HR Analysis

Consagous Technologies

26 April 2019

Objective: To predict which valuable employees will leave next.

Fields in the dataset include:

Employee satisfaction level

Last evaluation

Number of projects

Average monthly hours

Time spent at the company

Whether they have had a work accident

Whether they have had a promotion in the last 5 years

Department

Salary

Whether the employee has left

Required Library

```r
library(dplyr)
```

```
##
## 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
```

```r
library(ROSE)
```

```
## Loaded ROSE 0.0-3
```

```r
library(ggplot2)
library(ROCR)
```

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

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

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

```r
library(caret)
```

```
## Loading required package: lattice
```

```r
library(tree)
library(randomForest)
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

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

Read Data

```r
setwd("/home/consagous/Documents/Predictive Model POC")
md <- read.csv("hr_train.csv")
str(md)
```

```
## 'data.frame':    10499 obs. of  10 variables:
##  $ satisfaction_level   : num  0.42 0.66 0.55 0.22 0.2 0.83 0.87 0.85 0.89 0.45 ...
##  $ last_evaluation      : num  0.46 0.77 0.49 0.88 0.72 0.84 0.49 0.99 0.92 0.56 ...
##  $ number_project       : int  2 2 5 4 6 4 2 3 5 2 ...
##  $ average_montly_hours : int  150 171 240 213 224 206 251 208 237 154 ...
##  $ time_spend_company   : int  3 2 3 3 4 2 3 2 5 3 ...
##  $ Work_accident        : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ left                 : int  1 0 0 0 1 0 0 0 0 1 ...
##  $ promotion_last_5years: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ sales                : Factor w/ 10 levels "accounting","hr",..: 8 10 10 10 10 8 8 6 8 5 ...
##  $ salary               : Factor w/ 3 levels "high","low","medium": 3 3 1 3 3 3 3 2 3 2 ...
```

Overview the Balancing of Data

```r
table(md$left)
```

```
##
##    0    1
## 7424 3075
```

Here data is imbalance so balanced data set with over-sampling

```
over.md <- ovun.sample(left~., data=md,
                                  p=0.5, seed=1,
                                  method="over")$data
table(over.md$left)
```

```
##
##    0    1
## 7424 7369
```

Now the frequency of both the classes are somewhere same

Logistic Regression

Making Dummy Variables

```
str(over.md)
```

```
## 'data.frame':    14793 obs. of  10 variables:
##  $ satisfaction_level   : num  0.66 0.55 0.22 0.83 0.87 0.85 0.89 0.49 0.82 0.59 ...
##  $ last_evaluation      : num  0.77 0.49 0.88 0.84 0.49 0.99 0.92 0.63 0.58 0.97 ...
##  $ number_project       : int  2 5 4 4 2 3 5 3 5 3 ...
##  $ average_montly_hours : int  171 240 213 206 251 208 237 181 227 257 ...
##  $ time_spend_company   : int  2 3 3 2 3 2 5 3 3 3 ...
##  $ Work_accident        : int  0 0 1 0 0 0 0 1 0 0 ...
##  $ left                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ promotion_last_5years: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ sales                : Factor w/ 10 levels "accounting","hr",..: 10 10 10 8 8 6 8 6 7 5 ...
##  $ salary               : Factor w/ 3 levels "high","low","medium": 3 1 3 3 3 2 3 3 2 2 ...
```

There are two variables which are categorical so we convert them in dummy variables

```
table(over.md$sales)
```

```
##
##  accounting          hr          IT  management   marketing product_mng
##         786         773        1104         578         865         865
##       RandD       sales     support   technical
##         721        4110        2229        2762
```

There are 10 classes in variable, so we are making 9 dummy variable and taking least

frequency variabole manaagement as base category.

