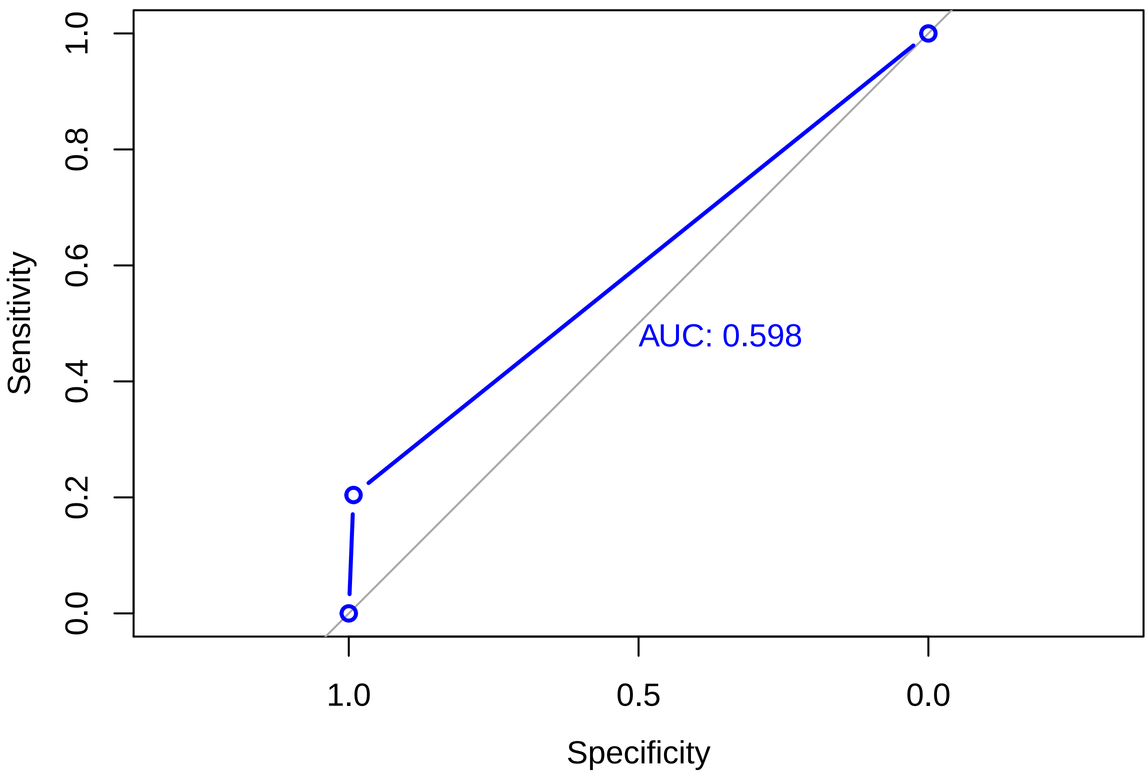
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  243   2
##        Yes  39  10
##
##                Accuracy : 0.8605
##                  95% CI : (0.8156, 0.898)
##     No Information Rate : 0.9592
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2807
##
##  Mcnemar's Test P-Value : 1.885e-08
##
##             Sensitivity : 0.8617
##             Specificity : 0.8333
##          Pos Pred Value : 0.9918
##          Neg Pred Value : 0.2041
##              Prevalence : 0.9592
##          Detection Rate : 0.8265
##    Detection Prevalence : 0.8333
```

```
##       Balanced Accuracy : 0.8475
##
##        'Positive' Class : No
##
```

AUC

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```



We can notice that our AUC has improved over the last RF with Raw Data

Using other Algorithms

We will use some other algorithms also to build a better model

Support Vector Machine

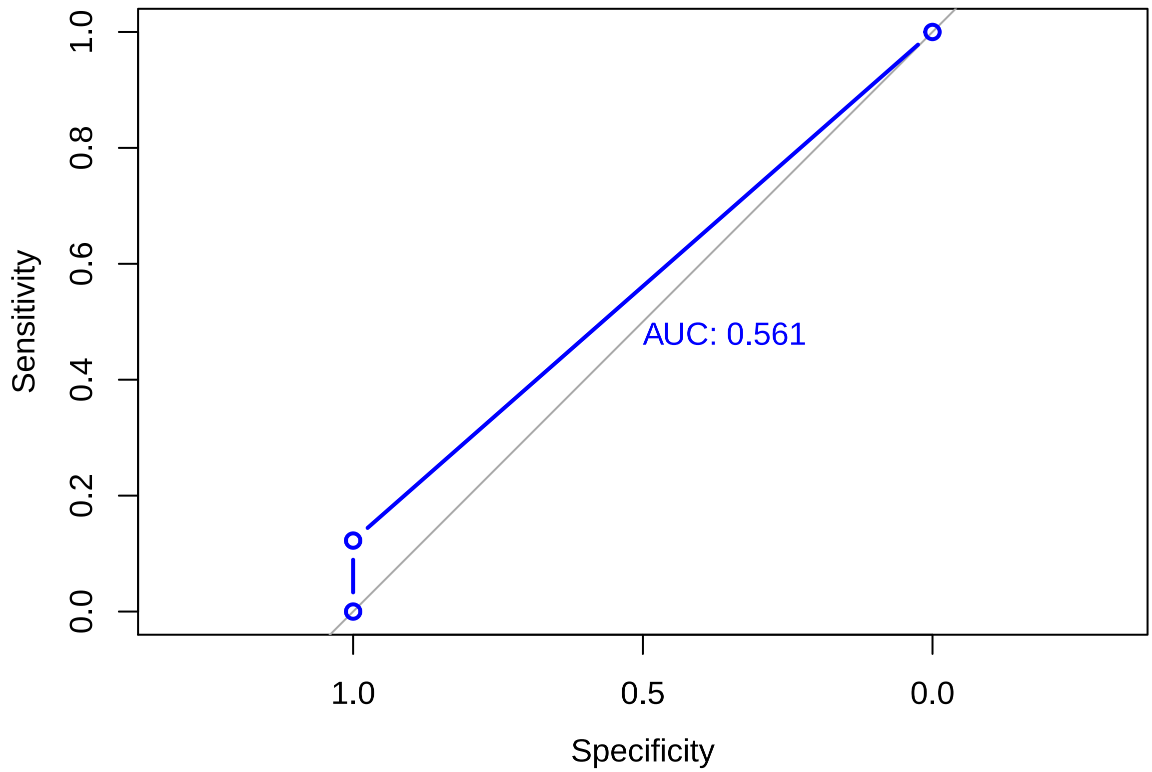
```
library(e1071)
svmData <- HR_data
set.seed(123)
indexes = sample(1:nrow(svmData), size=0.8*nrow(svmData))
SVMtrain.Data <- svmData[indexes,]
SVMtest.Data <- svmData[-indexes,]
tuned <- tune(svm,factor(Attrition)~.,data = SVMtrain.Data)
svm.model <- svm(SVMtrain.Data$Attrition~., data=SVMtrain.Data
                 ,type="C-classification", gamma=tuned$best.model$gamma
                 ,cost=tuned$best.model$cost
                 ,kernel="radial")
```

