Project VI: GAM, MARS, and PPR

Owusu Noah

09 26, 2023

Contents

1 Question 1 - Data Preparation

1.1 Importing the data

```
library(dplyr)
hr <- "HR_comma_sep.csv" %>%
  read.csv()

head(hr); dim(hr)
```

```
##   satisfaction_level last_evaluation number_project average_montly_hours
## 1               0.38            0.53              2                  157
## 2               0.80            0.86              5                  262
## 3               0.11            0.88              7                  272
## 4               0.72            0.87              5                  223
## 5               0.37            0.52              2                  159
## 6               0.41            0.50              2                  153
##   time_spend_company Work_accident left promotion_last_5years sales salary
## 1                  3             0    1                     0 sales    low
## 2                  6             0    1                     0 sales medium
## 3                  4             0    1                     0 sales medium
## 4                  5             0    1                     0 sales    low
## 5                  3             0    1                     0 sales    low
## 6                  3             0    1                     0 sales    low
```

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

Comment:

The **HR** data has 14999 observations and 10 variables, namely **satisfaction_level** ,**number of project** etc.

1.1.1 Changing the categorical variable *salary* to ordinal

```
hr$salary <- factor(hr$salary,
        levels=c("low", "medium","high"), ordered=TRUE)

class(hr$salary)
```

```
## [1] "ordered" "factor"
```

1.1.2 Changing the column name of variable sales to department

```r
colnames(hr)[colnames(hr) == 'sales'] <- 'department'

head(hr, 2)
```

```
##   satisfaction_level last_evaluation number_project average_montly_hours
## 1               0.38            0.53              2                  157
## 2               0.80            0.86              5                  262
##   time_spend_company Work_accident left promotion_last_5years department salary
## 1                  3             0    1                     0      sales    low
## 2                  6             0    1                     0      sales medium
```

1.1.3 Converting the target variable ”left” to categorical variable

```r
hr$left <- factor(hr$left, levels = c(0,1), labels = c("stayed", "left"))
class(hr$left)
```

```
## [1] "factor"
```

Comment:

Yes! the target variable **left** is now a categorical variable.

1.1.4 Inspecting missing values

```r
# Listing the missing rate for each variable.
miss.info <- function(dat, filename=NULL){
  vnames <- colnames(dat); vnames
  n <- nrow(dat)
  out <- NULL
  for (j in 1: ncol(dat)){
    vname <- colnames(dat)[j]
    x <- as.vector(dat[,j])
    n1 <- sum(is.na(x), na.rm=T)
    n2 <- sum(x=="NA", na.rm=T)
    n3 <- sum(x=="", na.rm=T)
    nmiss <- n1 + n2 + n3
    ncomplete <- n-nmiss
    out <- rbind(out, c(col.number=j, vname=vname,
                        mode=mode(x), n.levels=length(unique(x)),
                        ncomplete=ncomplete, miss.perc=nmiss/n))
  }
  out <- as.data.frame(out)
  row.names(out) <- NULL
  if (!is.null(filename)) write.csv(out, file = filename, row.names=F)
  return(out)
}
df <- knitr::kable(miss.info(hr), booktabs = T, format = "markdown")
kableExtra::kable_styling(df, bootstrap_options = "striped", full_width = F)
```

col.number	vname	mode	n.levels	ncomplete	miss.perc
1	satisfaction_level	numeric	92	14999	0
2	last_evaluation	numeric	65	14999	0
3	number_project	numeric	6	14999	0
4	average_montly_hours	numeric	215	14999	0
5	time_spend_company	numeric	8	14999	0
6	Work_accident	numeric	2	14999	0
7	left	character	2	14999	0
8	promotion_last_5years	numeric	2	14999	0
9	department	character	10	14999	0
10	salary	character	3	14999	0

Comment:

From the above table, it is clear that the **hr** dataset have no missing value.This is a desired outcome to guarantee a reliable conclusion and practicable subsequent analysis.

2 Question 2 - Exploratory Data Analysis

2.1 Checking for variable type

```
glimpse(hr)
```

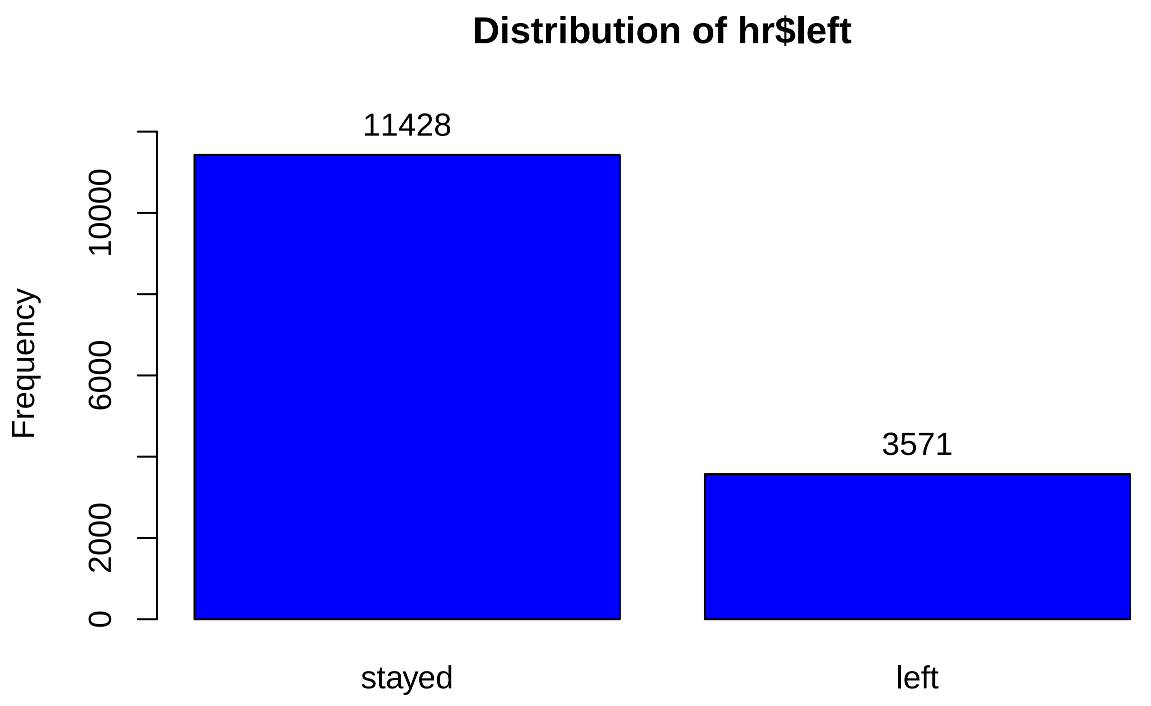
```
## Rows: 14,999
## Columns: 10
## $ satisfaction_level    <dbl> 0.38, 0.80, 0.11, 0.72, 0.37, 0.41, 0.10, 0.92, ~
## $ last_evaluation       <dbl> 0.53, 0.86, 0.88, 0.87, 0.52, 0.50, 0.77, 0.85, ~
## $ number_project        <int> 2, 5, 7, 5, 2, 2, 6, 5, 5, 2, 2, 6, 4, 2, 2, 2, ~
## $ average_montly_hours  <int> 157, 262, 272, 223, 159, 153, 247, 259, 224, 142~
## $ time_spend_company    <int> 3, 6, 4, 5, 3, 3, 4, 5, 5, 3, 3, 4, 5, 3, 3, 3, ~
## $ Work_accident         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ left                  <fct> left, left, left, left, left, left, left, left, ~
## $ promotion_last_5years <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ department            <chr> "sales", "sales", "sales", "sales", "sales", "sa~
## $ salary                <ord> low, medium, medium, low, low, low, low, low, lo~
```

Comment:

From the above output, variables **satisfaction_level** and **last_evaluation** are numeric(continuous) variable, **salary** is categorical (ordinal) variable , and **sales and left** are categorical (nominal) variable. The remaining variables are numeric(integer) variable.

2.2 Frequency distribution of target variable

```
#Inspecting frequency distribution of the target variable (left)
tab1(hr$left, decimal = 2, cum.percent = F, xlab = "Left the company", ylab = "Frequency", col = "blu
```



```
## hr$left :
##         Frequency Percent
## stayed      11428   76.19
## left         3571   23.81
##   Total     14999  100.00
```

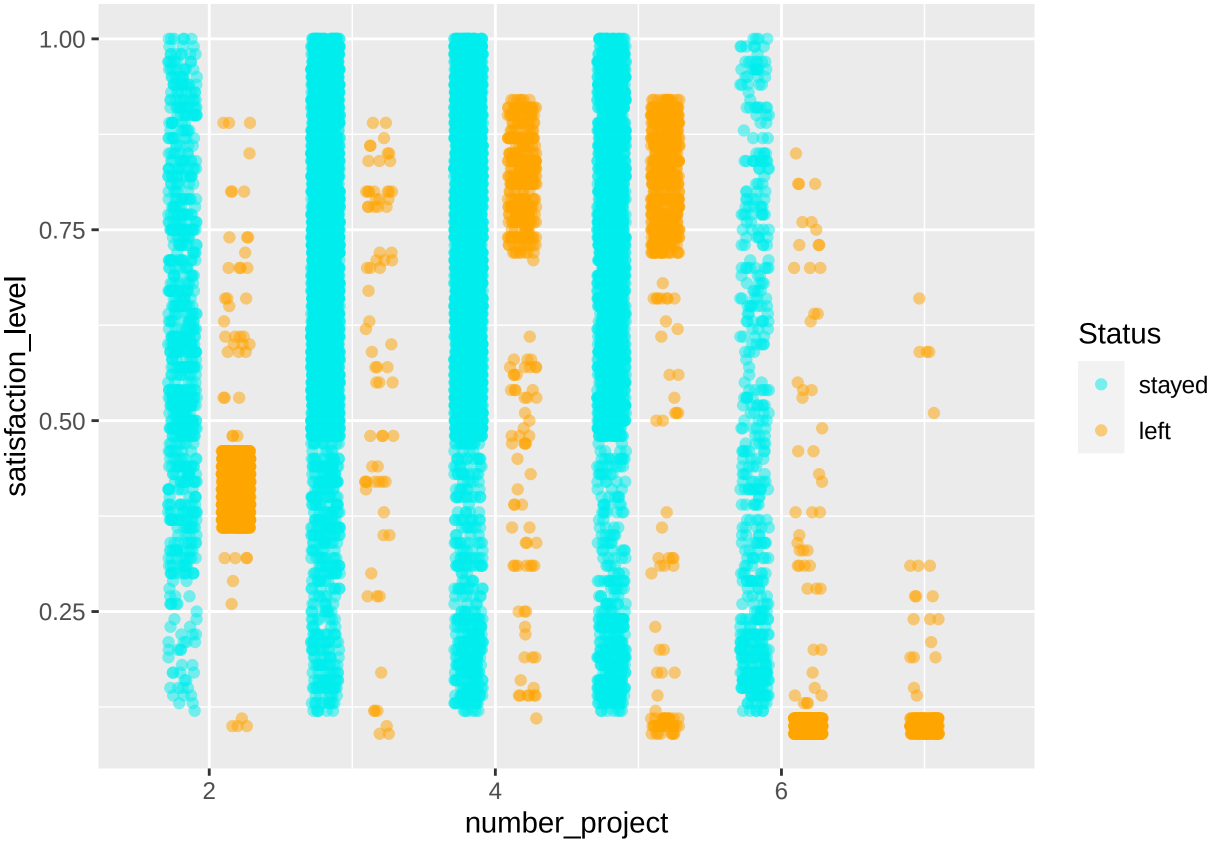
Comment:

The frequency table revealed 11428(76.19%) **stayed** counts and 3571(23.81%) **left** counts for the **left** variable. This classification case is neither completely balanced nor unbalanced. However, we will consider this scenario as a balanced classification and continue with our analysis since the proportion of employees who left is greater than 5% (which is the cutoff proportion for imbalanced classification).