```
over.md <- over.md %>%
  mutate(Support = as.numeric(sales == "support"),
        Technical =  as.numeric(sales == "technical"),
        Sales = as.numeric(sales == "sales"),
        IT = as.numeric(sales == "IT"),
```

```r
       Mktg = as.numeric(sales == "marketing"),
       Prod_Mgt = as.numeric(sales == "product_mng"),
       Acct = as.numeric(sales == "accounting"),
       RnD = as.numeric(sales == "RandD"),
       HR = as.numeric(sales == "hr")) %>%
  select(-sales)
```

Similar with variable "salary"

```r
table(over.md$salary)
```

```
##
##   high    low medium
##   1033   7545   6215
```

```r
over.md <- over.md %>%
  mutate(Low_Salary = as.numeric(salary == "low"),
         Med_Salary = as.numeric(salary == "medium")) %>%
  select(-salary)
```

Now look at the structure

```r
str(over.md)
```

```
## 'data.frame':    14793 obs. of  19 variables:
##  $ satisfaction_level   : num  0.66 0.55 0.22 0.83 0.87 0.85 0.89 0.49 0.82 0.59 ...
##  $ last_evaluation      : num  0.77 0.49 0.88 0.84 0.49 0.99 0.92 0.63 0.58 0.97 ...
##  $ number_project       : int  2 5 4 4 2 3 5 3 5 3 ...
##  $ average_montly_hours : int  171 240 213 206 251 208 237 181 227 257 ...
##  $ time_spend_company   : int  2 3 3 2 3 2 5 3 3 3 ...
##  $ Work_accident        : int  0 0 1 0 0 0 0 1 0 0 ...
##  $ left                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ promotion_last_5years: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Support              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Technical            : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ Sales                : num  0 0 0 1 1 0 1 0 0 0 ...
##  $ IT                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Mktg                 : num  0 0 0 0 0 0 0 0 0 1 ...
##  $ Prod_Mgt             : num  0 0 0 0 0 1 0 1 0 0 ...
##  $ Acct                 : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ RnD                  : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ HR                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Low_Salary           : num  0 0 0 0 0 1 0 0 1 1 ...
##  $ Med_Salary           : num  1 0 1 1 1 0 1 1 0 0 ...
```

Now all variables are in numeric form

Fit the Model

```r
fit <- glm(left~., data = over.md, family = "binomial")
summary(fit)
```

```
##
## Call:
## glm(formula = left ~ ., family = "binomial", data = over.md)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3341  -1.0192  -0.3781   1.0210   2.4321
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -0.8638936  0.1463680  -5.902 3.59e-09 ***
## satisfaction_level    -2.6186334  0.0760950 -34.413  < 2e-16 ***
## last_evaluation        0.7375834  0.1239337   5.951 2.66e-09 ***
## number_project        -0.2060478  0.0175754 -11.724  < 2e-16 ***
## average_montly_hours   0.0030506  0.0004281   7.126 1.03e-12 ***
## time_spend_company     0.2304014  0.0138785  16.601  < 2e-16 ***
## Work_accident         -0.6270805  0.0556267 -11.273  < 2e-16 ***
## promotion_last_5years -0.3113643  0.1361943  -2.286 0.022244 *
## Support                0.3126647  0.1060285   2.949 0.003189 **
## Technical              0.4020762  0.1039783   3.867 0.000110 ***
## Sales                  0.2560843  0.1010739   2.534 0.011289 *
## IT                     0.0283314  0.1159721   0.244 0.807003
## Mktg                   0.3227048  0.1198773   2.692 0.007103 **
## Prod_Mgt               0.1883979  0.1198394   1.572 0.115930
## Acct                   0.4470084  0.1230401   3.633 0.000280 ***
## RnD                    0.1552827  0.1248067   1.244 0.213431
## HR                     0.4200269  0.1226607   3.424 0.000616 ***
## Low_Salary             1.2276092  0.0810457  15.147  < 2e-16 ***
## Med_Salary             0.8839298  0.0815065  10.845  < 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: 20507  on 14792  degrees of freedom
## Residual deviance: 18175  on 14774  degrees of freedom
## AIC: 18213
##
## Number of Fisher Scoring iterations: 4
```

```r
fit <- glm(left~. -IT, data = over.md, family = "binomial")
summary(fit)
```