```r
svm.prd <- predict(svm.model,newdata=SVMtest.Data)
confusionMatrix(svm.prd,SVMtest.Data$Attrition)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  245  43
##        Yes   0   6
##
##                Accuracy : 0.8537
##                  95% CI : (0.8081, 0.8921)
##     No Information Rate : 0.8333
##     P-Value [Acc > NIR] : 0.1959
##
##                   Kappa : 0.1887
##
##  Mcnemar's Test P-Value : 1.504e-10
##
##             Sensitivity : 1.0000
##             Specificity : 0.1224
##          Pos Pred Value : 0.8507
##          Neg Pred Value : 1.0000
##              Prevalence : 0.8333
##          Detection Rate : 0.8333
##    Detection Prevalence : 0.9796
##       Balanced Accuracy : 0.5612
##
##        'Positive' Class : No
##
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```



We can observe that our AUC(0.561) and Accuracy (0.8537) compared to RF is bad. There is no False Negative and a lot of false positives.

Extreme Gradient Boost

To proceed with XG boost, we need to first tune the hyper parameters as below

```
#Tuning XBGTree using Caret Package
#Hyperparameters to tune:
#nrounds: Number of trees, default: 100
#max_depth: Maximum tree depth, default: 6
#eta: Learning rate, default: 0.3
#  gamma: Used for tuning of Regularization, default: 0
#  colsample_bytree: Column sampling, default: 1
#  min_child_weight: Minimum leaf weight, default: 1
#  subsample: Row sampling, default: 1
#  We'll break down the tuning of these into five sections:
#
#   Fixing learning rate eta and number of iterations nrounds
#  M aximum depth max_depth and child weight min_child_weight
#  Setting column colsample_bytree and row sampling subsample
#  Experimenting with different gamma values
#  Reducing the learning rate eta

# set seed
library(xgboost)
set.seed(123)
xgbData <- HR_data
indexes <- sample(1:nrow(xgbData), size=0.8*nrow(xgbData))
XGBtrain.Data <- xgbData[indexes,]
```

```r
XGBtest.Data <- xgbData[-indexes,]
# note to start nrounds from 200, as smaller learning rates result in errors so
# big with lower starting points that they'll mess the scales

formula = Attrition~.
nrounds <- 1000

tune_grid <- expand.grid(
  nrounds = seq(from = 200, to = nrounds, by = 50),
  eta = c(0.025, 0.05, 0.1, 0.3),
  max_depth = c(2, 3, 4, 5, 6,8,10,15,20,25,30,35,40,50),
  gamma = 0,
  colsample_bytree = 1,
  min_child_weight = 1,
  subsample = 1
)

tune_control <- caret::trainControl(
  method = "cv", # cross-validation
  number = 3, # with n folds
  classProbs = TRUE
)

xgb_tune <- caret::train(
  formula,
  data = XGBtrain.Data,
  trControl = tune_control,
  tuneGrid = tune_grid,
  method = "xgbTree"
)
predictions<-predict(xgb_tune,XGBtest.Data)
confusionMatrix(predictions,XGBtest.Data$Attrition) #0.8639

#output of best Tune is nrounds = 500 eta = 0.05

# helper function for the plots
tuneplot <- function(x, probs = .90) {
  ggplot(x) +
    coord_cartesian(ylim = c(quantile(x$results$RMSE, probs = probs), min(x$results$RMSE))) +
    theme_bw()
}

tuneplot(xgb_tune)

tune_grid2 <- expand.grid(
  nrounds = seq(from = 50, to = nrounds, by = 50), #700
  eta = xgb_tune$bestTune$eta,
  # max_depth = ifelse(xgb_tune$bestTune$max_depth == 2, #2
  #                    c(xgb_tune$bestTune$max_depth:30),
  #                    xgb_tune$bestTune$max_depth - 1:xgb_tune$bestTune$max_depth + 1),

  max_depth = c(1,2,3,5,10,15,20,25,30,35), #15
  gamma = 0,
```

```
  colsample_bytree = 1,
  min_child_weight = c(1, 2, 3), #3
  subsample = 1
)

xgb_tune2 <- caret::train(
  formula,
  data = XGBtrain.Data,
  trControl = tune_control,
  tuneGrid = tune_grid2,
  method = "xgbTree",
  verbose = TRUE
)

ggplot(xgb_tune2)
predictions2<-predict(xgb_tune2,XGBtest.Data)
confusionMatrix(predictions2,XGBtest.Data$Attrition) #0.8571

tune_grid3 <- expand.grid(
  nrounds = seq(from = 50, to = nrounds, by = 50), #150
  eta = xgb_tune$bestTune$eta,
  max_depth = xgb_tune2$bestTune$max_depth,
  gamma = 0,
  colsample_bytree = c(0.4, 0.6, 0.8, 1.0), #0.8
  min_child_weight = xgb_tune2$bestTune$min_child_weight,
  subsample = c(0.5, 0.75, 1.0) #0.75
)

xgb_tune3 <- caret::train(
  formula,
  data = XGBtrain.Data,
  trControl = tune_control,
  tuneGrid = tune_grid3,
  method = "xgbTree",
  verbose = TRUE
)

ggplot(xgb_tune3)
predictions3<-predict(xgb_tune3,XGBtest.Data)
confusionMatrix(predictions3,XGBtest.Data$Attrition) #.8707


tune_grid4 <- expand.grid(
  nrounds = seq(from = 50, to = nrounds, by = 50), #500
  eta = xgb_tune$bestTune$eta,
  max_depth = xgb_tune2$bestTune$max_depth,
  gamma = c(0, 0.05, 0.1, 0.5, 0.7, 0.9, 1.0), #0.1
  colsample_bytree = xgb_tune3$bestTune$colsample_bytree,
  min_child_weight = xgb_tune2$bestTune$min_child_weight,
  subsample = xgb_tune3$bestTune$subsample
)

xgb_tune4 <- caret::train(
```

```
  formula,
  data = XGBtrain.Data,
  trControl = tune_control,
  tuneGrid = tune_grid4,
  method = "xgbTree",
  verbose = TRUE
)
ggplot(xgb_tune4)
predictions4<-predict(xgb_tune4,XGBtest.Data)
confusionMatrix(predictions4,XGBtest.Data$Attrition) #0.8707


tune_grid5 <- expand.grid(
  nrounds = seq(from = 100, to = 10000, by = 100), #300
  eta = c(0.01, 0.015, 0.025, 0.05, 0.1), #0.025
  max_depth = xgb_tune2$bestTune$max_depth,
  gamma = xgb_tune4$bestTune$gamma,
  colsample_bytree = xgb_tune3$bestTune$colsample_bytree,
  min_child_weight = xgb_tune2$bestTune$min_child_weight,
  subsample = xgb_tune3$bestTune$subsample
)
xgb_tune5 <- caret::train(
  formula,
  data = XGBtrain.Data,
  trControl = tune_control,
  tuneGrid = tune_grid5,
  method = "xgbTree",
  verbose = TRUE
)

ggplot(xgb_tune5)
predictions5<-predict(xgb_tune5,XGBtest.Data)
confusionMatrix(predictions5,XGBtest.Data$Attrition) #0.8673
```