(2a)

2.3 Scatterplot of employees Satisfaction level vrs Number of project

```
ggplot(data = hr, aes(x= number_project, color = left))+
  geom_point(position=position_jitterdodge(),alpha=.5,
             aes (y =   satisfaction_level), bins = 2) +
  scale_color_manual("Status", values = c("cyan2", "orange"))
```



```
   labs(title =
"Employee's Satisfaction level vrs Number of projects (Left the Comapny - stayed/left)",
           x = "Number of projects", y = "Satisfaction level")+
            theme(plot.title = element_text(size=9),
                   axis.text.x = element_text(size=6) )
```

```
## NULL
```

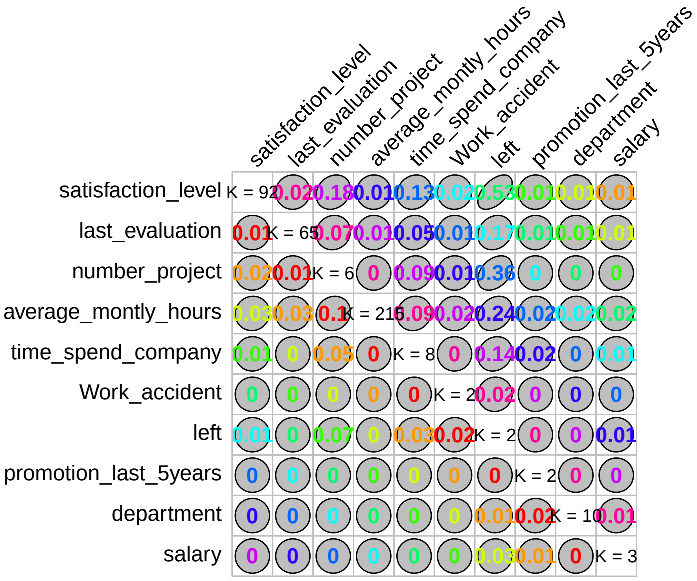
Comment:

From the plot above, we can see that employees who are satisfied with their work are less willing to leave. Thus, the more satisfied employees are, the less willingly they are to leave. However, the interesting thing is that we can find that not all employees with low-paying and unsatisfactory jobs left the company. Most of them continued working with the company. Finding these employees and understanding why they are unwilling to leave can provide valuable information to the HR department. Moreover, Among employees who left, had **2 number_project** were not satisfied since majority of them had satisfaction level below 0.50

(2b)

2.4 Correlation matrix among the variables in the hr dataset

```
library(GoodmanKruskal)
mat.hr <- GKtauDataframe(hr)
plot(mat.hr, dgts = 2, diagSize = 0.8)
```



Comment:

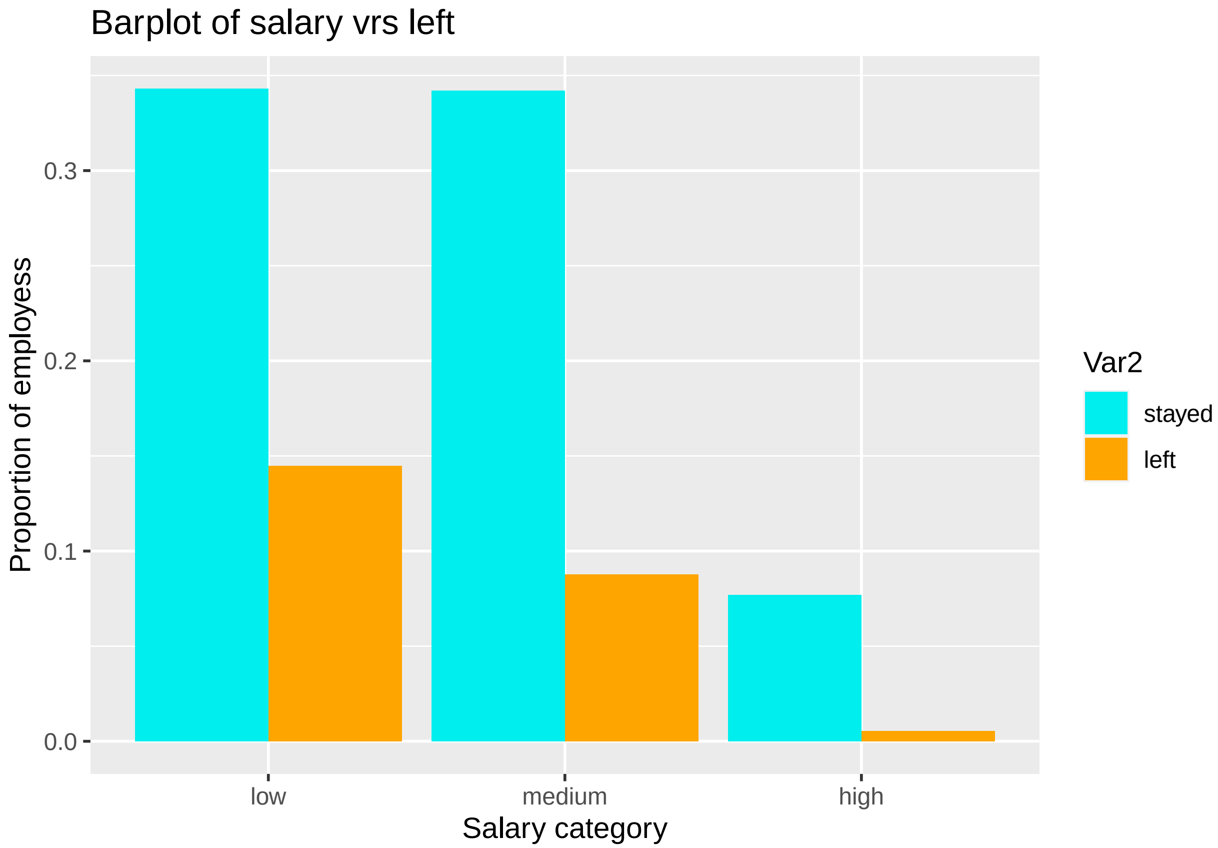
After carefully examining the data, The variables **satisfaction_level, last_evaluation, number_project** and **average_monthly_hours** exhibit slight forward associations with the target variable **left**, the reverse associations are much smaller.

The variable **satisfaction_level** exhibits a slight ability to explain variations in the other variables (ranging from **0.01** to **0.53**), the reverse associations are much smaller: the τ value from **satisfaction_level** to **left** is **0.53**,indicating quite a strong association while that from **satisfaction_level** to **number_project** is **0.18**.

2.5 Interesting finding 1

Salary VRS. Employee turnover

```
t1 <- table(hr$salary,hr$left)
df <- as.data.frame(prop.table(t1))

#round(prop.table(t1), 3)

ggplot(df, aes(x=Var1,y=Freq,fill=Var2)) +
  geom_bar(position="dodge",stat='identity')+
  ggtitle("Barplot of salary vrs left")+
  xlab("Salary category")+ ylab("Proportion of employess")+
  scale_fill_manual(values = c("cyan2","orange"))
```



Comment: From the above results, the data indicates that employees who left the company tend to have lower salaries when compared to employees who do not.

Among the employees who left the company, majority received low and medium salaries.

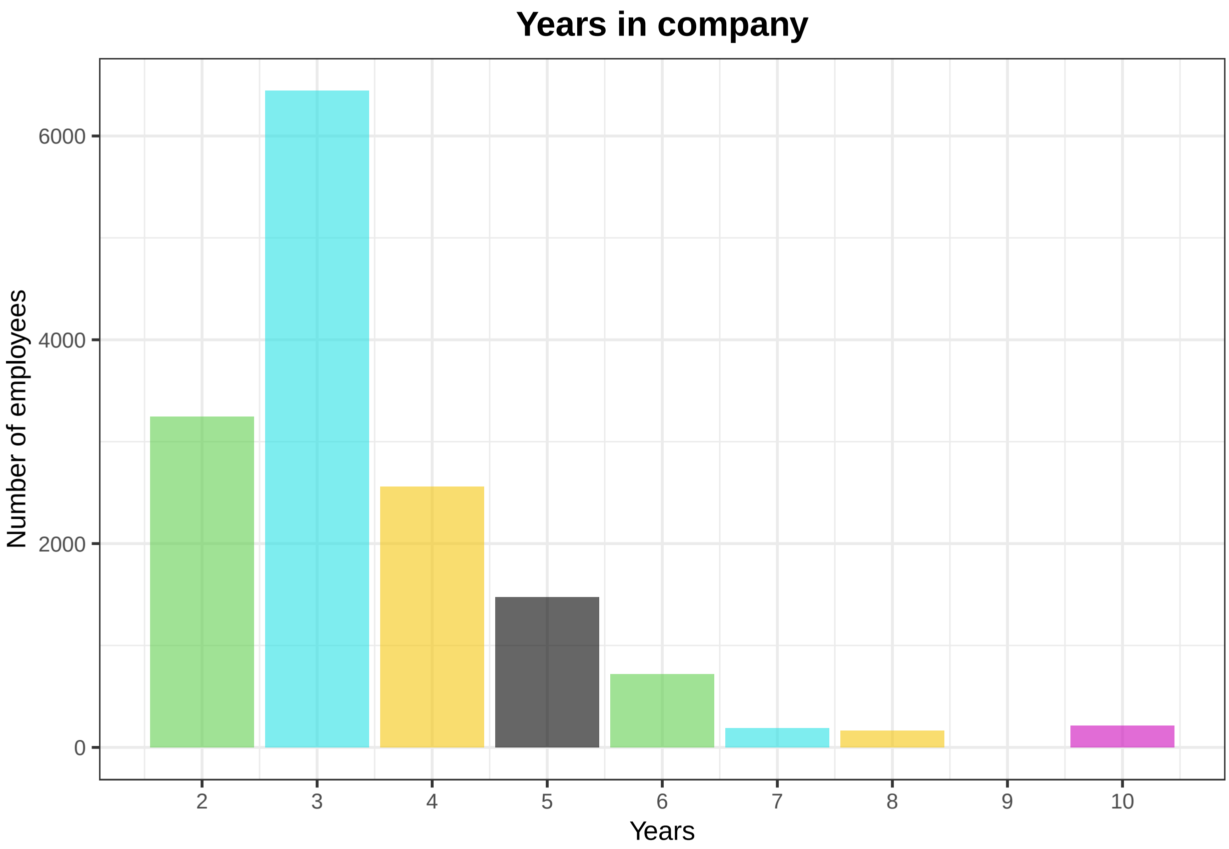
2.6 Interesting finding 2

The frequency distribution of time spent by employees in the company were first explored and a graphical representation drawn after using a barplot.

```
table(hr$time_spend_company)
```

```
##
##    2    3    4    5    6    7    8   10
## 3244 6443 2557 1473  718  188  162  214
```

```
ggplot(data = hr, aes(x = time_spend_company, y = ..count..))+
  geom_bar(fill = c(3,5,7,9,11,13,15,6),
                   alpha = 0.6)+
  scale_x_continuous(breaks= seq(0,10,1))+
  theme_bw()+
  ggtitle("Years in company")+
  theme(plot.title = element_text(hjust = 0.5))+
  labs(x = "Years", y = "Number of employees")+
  theme(plot.title = element_text(size = 13,face = "bold"),
              text = element_text(size = 10))
```



Comment:

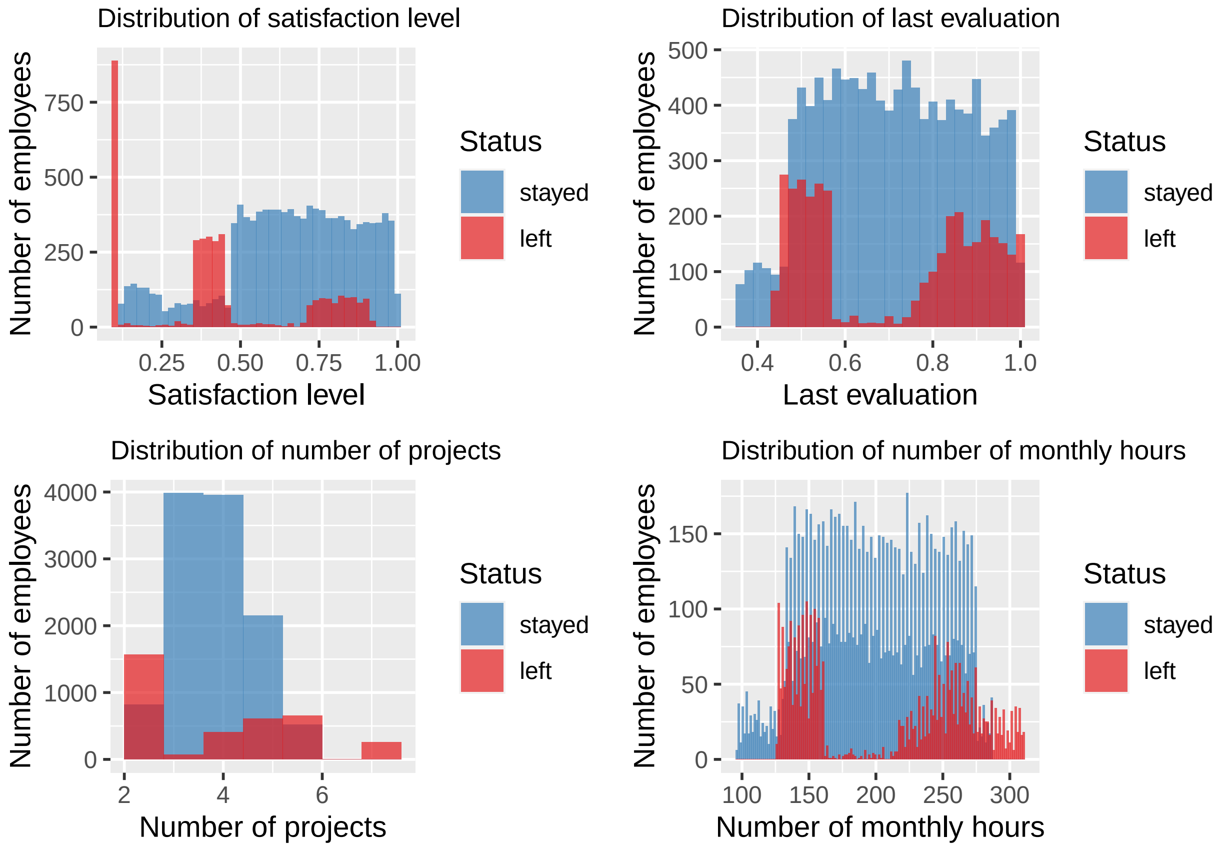
The range of time spent by an employee in the company is *2* to *10* years with *no* employee working for *9* years. It can also be seen from the above figure that over 6000 employees in the company spent *three years* whilst a few employees spent *eight years* with the company.

2.7 Interesting finding 3

Distribution of features grouped by the target variable

```
OverlayedHist <- function(mData, featureVar, grouper, mbinwidth, mTitle,  mxlab,
                          mylab, mlegendTitle){

  p <- ggplot(hr, aes(eval(parse(text = featureVar)), fill = eval(parse(text = grouper))))+
    geom_histogram(alpha = 0.7, position = 'identity', binwidth = mbinwidth) +
    scale_fill_manual(mlegendTitle, values=c("#377EB8","#E41A1C")) +
    ggtitle(mTitle) +xlab(mxlab) + ylab(mylab) +
    theme(plot.title = element_text(size=10))


  return(p)
}
```

```
p1 <- OverlayedHist(mData = hr, featureVar = "satisfaction_level",
                    grouper = "left", mbinwidth    =  0.02,
                    mTitle = "Distribution of satisfaction level",
                    mxlab  = "Satisfaction level", mylab = "Number of employees",
                    mlegendTitle = "Status")

p2 <- OverlayedHist(mData = hr, featureVar  = "last_evaluation",
                    grouper = "left", mbinwidth    =  0.02,
                    mTitle  = "Distribution of last evaluation",
                    mxlab   = "Last evaluation", mylab  = "Number of employees",
                    mlegendTitle = "Status")

p3 <- OverlayedHist(mData = hr,featureVar   = "number_project",grouper = "left",
                    mbinwidth = 0.8, mTitle= "Distribution of number of projects",
                    mxlab = "Number of projects", mylab = "Number of employees",
                    mlegendTitle = "Status")

p4 <- OverlayedHist(mData = hr,featureVar   = "average_montly_hours",
                    grouper = "left",mbinwidth    =  1.5,
                    mTitle ="Distribution of number of monthly hours",
                    mxlab = "Number of monthly hours",mylab= "Number of employees",
                    mlegendTitle = "Status")
ggarrange(p1,p2,p3,p4, ncol = 2, nrow = 2)
```



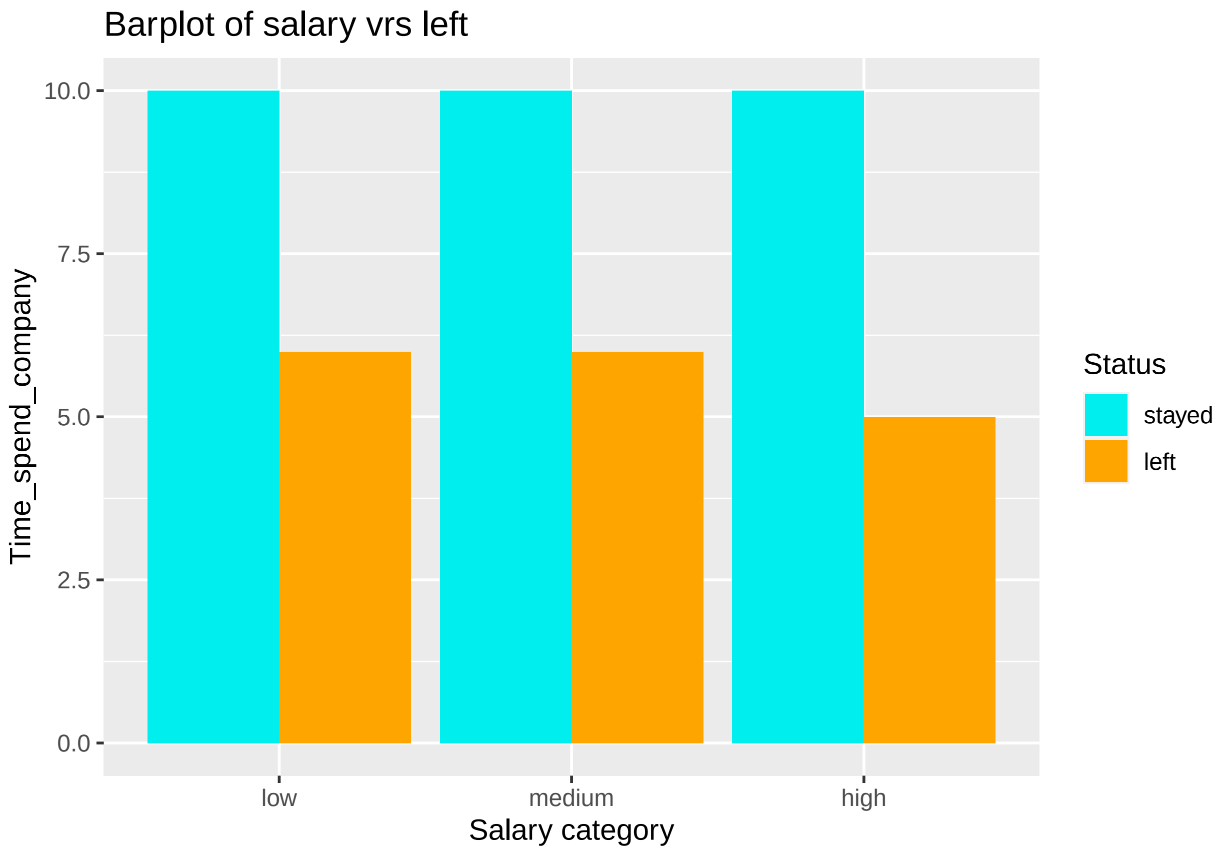
Comment:

From the plot, it seems that employees who quit their role are less satisfied than those who remain loyal to their company. Employees' last evaluation seems to follow a bi modal pattern. There are employees who left, performed really well and had evaluation score over 0.75 while there is another group who was under performing with an evaluation score less than 0.55. The same pattern is observed also in the distribution of monthly hours (bottom right plot) where there are two group of employees who quit. Those who put extra effort and those who worked significantly less than the average number of monthly hours.

Interesting finding 4

Relationship between department and salary

```
ggplot(hr, aes(x=salary,y= time_spend_company, fill  = left)) +
  geom_bar(position="dodge",stat='identity')+
  ggtitle("Barplot of salary vrs left")+
  xlab("Salary category")+ ylab("Time_spend_company")+
  scale_fill_manual("Status",values = c("cyan2","orange"))
```

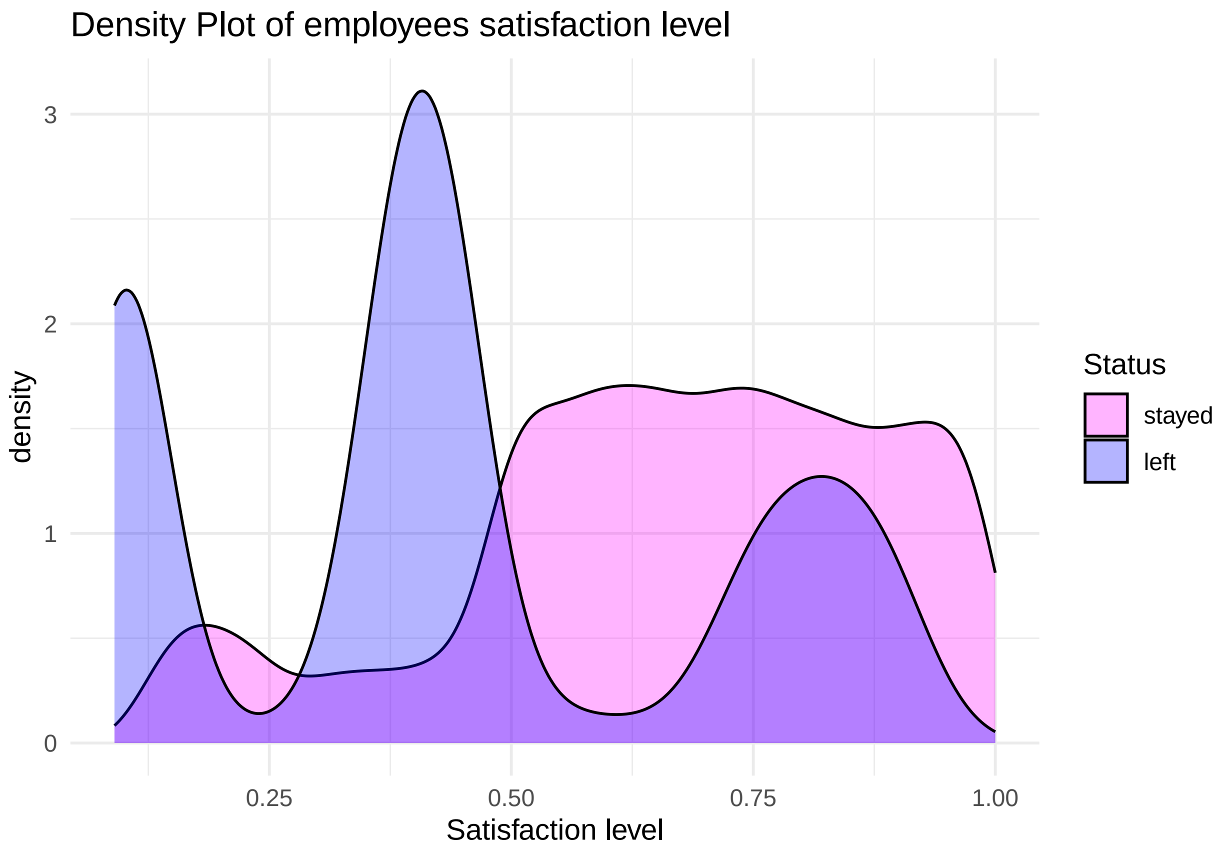


Comment:

From the just above plot, it seems that majority of the employees who left spent less that seven years in the company as well as being in the low and medium salary category. However, a certain proportion of the employees who left the company received high salary.