```
##
## Call:
## glm(formula = left ~ . - IT, family = "binomial", data = over.md)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3335  -1.0196  -0.3776   1.0217   2.4333
##
```

```
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)    
## (Intercept)          -0.8469457  0.1288336  -6.574 4.90e-11 ***
## satisfaction_level   -2.6187321  0.0760931 -34.415  < 2e-16 ***
## last_evaluation       0.7375626  0.1239312   5.951 2.66e-09 ***
## number_project       -0.2060135  0.0175745 -11.722  < 2e-16 ***
## average_montly_hours  0.0030540  0.0004279   7.138 9.47e-13 ***
## time_spend_company    0.2301091  0.0138249  16.645  < 2e-16 ***
## Work_accident        -0.6272628  0.0556203 -11.278  < 2e-16 ***
## promotion_last_5years -0.3143296  0.1356407  -2.317 0.020484 *  
## Support               0.2934423  0.0710234   4.132 3.60e-05 ***
## Technical             0.3829031  0.0681566   5.618 1.93e-08 ***
## Sales                 0.2369340  0.0637539   3.716 0.000202 ***
## Mktg                  0.3037038  0.0911854   3.331 0.000867 ***
## Prod_Mgt              0.1692725  0.0906976   1.866 0.061994 .  
## Acct                  0.4278486  0.0947716   4.515 6.35e-06 ***
## RnD                   0.1361551  0.0971592   1.401 0.161106    
## HR                    0.4008818  0.0943278   4.250 2.14e-05 ***
## Low_Salary            1.2303843  0.0802577  15.330  < 2e-16 ***
## Med_Salary            0.8865509  0.0808082  10.971  < 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: 20507  on 14792  degrees of freedom
## Residual deviance: 18175  on 14775  degrees of freedom
## AIC: 18211
## 
## Number of Fisher Scoring iterations: 4
```

```r
fit <- glm(left~. -IT -RnD, data = over.md, family = "binomial")
summary(fit)
```

```
## 
## Call:
## glm(formula = left ~ . - IT - RnD, family = "binomial", data = over.md)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.3311  -1.0191  -0.3757   1.0209   2.4350  
## 
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)    
## (Intercept)          -0.8087049  0.1258472  -6.426 1.31e-10 ***
## satisfaction_level   -2.6167450  0.0760728 -34.398  < 2e-16 ***
## last_evaluation       0.7360364  0.1238974   5.941 2.84e-09 ***
## number_project       -0.2059010  0.0175700 -11.719  < 2e-16 ***
## average_montly_hours  0.0030584  0.0004278   7.149 8.75e-13 ***
## time_spend_company    0.2292218  0.0138055  16.604  < 2e-16 ***
## Work_accident        -0.6257347  0.0556009 -11.254  < 2e-16 ***
## promotion_last_5years -0.3141788  0.1356979  -2.315 0.020598 *  
## Support               0.2513301  0.0642945   3.909 9.27e-05 ***
## Technical             0.3408683  0.0611395   5.575 2.47e-08 ***
## Sales                 0.1949122  0.0561975   3.468 0.000524 ***
```

```
## Mktg                   0.2618522  0.0861089   3.041 0.002358 **
## Prod_Mgt               0.1274370  0.0855963   1.489 0.136536
## Acct                   0.3859241  0.0898840   4.294 1.76e-05 ***
## HR                     0.3588770  0.0893936   4.015 5.96e-05 ***
## Low_Salary             1.2359951  0.0801532  15.420  < 2e-16 ***
## Med_Salary             0.8921637  0.0807020  11.055  < 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: 20507  on 14792  degrees of freedom
## Residual deviance: 18177  on 14776  degrees of freedom
## AIC: 18211
##
## Number of Fisher Scoring iterations: 4
```

```r
fit <- glm(left~. -IT -RnD -Prod_Mgt, data = over.md, family = "binomial")
summary(fit)
```