XGBoost

We are using the best tunned hyperparameters (mentioned in above section) in below model.

```
library(xgboost)
```

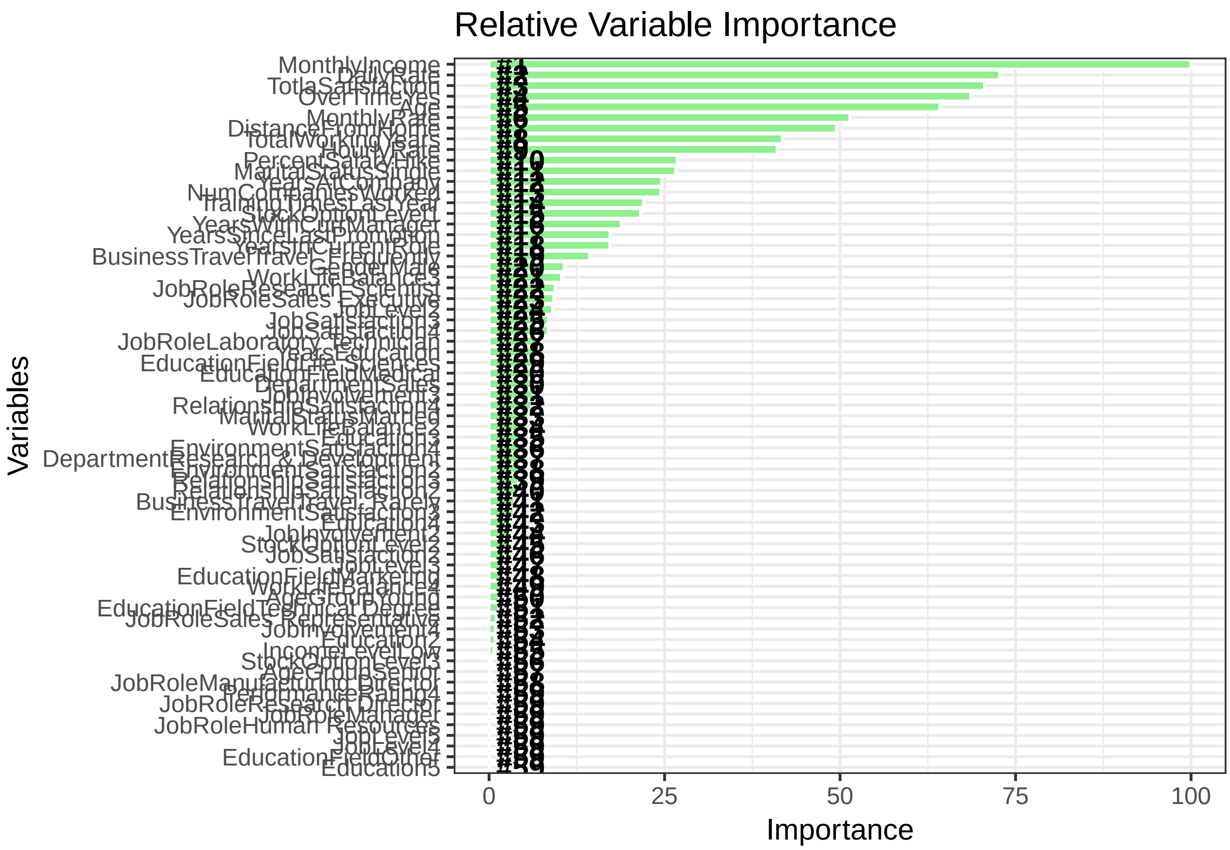
```
##
## Attaching package: 'xgboost'

## The following object is masked from 'package:dplyr':
##
##     slice
```

```
set.seed(123)
xgbData <- HR_data
indexes <- sample(1:nrow(xgbData), size=0.8*nrow(xgbData))
XGBtrain.Data <- xgbData[indexes,]
XGBtest.Data <- xgbData[-indexes,]

formula = Attrition~.
fitControl <- trainControl(method="cv", number = 3,classProbs = TRUE )
```

```
xgbGrid <- expand.grid(nrounds = 500,
                       max_depth = 15,
                       eta = .05,
                       gamma = 0.1,
                       colsample_bytree = .8,
                       min_child_weight = 3,
                       subsample = 0.75
)
XGB.model <- train(formula, data = XGBtrain.Data,
                   method = "xgbTree"
                   ,trControl = fitControl
                   , verbose=0
                   , maximize=FALSE
                   ,tuneGrid = xgbGrid
)
importance <- varImp(XGB.model)
varImportance <- data.frame(Variables = row.names(importance[[1]]),
                            Importance = round(importance[[1]]$Overall,2))
```