Density plot of employees satisfaction level

```
df.sat <- data.frame(hr$satisfaction_level, hr$left)
colnames(df.sat) <- c("satisfaction_level", "Status")
ggplot(df.sat, aes(x = satisfaction_level, fill = Status)) +
geom_density(aes(satisfaction_level), alpha = 0.3) + xlab("Satisfaction level") +
scale_fill_manual(values = c("magenta", "blue")) +
ggtitle("Density Plot of employees satisfaction level") +
theme_minimal()
```



Comment:

Among the employees who left, the distribution of their satisfaction level seems trimodal, where majority of them had satisfaction level within $0.25 - 0.50$.

3 Question 3 - Data Partitioning

```
set.seed(120) ## the set is to make the partition reproducible
train <- sample(nrow(hr), (2.0/3.0)*nrow(hr), replace = FALSE)
D1 <- hr[train, ] # training set
D2 <- hr[-train, ] # testing set

dim(D1); dim(D2)
```

```
## [1] 9999   10
```

```
## [1] 5000   10
```

Comment:

For the train and test data, the data were divided at random into two groups with ratios of $2:1$. The train data set had 9999 observations whilst the test set had 5000 observations for each of the 10 variables.

4 Question 4 - Logistic Regression

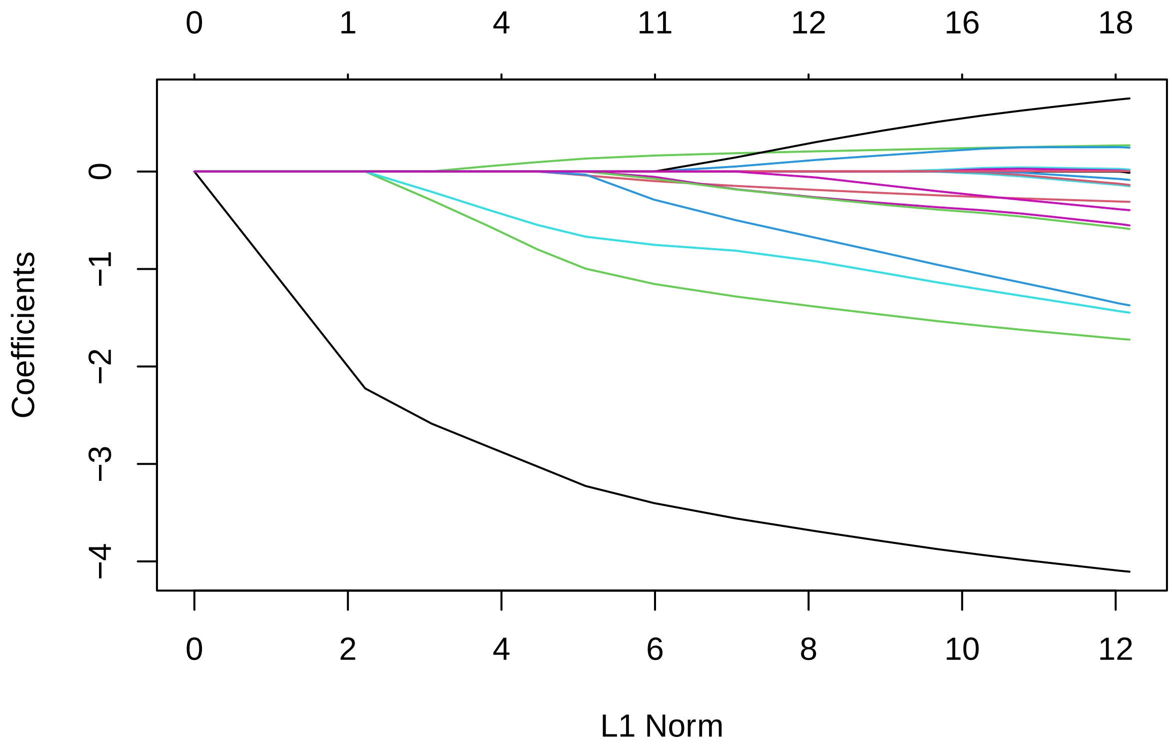
4.1 Fitting the Logistic regression model using LASSO as regularization technique

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```
X <- model.matrix(left ~ satisfaction_level + number_project +  time_spend_company +
factor(department) + last_evaluation +  average_montly_hours + Work_accident + promotion_last_5years
y <- D1$left

fit.lasso <- glmnet(x=X, y=y, family="binomial", alpha=1, nfolds = 10,
    lambda.min = 1e-4, nlambda = 30, standardize=T, thresh = 1e-07,
    maxit=1000)
plot(fit.lasso)
```



Comment:

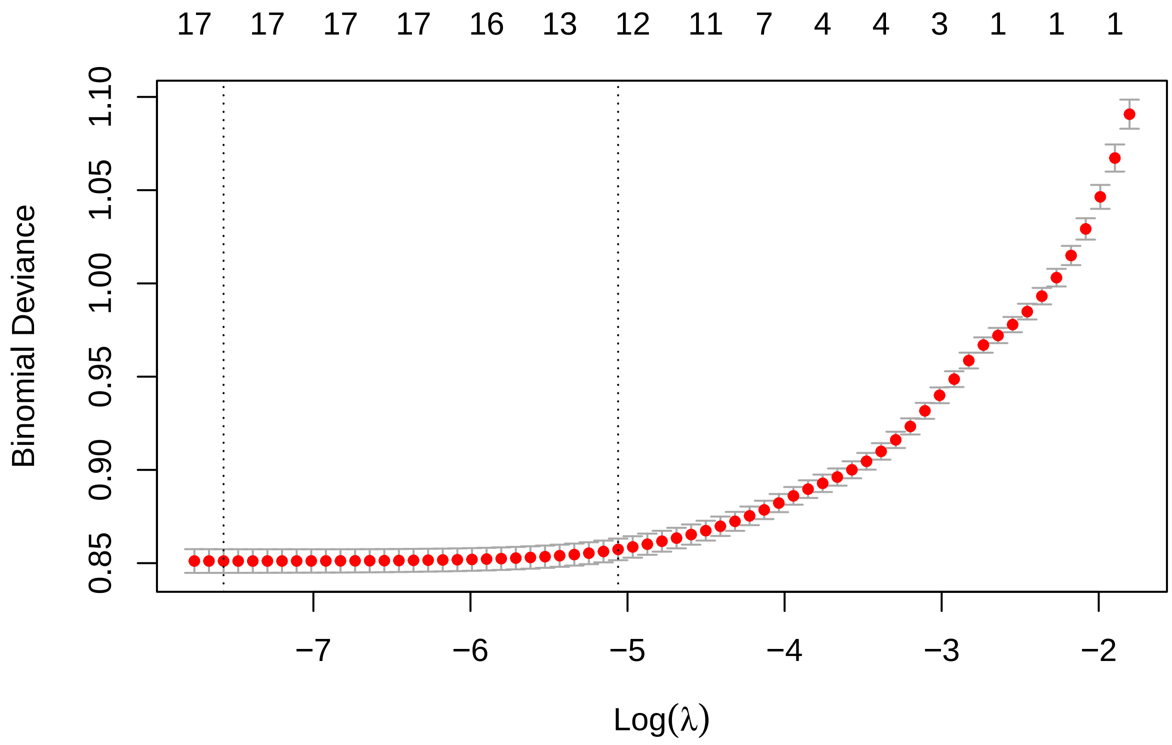
From above, the glmnet function acts as the elastic net regularization penalty, with the alpha value controlling it. A value of 1 indicates lasso, while a value of 0 indicates Ridge. A value of 1 was applied because lasso was chosen as the regularization penalty. The family specifies the kind of response variable, which in this case is binary. The number of lambda values was limited to thirty. Furthermore, the convergence threshold for coordinate descent was set to the default value. Finally, for all lambda values, the maximum number of passes over the data was set to 1000.

4.2 Choosing the best tuning parameter by Cross-validation

```
CV_model <- cv.glmnet(x=X, y=y, family="binomial", alpha = 1,
    lambda.min = 1e-4, thresh = 1e-07, type.measure = "deviance",
    maxit=1000)
CV_model
```

```
##
## Call:  cv.glmnet(x = X, y = y, type.measure = "deviance", family = "binomial",      alpha = 1, lam
##
## Measure: Binomial Deviance
##
##       Lambda Index Measure       SE Nonzero
## min 0.000515    63   0.8511 0.006346      17
## 1se 0.006346    36   0.8574 0.005799      12
```

```
plot(CV_model)
```



```
best_lambda <- CV_model$lambda.1se; best_lambda
```

```
## [1] 0.006345822
```

Comment:

The best lambda value is the largest value of lambda such that error is within 1 standard error of the minimum and it turns out to be 0.0063. The criteria used to select the tuning parameter is the 1*se* rule.

4.3 Fitting the model based on the best tuning parameter

```
fit.lasso <- glmnet(x=X, y=y, family="binomial", alpha = 1,
                    lambda=best_lambda, thresh = 1e-07,
                    maxit=1000)
fit.lasso$beta
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                                       s0
## (Intercept)                   .
## satisfaction_level           -3.716910763
## number_project               -0.197276634
## time_spend_company            0.211340861
## factor(department)hr          0.132170124
## factor(department)IT            .
## factor(department)management  -0.280623557
## factor(department)marketing     .
## factor(department)product_mng   .
## factor(department)RandD       -0.290449673
## factor(department)sales         .
## factor(department)support       .
## factor(department)technical     .
## last_evaluation                0.334559622
## average_montly_hours           0.003028576
## Work_accident                 -1.408681417
## promotion_last_5years         -0.720384720
## salary.L                      -0.952929102
## salary.Q                      -0.081955173
```

Comment:

No coefficient is shown for some predictors, because the lasso regression shrunk the coefficient all the way to zero. This means it was completely dropped from the model because it wasn't influential enough. Hence, with the law of parsimony, the model with 12 variables is the chosen model.

Fitting model with glm()

```
fit.glm <- glm(left ~ satisfaction_level + number_project   + time_spend_company +
department + last_evaluation +  average_montly_hours + Work_accident + promotion_last_5years + salary
family = binomial, data=D1)
summary(fit.glm)
```

```
##
## Call:
## glm(formula = left ~ satisfaction_level + number_project + time_spend_company +
##     department + last_evaluation + average_montly_hours + Work_accident +
##     promotion_last_5years + salary, family = binomial, data = D1)
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -0.4310309  0.1912214  -2.254 0.024190 *
## satisfaction_level   -4.1164802  0.1204457 -34.177  < 2e-16 ***
## number_project       -0.3138632  0.0261234 -12.015  < 2e-16 ***
## time_spend_company    0.2706893  0.0194581  13.911  < 2e-16 ***
## departmenthr          0.2320951  0.1611641   1.440 0.149834
## departmentIT         -0.1721513  0.1492696  -1.153 0.248791
## departmentmanagement -0.5762948  0.2026870  -2.843 0.004465 **
## departmentmarketing  -0.0345496  0.1621797  -0.213 0.831301
## departmentproduct_mng -0.1599950  0.1590559  -1.006 0.314462
## departmentRandD      -0.6140029  0.1792541  -3.425 0.000614 ***
## departmentsales      -0.1057400  0.1253150  -0.844 0.398785
## departmentsupport     0.0058153  0.1334903   0.044 0.965252
## departmenttechnical  -0.0100434  0.1303929  -0.077 0.938604
## last_evaluation       0.7613274  0.1832892   4.154 3.27e-05 ***
## average_montly_hours  0.0045790  0.0006343   7.219 5.25e-13 ***
## Work_accident        -1.7336713  0.1174828 -14.757  < 2e-16 ***
## promotion_last_5years -1.3934340  0.3119548  -4.467 7.94e-06 ***
## salary.L              -1.4646359  0.1154581 -12.685  < 2e-16 ***
## salary.Q              -0.4064279  0.0746921  -5.441 5.29e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 10918.6  on 9998  degrees of freedom
## Residual deviance:  8463.8  on 9980  degrees of freedom
## AIC: 8501.8
##
## Number of Fisher Scoring iterations: 5
```

Comment:

Yes! We have similar results with using the standard *glm()* as using the regularized logistic regression (with LASSO as penalty function)

Obtaining the associated odds ratio and the 95% confidence intervals for the odds ratio