```
##
## Call:
## glm(formula = left ~ . - IT - RnD - Prod_Mgt, family = "binomial",
##     data = over.md)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3306  -1.0191  -0.3767   1.0200   2.4360
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -0.7757037  0.1238441  -6.264 3.76e-10 ***
## satisfaction_level    -2.6153798  0.0760685 -34.382  < 2e-16 ***
## last_evaluation        0.7357267  0.1238790   5.939 2.87e-09 ***
## number_project        -0.2061907  0.0175682 -11.737  < 2e-16 ***
## average_montly_hours   0.0030601  0.0004278   7.153 8.50e-13 ***
## time_spend_company     0.2290596  0.0138038  16.594  < 2e-16 ***
## Work_accident         -0.6251979  0.0555997 -11.245  < 2e-16 ***
## promotion_last_5years -0.3243683  0.1354806  -2.394 0.016656 *
## Support                0.2162280  0.0597797   3.617 0.000298 ***
## Technical              0.3059019  0.0564095   5.423 5.86e-08 ***
## Sales                  0.1599718  0.0510229   3.135 0.001717 **
## Mktg                   0.2271277  0.0828665   2.741 0.006127 **
## Acct                   0.3509725  0.0867401   4.046 5.20e-05 ***
## HR                     0.3238689  0.0862207   3.756 0.000172 ***
## Low_Salary             1.2387648  0.0801179  15.462  < 2e-16 ***
## Med_Salary             0.8952847  0.0806599  11.100  < 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: 20507  on 14792  degrees of freedom
## Residual deviance: 18179  on 14777  degrees of freedom
## AIC: 18211
```

```
##
## Number of Fisher Scoring iterations: 4
```

Final Model

```
model <- glm(left ~ satisfaction_level + last_evaluation + number_project +
                       average_montly_hours + time_spend_company + Work_accident +
                       promotion_last_5years + Support + Technical + Sales +
                       Mktg + Acct + HR + Low_Salary + Med_Salary, data = over.md, family = "binomial")
```

Prediction with training data

```
over.md$score=predict(model, newdata=over.md,type = "response")
head(over.md$left)
```

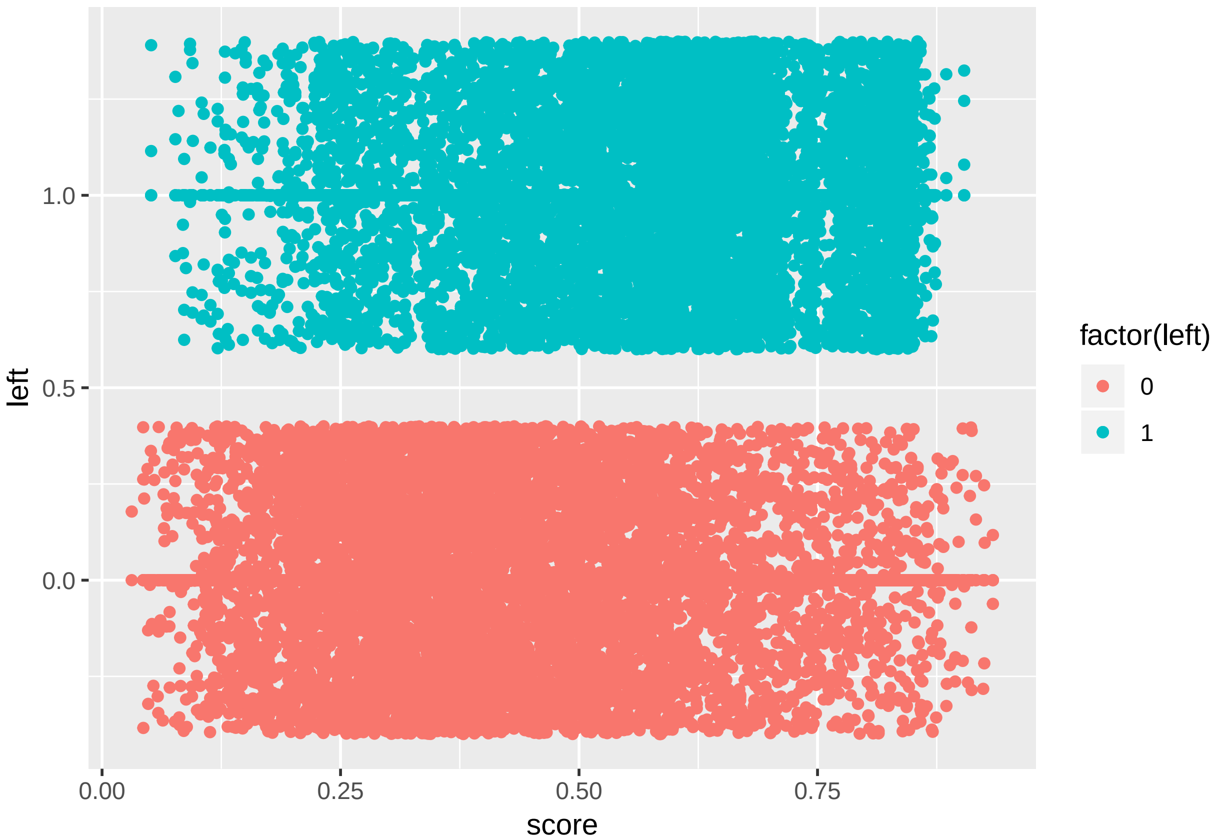
```
## [1] 0 0 0 0 0 0
```