We are now creating a rank variable based on importance of variables

```
rankImportance <- varImportance %>%
  mutate(Rank = paste0('#',dense_rank(desc(Importance))))
ggplot(rankImportance, aes(x = reorder(Variables, Importance),
                           y = Importance)) +
  geom_bar(stat='identity',colour="white", fill = "lightgreen") +
  geom_text(aes(x = Variables, y = 1, label = Rank),
            hjust=0, vjust=.5, size = 4, colour = 'black',
            fontface = 'bold') +
  labs(x = 'Variables', title = 'Relative Variable Importance') +
  coord_flip() +
  theme_bw()
```



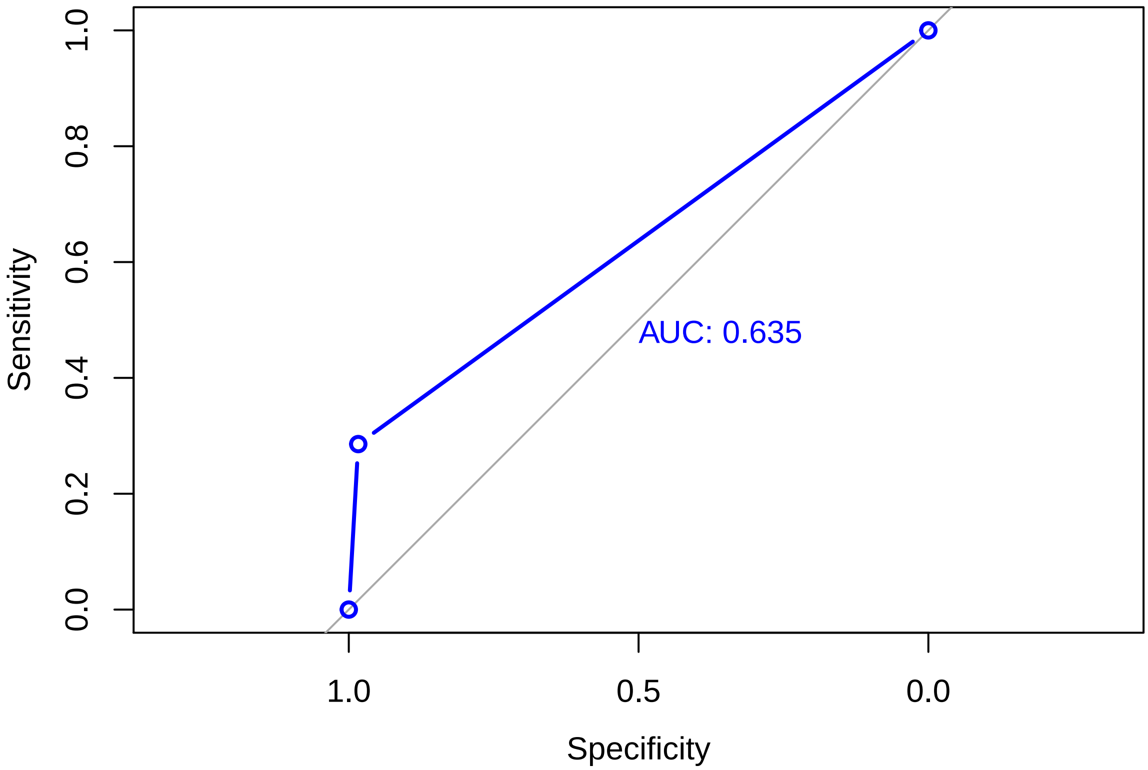
We can observe that Monthly Income, OverTime, Total Satisfaction, Total Working Years and Daily Rate are top 5.

```
XGB.prd <- predict(XGB.model,XGBtest.Data)
confusionMatrix(XGB.prd, XGBtest.Data$Attrition)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  241  35
##        Yes   4  14
##
##                Accuracy : 0.8673
##                  95% CI : (0.8231, 0.9039)
##     No Information Rate : 0.8333
##     P-Value [Acc > NIR] : 0.06545
##
##                   Kappa : 0.3607
##
##  Mcnemar's Test P-Value : 1.556e-06
##
##             Sensitivity : 0.9837
##             Specificity : 0.2857
##          Pos Pred Value : 0.8732
##          Neg Pred Value : 0.7778
##              Prevalence : 0.8333
##          Detection Rate : 0.8197
```

```
##    Detection Prevalence : 0.9388
##       Balanced Accuracy : 0.6347
##
##        'Positive' Class : No
##
```

```
XGB.plot <- plot.roc (as.numeric(XGBtest.Data$Attrition), as.numeric(XGB.prd),lwd=2, type="b", print.auc
```



We can see that accuracy is improved and AUC as well.