```
exp(cbind(OR = coef(fit.glm), confint(fit.glm)))
```

```
##                               OR      2.5 %     97.5 %
## (Intercept)           0.64983883 0.44614546 0.94428904
## satisfaction_level    0.01630179 0.01285197 0.02060857
## number_project        0.73061898 0.69399337 0.76883856
## time_spend_company    1.31086776 1.26183635 1.36188175
## departmenthr          1.26123963 0.91967977 1.73032671
## departmentIT          0.84185182 0.62856595 1.12871101
## departmentmanagement  0.56197676 0.37590669 0.83275661
## departmentmarketing   0.96604038 0.70280945 1.32760184
## departmentproduct_mng 0.85214806 0.62375019 1.16391943
## departmentRandD       0.54118024 0.37980128 0.76728870
## departmentsales       0.89965856 0.70495856 1.15241598
## departmentsupport     1.00583229 0.77536037 1.30876475
## departmenttechnical   0.99000683 0.76791301 1.28055352
## last_evaluation       2.14111636 1.49554521 3.06820783
## average_montly_hours  1.00458954 1.00334378 1.00584216
## Work_accident         0.17663474 0.13951433 0.22122399
## promotion_last_5years 0.24822144 0.12870740 0.44118732
## salary.L              0.23116214 0.18294230 0.28790543
## salary.Q              0.66602514 0.57335870 0.76875648
```

Comment:

From the above output, all the variables whose confidence interval does not include 1 are significant. In particular, variables **satisfaction_level, last_evaluation, number_project, time_spend_company, work_accident, promotion_last_5years** and **average_monthly_hours** are all significant in explaining the variation in the log(odds) of an employee leaving the company.

4.4 Evaluating the model using the test data

```
X.test <- model.matrix (left ~ satisfaction_level + number_project +
                  time_spend_company +factor(department) + last_evaluation +
                  average_montly_hours + Work_accident + promotion_last_5years +
                  salary, data = D2)

pred.LASSO <- predict(fit.lasso, newx = X.test, s=best_lambda, type="response")
```

4.5 Plotting the ROC or AUC for the Logistic regression

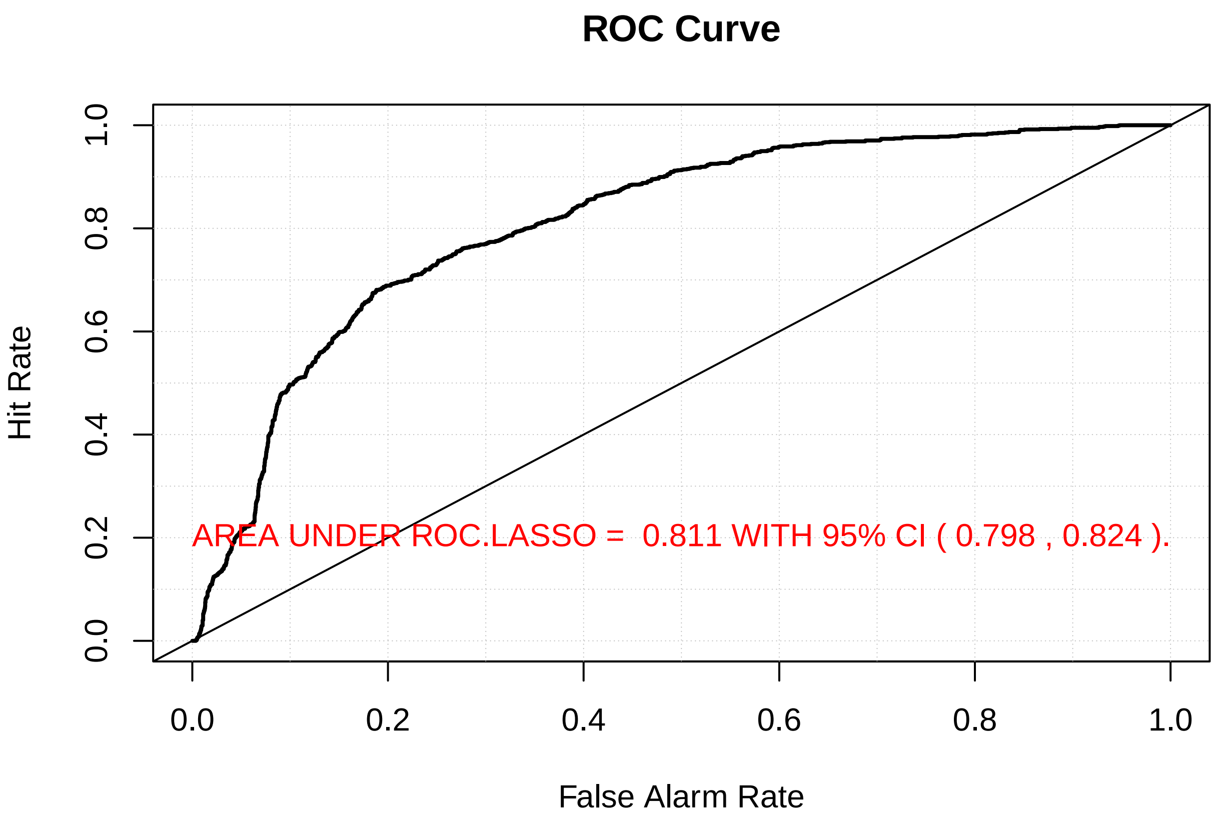
```
library(verification)
library(cvAUC)
D2$left <- ifelse(D2$left == "stayed", 0,1)
yobs <- D2$left
AUC.LASSO <- ci.cvAUC(as.vector(pred.LASSO), labels=yobs, folds=1:NROW(D2),
                confidence=0.95); AUC.LASSO
```

```
## $cvAUC
## [1] 0.8109612
##
## $se
## [1] 0.00674721
##
## $ci
## [1] 0.7977369 0.8241855
##
## $confidence
## [1] 0.95
```

```
mod.LASSO <- verify(obs = yobs, pred = as.vector(pred.LASSO))
```

```
## If baseline is not included, baseline values  will be calculated from the  sample obs.
```

```
## If baseline is not included, baseline values will be calculated from the sample obs.

roc.plot(mod.LASSO, plot.thres = NULL, col="darkblue")
text(x=0.50, y=0.2, paste("AREA UNDER ROC.LASSO = ",
round(AUC.LASSO$cvAUC, digits = 3),"WITH 95% CI (",
round(AUC.LASSO$ci[1],3),",",round(AUC.LASSO$ci[2],3), ").", sep=" "),
col="red", cex=1)
```



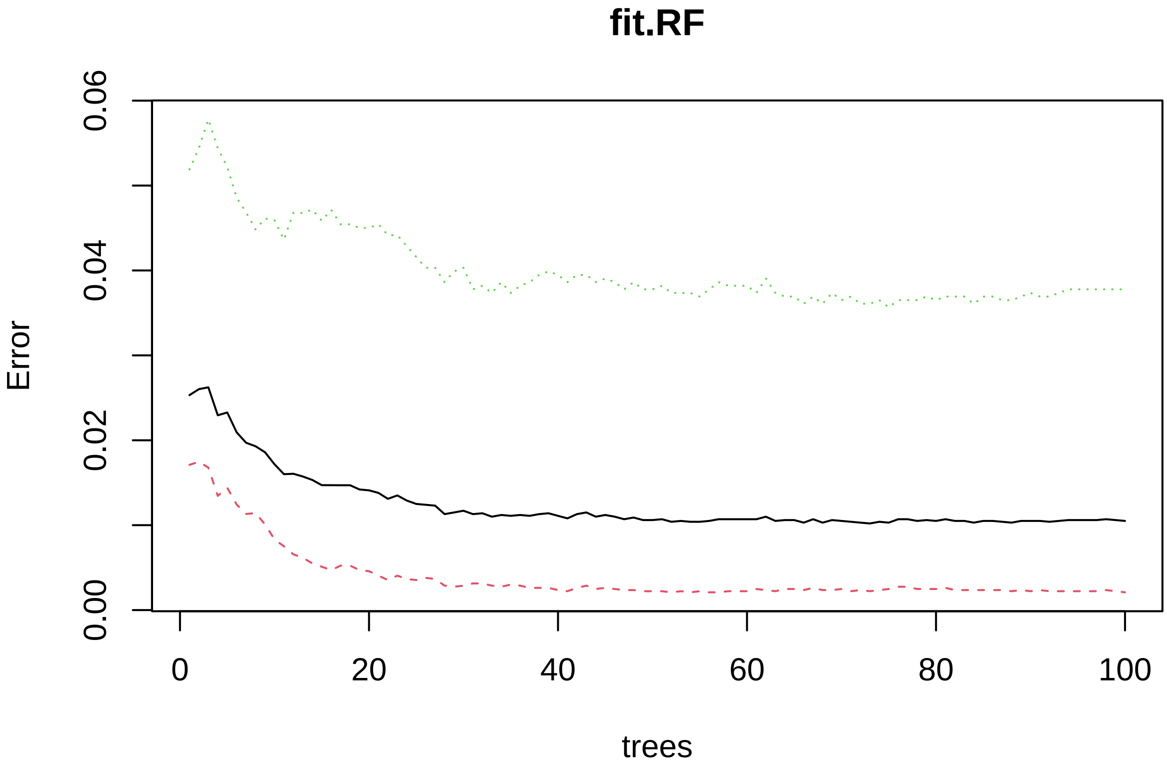
Comment:

The AUC from the LASSO technique is 0.81 which is high and it tells us that the model is a good fit and has good prediction accuracy.From the output, it can be concluded from the confidence interval that the model shows good discrimination.

5 QUESTION 5 - Random Forest

```
library(randomForest)
fit.RF <- randomForest(left ~., data=D1,importance=TRUE, proximity=TRUE, ntree=100)
fit.RF; plot(fit.RF)
```

```
##
## Call:
##  randomForest(formula = left ~ ., data = D1, importance = TRUE,      proximity = TRUE, ntree = 100
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 1.05%
## Confusion matrix:
##        stayed left class.error
## stayed   7627   16 0.002093419
## left       89 2267 0.037775891
```