```
head(over.md$score)
```

```
## [1] 0.4588070 0.2391929 0.5954530 0.2670839 0.3560744 0.3645666
```

Overview through the graph

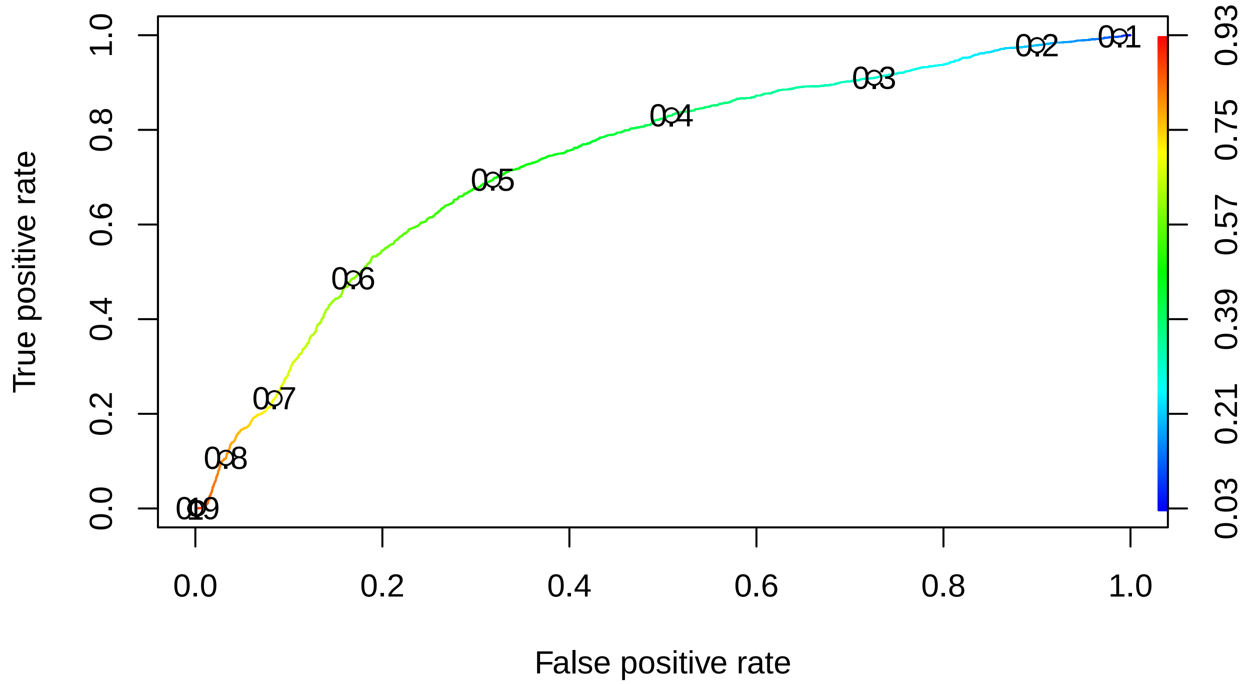
```
library(ggplot2)
ggplot(over.md,aes(y=left,x=score,color=factor(left)))+
  geom_point()+geom_jitter()
```



Here to much overlapping in prediction it could be improved by taking set.seed but here we are proceed with same

Consideration of Cutoff Value