Unbalanaced Data Issue

We had observerd that our data is highly unbalanced (as shown in Attrition visualization graph).

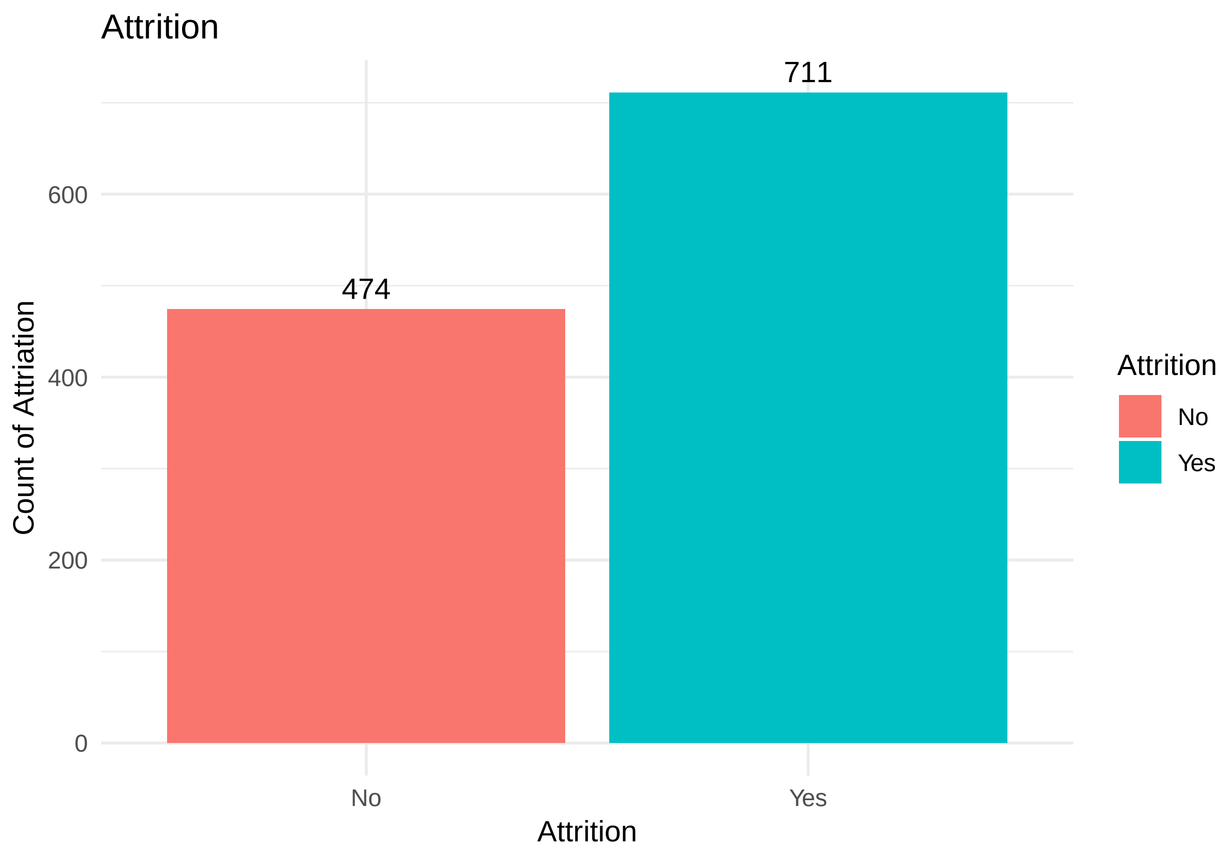
Lets solve the unbalanced data problem in the dataset using SMOTE method.

```
library(DMwR)
Classcount = table(HR_data$Attrition)
# Over Sampling
over = ( (0.6 * max(Classcount)) - min(Classcount) ) / min(Classcount) #2.121
# Under Sampling
under = (0.4 * max(Classcount)) / (min(Classcount) * over) # 0.98

over = round(over, 1) * 100 #210
under = round(under, 1) * 100 #100
#Generate the balanced data set
BalancedData = SMOTE(Attrition~., HR_data, perc.over = over, k = 5, perc.under = under)
```

Lets check the output of the Balancing

```
BalancedData %>%
  group_by(Attrition) %>%
```

```r
  tally() %>%
  ggplot(aes(x = Attrition, y = n,fill=Attrition)) +
  geom_bar(stat = "identity") +
  theme_minimal()+
  labs(x="Attrition", y="Count of Attriation")+
  ggtitle("Attrition")+
  geom_text(aes(label = n), vjust = -0.5, position = position_dodge(0.9))
```