```
yhat.RF <- predict(fit.RF, newdata=D2, type="prob")[, 2]
```

Comment:

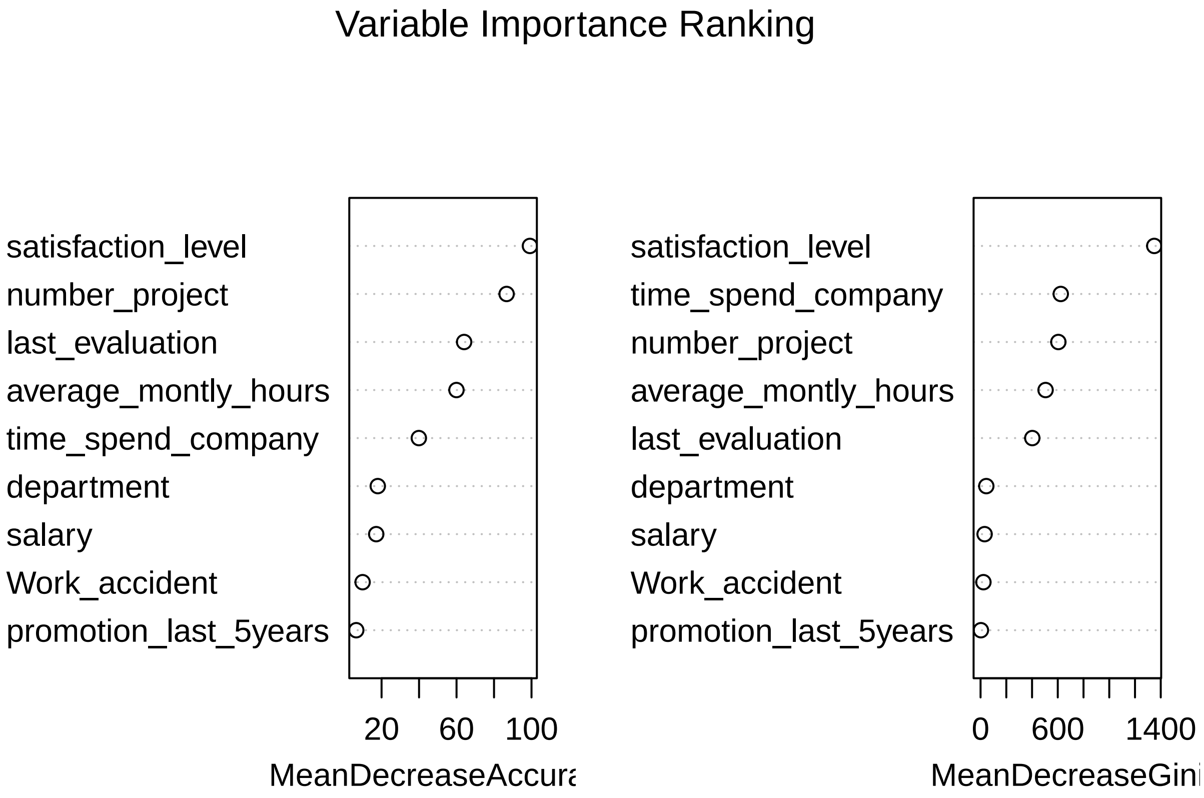
From the plot above, 100 decision trees has been built using the random forest algorithm based learning. We plotted the error rate across decision trees. The plot seems to indicate that after 20 decision trees, there is not a significant reduction in error rate.

VARIABLE IMPORTANCE RANKING

```
round(importance(fit.RF), 2)
```

```
##                         stayed   left MeanDecreaseAccuracy MeanDecreaseGini
## satisfaction_level       30.18 121.30                99.16          1349.28
## last_evaluation          10.37  64.18                64.04           402.00
## number_project           25.78  80.50                86.72           604.39
## average_montly_hours     28.69  53.11                59.96           505.16
## time_spend_company       23.94  37.66                39.89           623.07
## Work_accident             4.28   9.79                 9.88            22.31
## promotion_last_5years     3.48   5.31                 6.49             2.81
## department                5.26  27.08                17.98            44.57
## salary                    8.11  16.16                17.16            31.64
```

```
varImpPlot(fit.RF, main="Variable Importance Ranking")
```

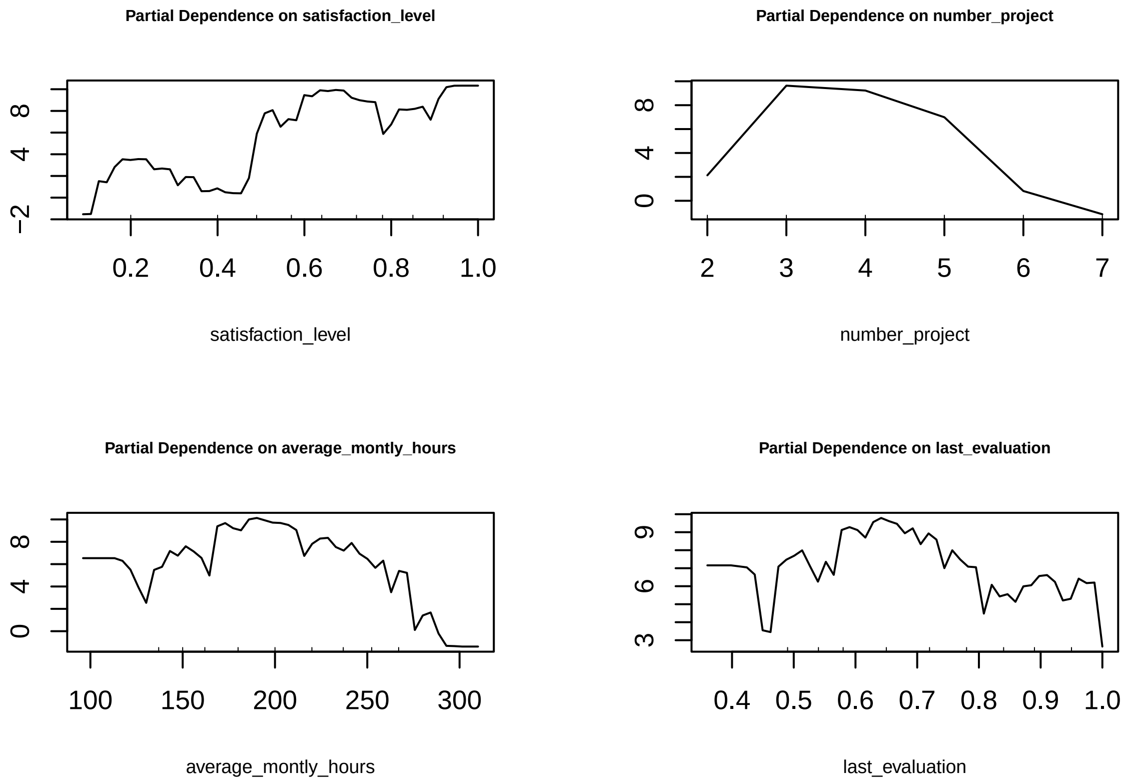


Comment:

According to the variable importance ranking for the random forest, the top four variables are **satisfaction level**, **number of project**, **last_evaluation** and **average_monthly_hours** based on the MeanDecreaseAccuracy. Promotion_last_5years is the least significant variable.

Partial independence plot

```
par(mfrow=c(2,2))
partialPlot(fit.RF, pred.data=D1, x.var=satisfaction_level,
            rug=TRUE, cex.lab=0.7, cex.main=0.6)
partialPlot(fit.RF, pred.data=D1, x.var=number_project,
            rug=TRUE, cex.lab=0.7, cex.main=0.6)
partialPlot(fit.RF, pred.data=D1, x.var=average_montly_hours,
            rug=TRUE, cex.lab=0.7, cex.main=0.6)
partialPlot(fit.RF, pred.data=D1, x.var=last_evaluation,
            rug=TRUE, cex.lab=0.7, cex.main=0.6)
```



Comment:

The considerable nonlinearity displayed in the plots above demonstrates that the logistic regression model is inadequate.

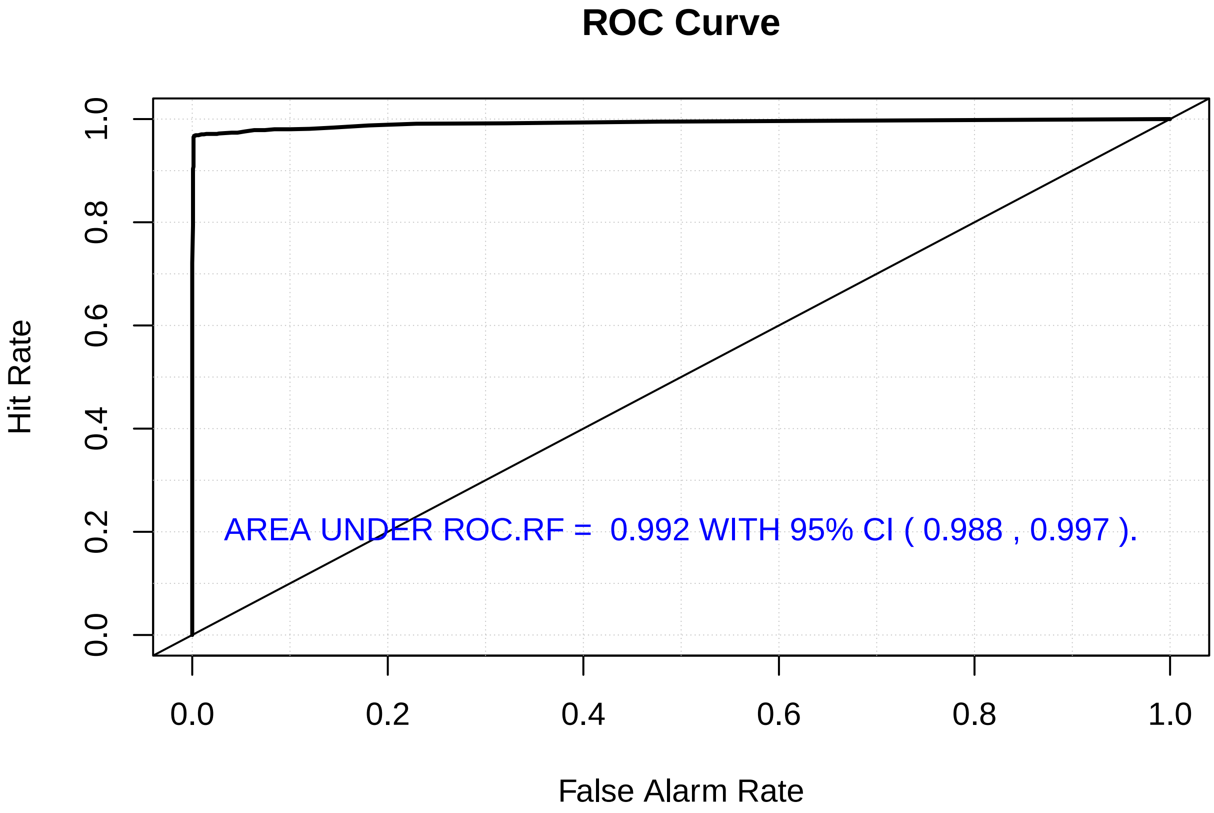
```
AUC.RF <- ci.cvAUC(yhat.RF, labels=yobs, folds=1:NROW(D2),
                   confidence=0.95); AUC.RF
```

```
## $cvAUC
## [1] 0.9923417
##
## $se
## [1] 0.00224594
##
## $ci
## [1] 0.9879397 0.9967436
##
## $confidence
## [1] 0.95
```

```
mod.RF <- verify(obs = yobs, pred = yhat.RF)
```

```
## If baseline is not included, baseline values  will be calculated from the  sample obs.
```

```
roc.plot(mod.RF, plot.thres = NULL, col="darkblue")
text(x=0.50, y=0.2, paste("AREA UNDER ROC.RF = ",
round(AUC.RF$cvAUC, digits = 3),"WITH 95% CI (",
round(AUC.RF$ci[1],3),",",round(AUC.RF$ci[2],3), ").", sep=" "),
col="blue", cex=1)
```



Comment:

The AUC from the Random Forest is 0.99 which is high and it tells us that the model is a good fit and has good prediction accuracy.From the output, it can be concluded from the confidence interval that the model shows good discrimination.

6 QUESTION 6 - Generalized Additive Model (GAM)

6.1 Fitting the model and summary

```
# install.packages("gam")
library(gam)
gam.fit1 <- gam(left ~ s(satisfaction_level) + s(last_evaluation) + number_project
                + lo(satisfaction_level,last_evaluation) +average_montly_hours
                + s(time_spend_company) + Work_accident +promotion_last_5years +
                  salary, data=D1, na=na.gam.replace,
                control = gam.control(epsilon=1e-04, bf.epsilon = 1e-04,
                                      maxit=50,bf.maxit = 50), family="binomial")

gam.fit2 <- gam(left ~ lo(satisfaction_level) + s(last_evaluation) + lo(number_project)
                + lo(average_montly_hours) + time_spend_company + Work_accident +
                  promotion_last_5years  + salary, data=D1, na=na.gam.replace,
                control = gam.control(epsilon=1e-04, bf.epsilon = 1e-04, maxit=50,
                                      bf.maxit = 50), family="binomial")
```

Comment:

The above GAM models were defined as the sum of smooth covariate functions plus a standard parametric component of the linear predictors. Because the answer variable had binary outcomes, the family was set to binary. Also, the epsilon value is used to judge the conversion of the GLM IRLS loop, with a maximum of 50 IRLS iterations. The best model was then chosen using the BIC criterion. Furthermore, the AIC was used to pick the significant variables in the model using stepwise selection.