```
ROCRPred <- prediction(over.md$score, over.md$left)
ROCRPerf <- performance(ROCRPred, "tpr", "fpr")
plot(ROCRPerf, colorize=TRUE, print.cutoffs.at=seq(.1, by = 0.1))
```



Here 0.5 looks better cutoff for this model

Area under the curve

```
auc <- performance(ROCRPred, "auc")
auc <- unlist(slot(auc, "y.values"))
auc <- round(auc,4)
auc
```

```
## [1] 0.7314
```

Making Predictions

```
res <- predict(model, over.md, type = "response")
PredictedValue <- res>.5
pv <- as.numeric(PredictedValue)
pv <- as.factor(pv)
over.md$left <- as.factor(over.md$left)
confusionMatrix(pv, over.md$left)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 5062 2251
##          1 2362 5118
##
##                Accuracy : 0.6882
##                  95% CI : (0.6806, 0.6956)
```

```
##     No Information Rate : 0.5019
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.3764
##
##  Mcnemar's Test P-Value : 0.1053
##
##             Sensitivity : 0.6818
##             Specificity : 0.6945
##          Pos Pred Value : 0.6922
##          Neg Pred Value : 0.6842
##              Prevalence : 0.5019
##          Detection Rate : 0.3422
##    Detection Prevalence : 0.4944
##       Balanced Accuracy : 0.6882
##
##        'Positive' Class : 0
##
```

Result: The accuracy of Logistic Regression model is 68.82%, and Sensitivity(True Positive Rate)

is 68.18% and Specificity(True Negative Rate) is 69.45%, the difference between them is minimum

means model is good.

Prediction for Test Data:

```r
test.md <- read.csv("hr_test.csv")
str(test.md)
```

```
## 'data.frame':    4500 obs. of  9 variables:
##  $ satisfaction_level   : num  0.38 0.8 0.1 0.45 0.11 0.41 0.38 0.45 0.4 0.4 ...
##  $ last_evaluation      : num  0.53 0.86 0.77 0.54 0.81 0.55 0.54 0.47 0.53 0.49 ...
##  $ number_project       : int  2 5 6 2 6 2 2 2 2 2 ...
##  $ average_montly_hours : int  157 262 247 135 305 148 143 160 158 135 ...
##  $ time_spend_company   : int  3 6 4 3 4 3 3 3 3 3 ...
##  $ Work_accident        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ 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 2 2 2 2 2 2 2 2 ...
```

```r
test.data <- test.md %>%
  mutate(Support = as.numeric(sales == "support"),
         Technical =  as.numeric(sales == "technical"),
         Sales = as.numeric(sales == "sales"),
         IT = as.numeric(sales == "IT"),
         Mktg = as.numeric(sales == "marketing"),
         Prod_Mgt = as.numeric(sales == "product_mng"),
         Acct = as.numeric(sales == "accounting"),
         RnD = as.numeric(sales == "RandD"),
         HR = as.numeric(sales == "hr")) %>%
```