XG Boost with Balanced data

Tunning hyperparameters again for the balanced data

```r
# Tuning XBGTree using Caret Package
#  Hyperparameters to tune:
# nrounds: Number of trees, default: 100
#  max_depth: Maximum tree depth, default: 6
#  eta: Learning rate, default: 0.3
#  gamma: Used for tuning of Regularization, default: 0
#  colsample_bytree: Column sampling, default: 1
#  min_child_weight: Minimum leaf weight, default: 1
#  subsample: Row sampling, default: 1
#  We'll break down the tuning of these into five sections:
#
#   Fixing learning rate eta and number of iterations nrounds
#  M aximum depth max_depth and child weight min_child_weight
#  Setting column colsample_bytree and row sampling subsample
#  Experimenting with different gamma values
```

```r
#  Reducing the learning rate eta

# set seed
library(xgboost)
set.seed(123)
xgbData <-  BalancedData
indexes = sample(1:nrow(xgbData), size=0.8*nrow(xgbData))
BLtrain.Data <- xgbData[indexes,]
BLtest.Data <- xgbData[-indexes,]
# note to start nrounds from 200, as smaller learning rates result in errors so
# big with lower starting points that they'll mess the scales


formula = Attrition~.
nrounds <- 1000

tune_grid <- expand.grid(
  nrounds = seq(from = 200, to = nrounds, by = 50), #750
  eta = c(0.025, 0.05, 0.1, 0.3), #0.05
  max_depth = c(2, 3, 4, 5, 6,8,10,15,20,25,30), #2 (2, 3, 4,)
  gamma = 0,
  colsample_bytree = 1,
  min_child_weight = 1,
  subsample = 1
)

tune_control <- caret::trainControl(
  method = "cv", # cross-validation
  number = 3, # with n folds
  classProbs = TRUE
)

xgb_tune <- caret::train(
  formula,
  data = BLtrain.Data,
  trControl = tune_control,
  tuneGrid = tune_grid,
  method = "xgbTree"
)
predictions<-predict(xgb_tune,BLtest.Data)
confusionMatrix(predictions,BLtest.Data$Attrition) #0.9325
ggplot(xgb_tune)


#######2
tune_grid2 <- expand.grid(
  nrounds = seq(from = 50, to = nrounds, by = 50), #1000
  eta = xgb_tune$bestTune$eta,

  max_depth = c(1,2,3,5,10,15,20,25,30,35), #2
  gamma = 0,
  colsample_bytree = 1,
  min_child_weight = c(1, 2, 3), #2
```

```
  subsample = 1
)

xgb_tune2 <- caret::train(
  formula,
  data = BLtrain.Data,
  trControl = tune_control,
  tuneGrid = tune_grid2,
  method = "xgbTree",
  verbose = TRUE
)

ggplot(xgb_tune2)
predictions2<-predict(xgb_tune2,BLtest.Data)
confusionMatrix(predictions2,BLtest.Data$Attrition) #0.9325


###########3
tune_grid3 <- expand.grid(
  nrounds = seq(from = 50, to = nrounds, by = 50), #850
  eta = xgb_tune$bestTune$eta,
  max_depth = xgb_tune2$bestTune$max_depth,
  gamma = 0,
  colsample_bytree = c(0.4, 0.6, 0.8, 1.0), #0.6
  min_child_weight = xgb_tune2$bestTune$min_child_weight,
  subsample = c(0.5, 0.75, 1.0) #0.5
)

xgb_tune3 <- caret::train(
  formula,
  data = BLtrain.Data,
  trControl = tune_control,
  tuneGrid = tune_grid3,
  method = "xgbTree",
  verbose = TRUE
)

ggplot(xgb_tune3)
predictions3<-predict(xgb_tune3,BLtest.Data)
confusionMatrix(predictions3,BLtest.Data$Attrition) #0.8945

################4
tune_grid4 <- expand.grid(
  nrounds = seq(from = 50, to = nrounds, by = 50), #850
  eta = xgb_tune$bestTune$eta,
  max_depth = xgb_tune2$bestTune$max_depth,
  gamma = c(0, 0.05, 0.1, 0.5, 0.7, 0.9, 1.0), #0.7
  colsample_bytree = xgb_tune3$bestTune$colsample_bytree,
  min_child_weight = xgb_tune2$bestTune$min_child_weight,
  subsample = xgb_tune3$bestTune$subsample
)

xgb_tune4 <- caret::train(
```

```
  formula,
  data = BLtrain.Data,
  trControl = tune_control,
  tuneGrid = tune_grid4,
  method = "xgbTree",
  verbose = TRUE
)
ggplot(xgb_tune4)
predictions4<-predict(xgb_tune4,BLtest.Data)
confusionMatrix(predictions4,BLtest.Data$Attrition) #0.8987



######################5
tune_grid5 <- expand.grid(
  nrounds = seq(from = 100, to = 10000, by = 100), #2000
  eta = c(0.01, 0.015, 0.025, 0.05, 0.1), #0.015
  max_depth = xgb_tune2$bestTune$max_depth,
  gamma = xgb_tune4$bestTune$gamma,
  colsample_bytree = xgb_tune3$bestTune$colsample_bytree,
  min_child_weight = xgb_tune2$bestTune$min_child_weight,
  subsample = xgb_tune3$bestTune$subsample
)
xgb_tune5 <- caret::train(
  formula,
  data = BLtrain.Data,
  trControl = tune_control,
  tuneGrid = tune_grid5,
  method = "xgbTree",
  verbose = TRUE
)

ggplot(xgb_tune5)
predictions5<-predict(xgb_tune5,BLtest.Data)
confusionMatrix(predictions5,BLtest.Data$Attrition) #0.8987
```