```
summary(gam.fit1)
```

```
##
## Call: gam(formula = left ~ s(satisfaction_level) + s(last_evaluation) +
##     number_project + lo(satisfaction_level, last_evaluation) +
##     average_montly_hours + s(time_spend_company) + Work_accident +
##     promotion_last_5years + salary, family = "binomial", data = D1,
##     na.action = na.gam.replace, control = gam.control(epsilon = 1e-04,
##         bf.epsilon = 1e-04, maxit = 50, bf.maxit = 50))
## Deviance Residuals:
##       Min        1Q    Median        3Q       Max
## -2.652941 -0.406716 -0.173566 -0.009963  3.622023
##
## (Dispersion Parameter for binomial family taken to be 1)
##
##     Null Deviance: 10918.56 on 9998 degrees of freedom
## Residual Deviance: 4687.692 on 9973.802 degrees of freedom
## AIC: 4738.087
##
## Number of Local Scoring Iterations: 1
##
## Anova for Parametric Effects
##                           Df Sum Sq Mean Sq  F value    Pr(>F)
## s(satisfaction_level)    1.0  437.0  436.97 504.3120 < 2.2e-16 ***
## s(last_evaluation)       1.0   54.3   54.25  62.6136 2.782e-15 ***
## number_project           1.0    4.4    4.42   5.1012  0.023931 *
## average_montly_hours     1.0   84.8   84.80  97.8728 < 2.2e-16 ***
## s(time_spend_company)    1.0  247.3  247.25 285.3598 < 2.2e-16 ***
## Work_accident            1.0  123.9  123.87 142.9612 < 2.2e-16 ***
## promotion_last_5years    1.0    7.6    7.59   8.7624  0.003082 **
## salary                   2.0  114.2   57.09  65.8890 < 2.2e-16 ***
## Residuals             9973.8 8642.0    0.87
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                                           Npar Df Npar Chisq    P(Chi)
## (Intercept)
## s(satisfaction_level)                         3.0     650.33 < 2.2e-16 ***
## s(last_evaluation)                            3.0     312.58 < 2.2e-16 ***
## number_project
## lo(satisfaction_level, last_evaluation)     6.2    1194.16 < 2.2e-16 ***
## average_montly_hours
## s(time_spend_company)                         3.0     289.31 < 2.2e-16 ***
## Work_accident
## promotion_last_5years
## salary
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(gam.fit2)
```

```
##
## Call: gam(formula = left ~ lo(satisfaction_level) + s(last_evaluation) +
##     lo(number_project) + lo(average_montly_hours) + time_spend_company +
##     Work_accident + promotion_last_5years + salary, family = "binomial",
##     data = D1, na.action = na.gam.replace, control = gam.control(epsilon = 1e-04,
##         bf.epsilon = 1e-04, maxit = 50, bf.maxit = 50))
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.62125 -0.36882 -0.15941 -0.03214  3.50807
##
## (Dispersion Parameter for binomial family taken to be 1)
##
##     Null Deviance: 10918.56 on 9998 degrees of freedom
## Residual Deviance: 4687.902 on 9977.343 degrees of freedom
## AIC: 4731.217
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                                Df Sum Sq Mean Sq F value    Pr(>F)
## lo(satisfaction_level)        1.0   31.1  31.124  31.178 2.415e-08 ***
## s(last_evaluation)            1.0  147.8 147.800 148.059 < 2.2e-16 ***
## lo(number_project)            1.0  101.2 101.204 101.382 < 2.2e-16 ***
## lo(average_montly_hours)      1.0  118.6 118.577 118.785 < 2.2e-16 ***
## time_spend_company          1.0  130.2 130.190 130.418 < 2.2e-16 ***
## Work_accident               1.0  134.7 134.707 134.943 < 2.2e-16 ***
## promotion_last_5years       1.0   13.9  13.936  13.961 0.0001877 ***
## salary                      2.0  137.4  68.698  68.818 < 2.2e-16 ***
## Residuals                9977.3 9959.9   0.998
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                          Npar Df Npar Chisq    P(Chi)
## (Intercept)
## lo(satisfaction_level)       2.3     469.56 < 2.2e-16 ***
## s(last_evaluation)           3.0     384.06 < 2.2e-16 ***
## lo(number_project)           4.0     830.85 < 2.2e-16 ***
## lo(average_montly_hours)     2.3     333.27 < 2.2e-16 ***
## time_spend_company
## Work_accident
## promotion_last_5years
## salary
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Comment:

Taking into account the parametric effects, all variables were statistically significant at 0.001 significance level. For the non-parameteric effects, the variables *time_spend_company, Work_accident, promotion_last_5years* and *salary* were statistically insignificant at any of the lest level whilst the remaining variables were all statistically significant at 0.001 alpha level.

6.2 Selecting the best model using the BIC criterion

```r
b.df <- BIC(gam.fit1, gam.fit2)

b.df <- knitr::kable(b.df, booktabs = T, format = "markdown", digits = 2)
kableExtra::kable_styling(b.df, bootstrap_options = "striped", full_width = F)
```

	df	BIC
gam.fit1	10	4810.19
gam.fit2	10	4803.32

Comment:

Based on the BIC values computed above, **gam.fit2** model comparatively can be considered as the best since it has the smaller BIC value.

6.2.1 Stepwise selection using the best model from previous results

```
fit.step <- step.Gam(gam.fit2, scope=list("satisfaction_level"=~1 + satisfaction_level+
                    lo(satisfaction_level),
"last_evaluation"=~1 + last_evaluation + lo(last_evaluation, 3) + s(last_evaluation, 2),
"number_project"=~1 + number_project + s(number_project, 2) + s(number_project, 4),
    "average_monthly_hours"=~1 + average_montly_hours + s(average_montly_hours, 3) +
         s(average_montly_hours, 6),
           "time_spend_company"=~1 + time_spend_company + lo(time_spend_company, 3) +
            s(time_spend_company, 2),
           "Work_accident"=~1+Work_accident,
           "promotion_last_5years"=~1 + promotion_last_5years,
           # "sales"=~1 + sales,
           "salary"=~1 + salary),
           scale=2, steps=1000, parallel=T, direction="both")
```

```
## Start:  left ~ lo(satisfaction_level) + s(last_evaluation) + lo(number_project) +      lo(average_
```

```
summary(fit.step)
```

```
##
## Call: gam(formula = left ~ lo(satisfaction_level) + s(last_evaluation) +
##     lo(number_project) + lo(average_montly_hours) + time_spend_company +
##     Work_accident + promotion_last_5years + salary, family = "binomial",
##     data = D1, na.action = na.gam.replace, control = gam.control(epsilon = 1e-04,
##         bf.epsilon = 1e-04, maxit = 50, bf.maxit = 50))
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.62125 -0.36882 -0.15941 -0.03214  3.50807
##
## (Dispersion Parameter for binomial family taken to be 1)
##
##     Null Deviance: 10918.56 on 9998 degrees of freedom
## Residual Deviance: 4687.902 on 9977.343 degrees of freedom
## AIC: 4731.217
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                                Df Sum Sq Mean Sq F value    Pr(>F)
## lo(satisfaction_level)        1.0   31.1  31.124  31.178 2.415e-08 ***
## s(last_evaluation)            1.0  147.8 147.800 148.059 < 2.2e-16 ***
## lo(number_project)            1.0  101.2 101.204 101.382 < 2.2e-16 ***
## lo(average_montly_hours)      1.0  118.6 118.577 118.785 < 2.2e-16 ***
## time_spend_company            1.0  130.2 130.190 130.418 < 2.2e-16 ***
## Work_accident                 1.0  134.7 134.707 134.943 < 2.2e-16 ***
## promotion_last_5years         1.0   13.9  13.936  13.961 0.0001877 ***
## salary                        2.0  137.4  68.698  68.818 < 2.2e-16 ***
## Residuals                  9977.3 9959.9   0.998
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Anova for Nonparametric Effects
##                           Npar Df Npar Chisq    P(Chi)
## (Intercept)
## lo(satisfaction_level)        2.3     469.56 < 2.2e-16 ***
## s(last_evaluation)            3.0     384.06 < 2.2e-16 ***
## lo(number_project)            4.0     830.85 < 2.2e-16 ***
## lo(average_montly_hours)      2.3     333.27 < 2.2e-16 ***
## time_spend_company
## Work_accident
## promotion_last_5years
## salary
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Comment:

For the parametric effects, all variables were statistically significant at 0.001 significance level. For the non-parametric effects, the variables *time_spend_company, Work_accident, promotion_last_5years* and *salary* were statistically insignificant at any of the lest level whilst the remaining variables were all statistically significant at 0.001 alpha level.

```
anova(fit.step, gam.fit2)
```

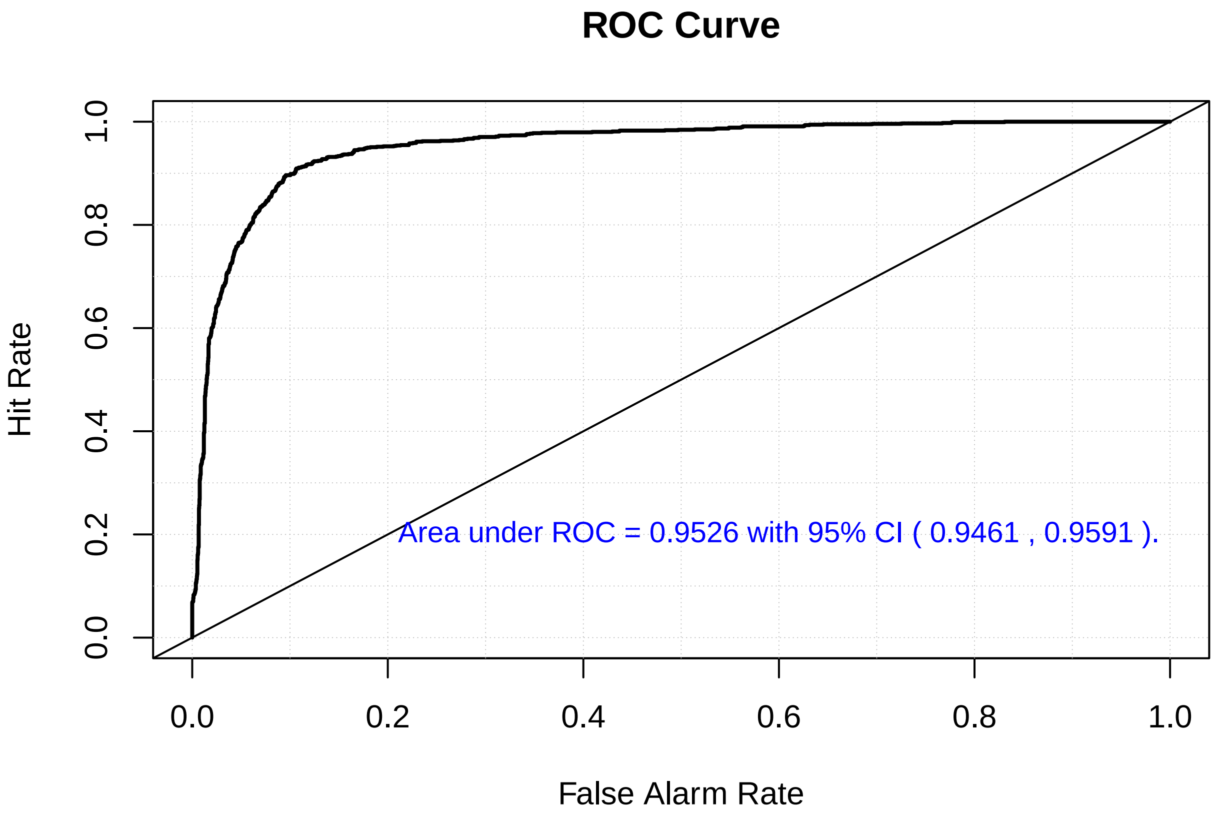
```
## Analysis of Deviance Table
##
## Model 1: left ~ lo(satisfaction_level) + s(last_evaluation) + lo(number_project) +
##     lo(average_montly_hours) + time_spend_company + Work_accident +
##     promotion_last_5years + salary
## Model 2: left ~ lo(satisfaction_level) + s(last_evaluation) + lo(number_project) +
##     lo(average_montly_hours) + time_spend_company + Work_accident +
##     promotion_last_5years + salary
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1    9977.3     4687.9
## 2    9977.3     4687.9  0        0
```

6.3 ROC and AUC for GAM