```r
  select(-sales)

test.data<- test.data %>%
  mutate(Low_Salary = as.numeric(salary == "low"),
         Med_Salary = as.numeric(salary == "medium")) %>%
  select(-salary)

str(test.data)
```

```
## 'data.frame':    4500 obs. of  18 variables:
##  $ satisfaction_level   : num  0.38 0.8 0.1 0.45 0.11 0.41 0.38 0.45 0.4 0.4 ...
##  $ last_evaluation      : num  0.53 0.86 0.77 0.54 0.81 0.55 0.54 0.47 0.53 0.49 ...
##  $ number_project       : int  2 5 6 2 6 2 2 2 2 2 ...
##  $ average_montly_hours : int  157 262 247 135 305 148 143 160 158 135 ...
##  $ time_spend_company   : int  3 6 4 3 4 3 3 3 3 3 ...
##  $ Work_accident        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ promotion_last_5years: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Support              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Technical            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Sales                : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ IT                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Mktg                 : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Prod_Mgt             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Acct                 : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ RnD                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ HR                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Low_Salary           : num  1 0 1 1 1 1 1 1 1 1 ...
##  $ Med_Salary           : num  0 1 0 0 0 0 0 0 0 0 ...
```

```r
res_test <- predict(model, newdata =test.data, type = "response")
PredictedValue <- res_test>.5
pv <- as.numeric(PredictedValue)
table(pv)
```

```
## pv
##    0    1
## 2554 1946
```

Through testing data the prediction of our model shows that out of 4500 employees, 1946 employees

may leave next.

Decision Tree Model

```r
str(md)
```

```
## 'data.frame':    10499 obs. of  10 variables:
##  $ satisfaction_level   : num  0.42 0.66 0.55 0.22 0.2 0.83 0.87 0.85 0.89 0.45 ...
##  $ last_evaluation      : num  0.46 0.77 0.49 0.88 0.72 0.84 0.49 0.99 0.92 0.56 ...
##  $ number_project       : int  2 2 5 4 6 4 2 3 5 2 ...
##  $ average_montly_hours : int  150 171 240 213 224 206 251 208 237 154 ...
##  $ time_spend_company   : int  3 2 3 3 4 2 3 2 5 3 ...
```

```
##  $ Work_accident        : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ left                 : int  1 0 0 0 1 0 0 0 0 1 ...
##  $ promotion_last_5years: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ sales                : Factor w/ 10 levels "accounting","hr",..: 8 10 10 10 10 8 8 6 8 5 ...
##  $ salary               : Factor w/ 3 levels "high","low","medium": 3 3 1 3 3 3 3 2 3 2 ...
```

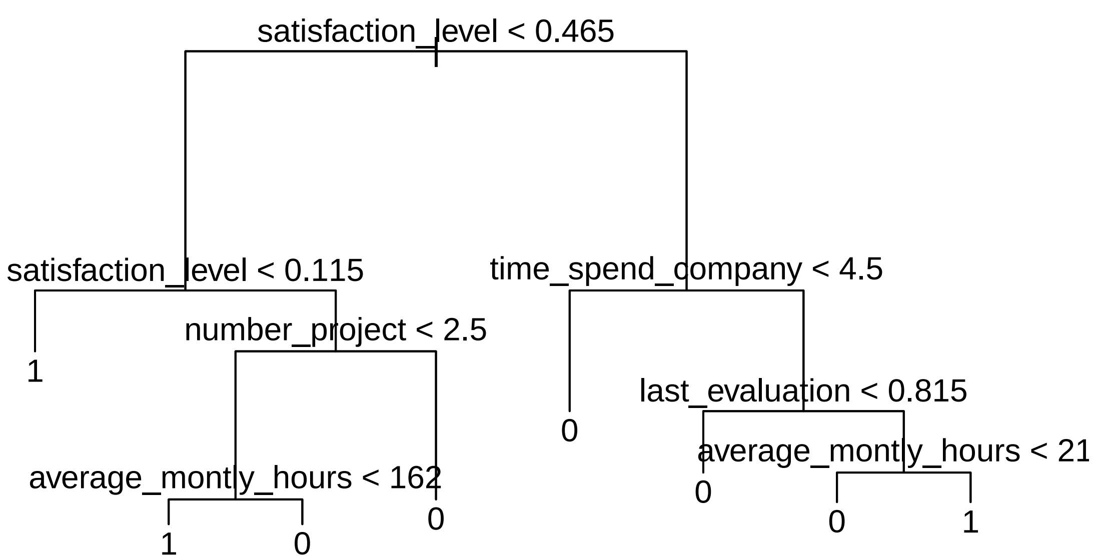
```
md$left <- as.factor(md$left)
```

Making of DT Model