XGBoost

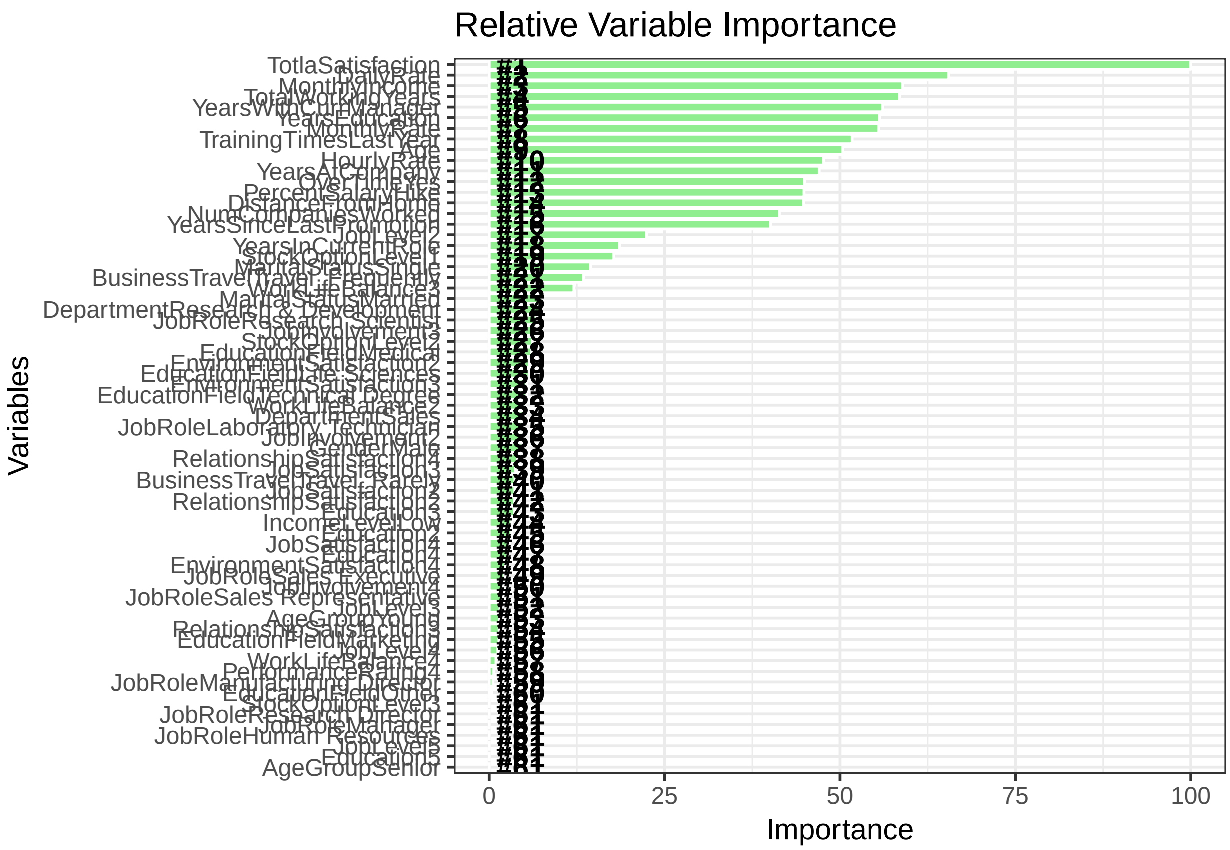
We are using xgboost with balanced data. Also, using best tunned hyperparameters.

```
set.seed(123)
xgbData <- BalancedData
indexes = sample(1:nrow(xgbData), size=0.8*nrow(xgbData))
BLtrain.Data <- xgbData[indexes,]
BLtest.Data <- xgbData[-indexes,]

formula = Attrition~.
fitControl <- trainControl(method="cv", number = 3,classProbs = TRUE )
NewxgbGrid <- expand.grid(nrounds = 800,
                          max_depth = 2,
                          eta = .05,
                          gamma = 0,
                          colsample_bytree = 0.8,
                          min_child_weight = 2,
```

```
                          subsample = 0.5
)

# from tunegrid 5
NewxgbGrid <- expand.grid(nrounds = 3000,
                          max_depth = 5,
                          eta = .05,
                          gamma = 0.1,
                          colsample_bytree = 0.6,
                          min_child_weight = 2,
                          subsample = 0.5
)


NewXGB.model = train(formula, data = BLtrain.Data,
                     method = "xgbTree"
                     ,trControl = fitControl
                     , verbose=0
                     , maximize=FALSE
                     ,tuneGrid = NewxgbGrid
                     ,na.action = na.pass
)

importance <- varImp(NewXGB.model)
varImportance <- data.frame(Variables = row.names(importance[[1]]),
                            Importance = round(importance[[1]]$Overall,2))
```

Creating a rank variable based on importance

```
rankImportance <- varImportance %>%
  mutate(Rank = paste0('#',dense_rank(desc(Importance))))
ggplot(rankImportance, aes(x = reorder(Variables, Importance),
                           y = Importance)) +
  geom_bar(stat='identity',colour="white", fill = "lightgreen") +
  geom_text(aes(x = Variables, y = 1, label = Rank),
            hjust=0, vjust=.5, size = 4, colour = 'black',
            fontface = 'bold') +
  labs(x = 'Variables', title = 'Relative Variable Importance') +
  coord_flip() +
  theme_bw()
```



We can observe that Total Satisfaction, Daily Rate, Monthly Income, Total Working Years and Years with Current Manager are top 5.

Lets check the accuracy now.

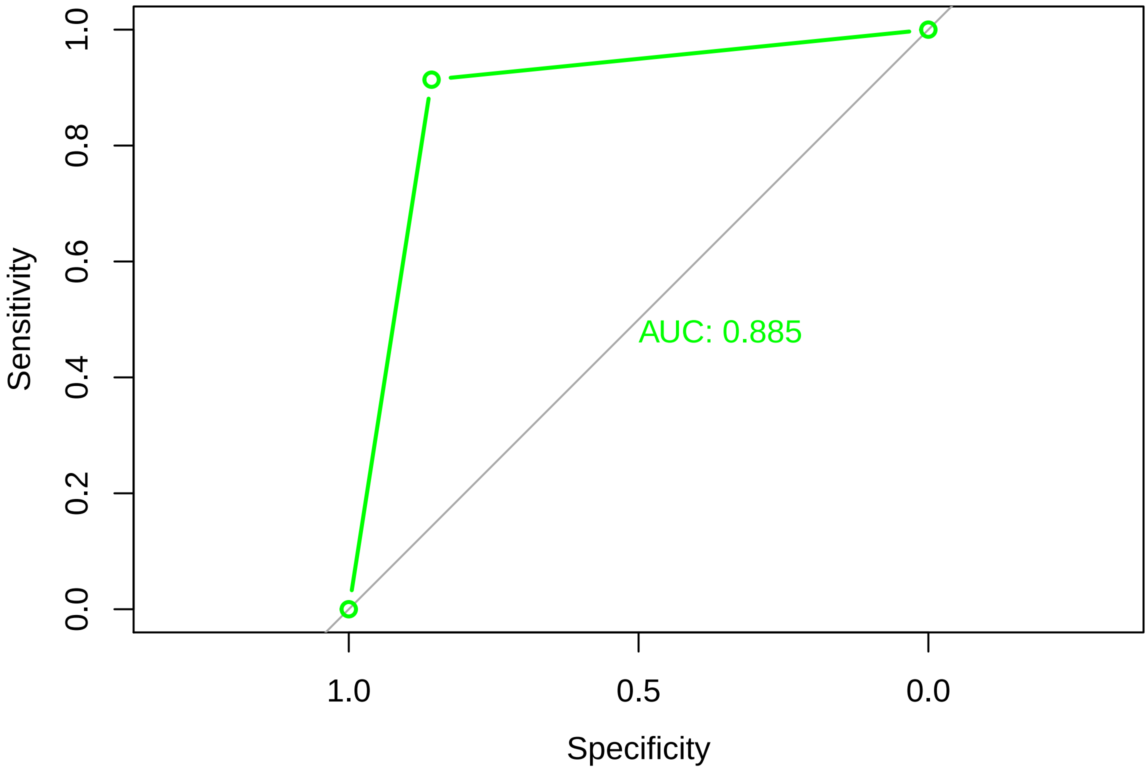
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No   84  12
##        Yes  14 127
##
##                Accuracy : 0.8903
##                  95% CI : (0.8434, 0.9271)
##     No Information Rate : 0.5865
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.7731
##
##  Mcnemar's Test P-Value : 0.8445
##
##             Sensitivity : 0.8571
##             Specificity : 0.9137
##          Pos Pred Value : 0.8750
##          Neg Pred Value : 0.9007
##              Prevalence : 0.4135
##          Detection Rate : 0.3544
##    Detection Prevalence : 0.4051
```

```
##        Balanced Accuracy : 0.8854
##
##         'Positive' Class : No
##
```

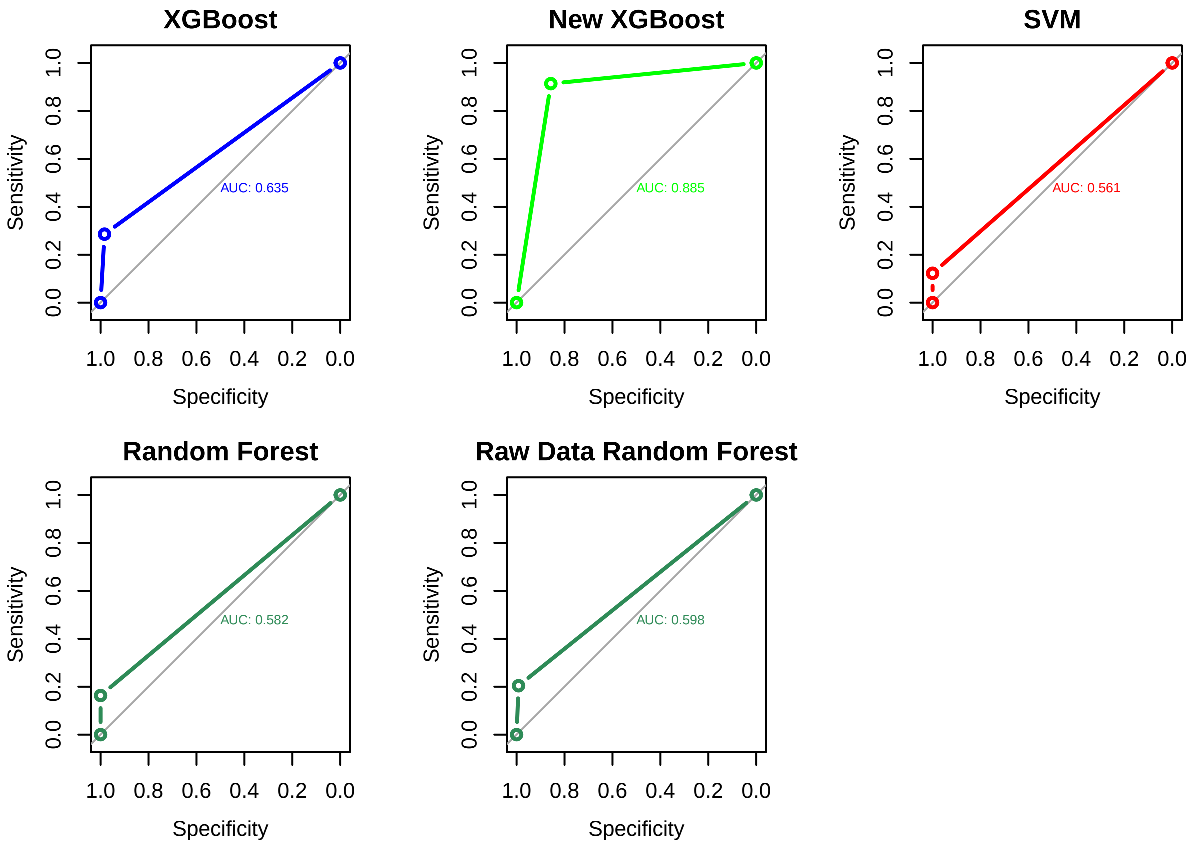
AUC

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```



Result



As we can see, Of all the models xgboost with balanced data gives the best result. Accuracy - 0.8903 & AUC - 0.885

Also,the optimizer may find a different local minimum, so the accuracy for the run might be different in different computers.

In my other macbook the same code gave an accuracy of .8974

Conclusion

We tried using different approach to build our model to predict if an employee is going to leave or will continue working in the same company.

The best among all the approach is xgboost (with balanced data) with an accuracy of 0.8903 (it could be different in different system as mentioned in “Result” section)

We used Total Satisfaction, Daily Rate, Monthly Income, Total Working Years and Years with Current Manager to build our model using xgboost (with balanced data) to predict if the employee will leave the company or not.