```
tree.hr=tree(left~.,data=md)
tree.hr
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 10499 12700.00 0 ( 0.70711 0.29289 )
##    2) satisfaction_level < 0.465 2910  3940.00 1 ( 0.41031 0.58969 )
##      4) satisfaction_level < 0.115 632   368.90 1 ( 0.08544 0.91456 ) *
##      5) satisfaction_level > 0.115 2278  3158.00 0 ( 0.50044 0.49956 )
##       10) number_project < 2.5 1199  1192.00 1 ( 0.19766 0.80234 )
##         20) average_montly_hours < 162 1115   945.20 1 ( 0.15067 0.84933 ) *
##         21) average_montly_hours > 162 84    78.83 0 ( 0.82143 0.17857 ) *
##       11) number_project > 2.5 1079   959.90 0 ( 0.83689 0.16311 ) *
##    3) satisfaction_level > 0.465 7589  7133.00 0 ( 0.82093 0.17907 )
##      6) time_spend_company < 4.5 6199  4392.00 0 ( 0.88627 0.11373 ) *
##      7) time_spend_company > 4.5 1390  1922.00 0 ( 0.52950 0.47050 )
##       14) last_evaluation < 0.815 550   456.20 0 ( 0.85455 0.14545 ) *
##       15) last_evaluation > 0.815 840  1049.00 1 ( 0.31667 0.68333 )
##         30) average_montly_hours < 214 147   137.20 0 ( 0.82313 0.17687 ) *
##         31) average_montly_hours > 214 693   710.90 1 ( 0.20924 0.79076 ) *
```

```
plot(tree.hr)
text(tree.hr,pretty=0)
```



Summary of DT Model

```
summary(tree.hr)
```

```
##
## Classification tree:
## tree(formula = left ~ ., data = md)
## Variables actually used in tree construction:
## [1] "satisfaction_level"   "number_project"       "average_montly_hours"
## [4] "time_spend_company"   "last_evaluation"
## Number of terminal nodes:  8
## Residual mean deviance:  0.7672 = 8049 / 10490
## Misclassification error rate: 0.1304 = 1369 / 10499
```

Here the misclassification error is only 13.04% with 8 terminal nodes.

```
tree.pred=predict(tree.hr,newdata=md,type="class")
table(tree.pred, md$left)
```

```
##
## tree.pred    0    1
##         0 7057 1002
##         1  367 2073
```

Prediction for Test Data:

```
tree.pred=predict(tree.hr,test.md,type="class")
table(tree.pred)
```

```
## tree.pred
##    0    1
## 3448 1052
```

Through testing data the prediction of our DT model shows that out of 4500 employees, 1052 employees

may leave next.

Random Forest Model

```
class_hr=randomForest(left~.,data=md)
class_hr
```

```
##
## Call:
##  randomForest(formula = left ~ ., data = md)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
```

```
##         OOB estimate of  error rate: 12.58%
## Confusion matrix:
##      0    1 class.error
## 0 7047  377  0.05078125
## 1  944 2131  0.30699187
```

The RF Model shows the misclassification error 12.58% only.

Prediction for Test Data:

```
forest.pred=predict(class_hr,newdata=test.md)
table(forest.pred)
```

```
## forest.pred
##    0    1
## 3421 1079
```

Through testing data the prediction of our RF model shows that out of 4500 employees, 1078 employees

may leave next.

Recommandation:

The RF Model shows the misclassification error 12.58% only which is less than both LR Model (31.18%)

and DT Model (13.69%)

So the RF model is recommended for HR Analysis.

NOTE:

Still there is scope of optimization of models there, Here in this POC we have not applied.