```
yhat <- gam.fit2$fitted.values
gam.pred <- predict(gam.fit2, newdata=D2, type="response", se.fit=FALSE)
library(cvAUC)
gam.AUC <- ci.cvAUC(predictions = gam.pred, labels=yobs,
                    folds=1:length(gam.pred), confidence=0.95);
gam.auc.ci <- round(gam.AUC$ci, digits=4)

library(verification)
mod.gam <- verify(obs=yobs, pred=gam.pred)
```

```
## If baseline is not included, baseline values  will be calculated from the  sample obs.
```

```
#par(mfrow=c(1,1), mar=rep(4, 4))
roc.plot(mod.gam, plot.thres = NULL)
text(x=0.6, y=0.2, paste("Area under ROC =", round(gam.AUC$cvAUC, digits=4),
                  "with 95% CI (", gam.auc.ci[1], ",", gam.auc.ci[2], ").",
                         sep=" "), col="blue", cex=0.9)
```



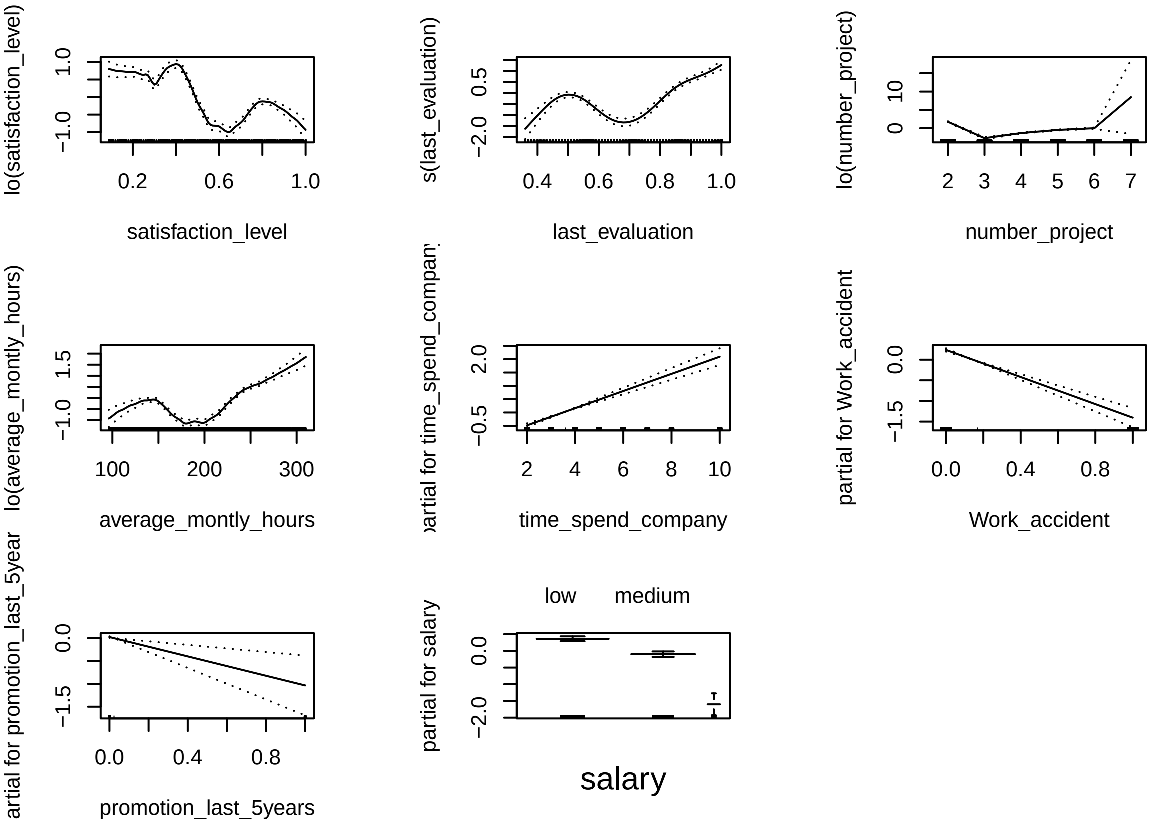
```
auc.gam<-round(gam.AUC$cvAUC, digits=4)
```

Comment:

The AUC from the **GAM** model is 0.95 which is high and it tells us that the model is a good fit and has good prediction accuracy.From the output, it can be concluded from the confidence interval that the model shows good discrimination.

Plotting the (nonlinear) functional forms for continuous predictors.

```
par(mfrow=c(3,3), mar = rep(4,4))
plot(gam.fit2, se =TRUE)
```



Comment:

In the backfitting algorithm, each smoothing parameter was calculated adaptively. Because smoothing splines are utilized in this scenario, the tuning parameter is automatically optimized using minimal GCV. Stepwise selection with BIC was also used to choose variables. The considerable nonlinearity displayed in the plots above demonstrates that the logistic regression model is inadequate.

7 QUESTION 7 - Multivariate Adaptive Regression Splines (MARS)

```
# install.packages("earth")
library(earth)    # for MARS
library(pdp)      # for partial dependence plots
library(vip)
library(caret)# for variable importance plots
```

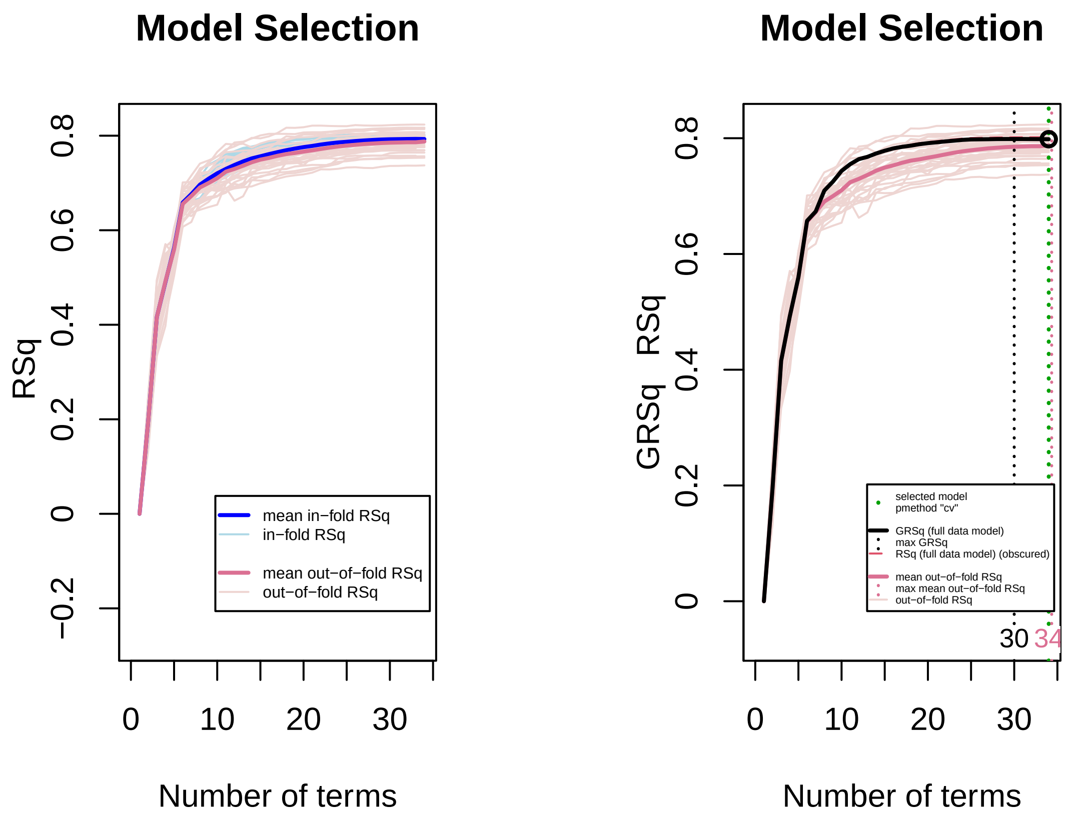
7.1 Fitting MARS model

```
fit.mars <- earth(left ~ .,  data = D1, degree=3, ncross=3,
    glm=list(family=binomial(link = "logit")),
    pmethod="cv", nfold=10) # tuning parameter degree = 3

summary(fit.mars)
```

```
## Call: earth(formula=left~., data=D1, pmethod="cv",
##             glm=list(family=binomial(link="logit")), degree=3, nfold=10,
##             ncross=3)
##
## GLM coefficients
##                                                                                            left
## (Intercept)                                                                             -4.3889
## h(0.15-satisfaction_level)                                                             152.8452
## h(satisfaction_level-0.15)                                                               4.3650
## h(3-number_project)                                                                     10.2973
## h(number_project-3)                                                                      0.8549
## h(satisfaction_level-0.15) * h(last_evaluation-0.99)                                   232.4778
## h(satisfaction_level-0.15) * h(0.99-last_evaluation)                                     2.0527
## h(0.23-satisfaction_level) * h(number_project-3)                                        -3.3099
## h(satisfaction_level-0.23) * h(number_project-3)                                        -0.6362
## h(0.38-satisfaction_level) * h(3-number_project)                                       -34.1717
## h(satisfaction_level-0.38) * h(3-number_project)                                       -21.4896
## h(satisfaction_level-0.15) * h(average_montly_hours-286)                                 5.1564
## h(satisfaction_level-0.15) * h(286-average_montly_hours)                                -0.0425
## h(satisfaction_level-0.15) * h(time_spend_company-5)                                   -11.8019
## h(satisfaction_level-0.15) * h(5-time_spend_company)                                    -2.2165
## h(satisfaction_level-0.15) * h(time_spend_company-4)                                     8.4583
## h(0.46-last_evaluation) * h(3-number_project)                                         -135.4747
## h(last_evaluation-0.46) * h(3-number_project)                                          -16.8712
## h(3-number_project) * h(average_montly_hours-131)                                       -0.0223
## h(3-number_project) * h(131-average_montly_hours)                                       -0.3664
## h(3-number_project) * h(time_spend_company-3)                                           -1.2582
## h(3-number_project) * h(3-time_spend_company)                                           -3.5028
## h(number_project-3) * h(time_spend_company-5)                                           -0.5235
## h(number_project-3) * h(5-time_spend_company)                                            0.0546
## h(satisfaction_level-0.38) * h(last_evaluation-0.45) * h(3-number_project)              51.8794
## h(satisfaction_level-0.38) * h(0.45-last_evaluation) * h(3-number_project)          -11529.3660
## h(satisfaction_level-0.15) * h(0.99-last_evaluation) * h(time_spend_company-5)          20.0467
## h(satisfaction_level-0.15) * h(0.99-last_evaluation) * h(5-time_spend_company)          -1.4815
## h(satisfaction_level-0.15) * h(0.99-last_evaluation) * h(time_spend_company-4)         -18.7074
## h(satisfaction_level-0.23) * h(number_project-3) * h(time_spend_company-5)            1.7461
## h(satisfaction_level-0.23) * h(number_project-3) * h(5-time_spend_company)           -0.0998
## h(satisfaction_level-0.23) * h(number_project-3) * h(time_spend_company-4)           -0.2850
## h(satisfaction_level-0.15) * h(average_montly_hours-283) * h(5-time_spend_company)   -0.3454
## h(satisfaction_level-0.15) * h(283-average_montly_hours) * h(5-time_spend_company)    0.0174
##
## GLM (family binomial, link logit):
##  nulldev   df       dev   df   devratio     AIC iters converged
##  10918.6 9998   2316.59 9965      0.788    2385    20         1
##
## Earth selected 34 of 34 terms, and 5 of 18 predictors (pmethod="cv")
## Termination condition: Reached nk 37
## Importance: satisfaction_level, number_project, time_spend_company, ...
## Number of terms at each degree of interaction: 1 4 19 10
## Earth GRSq 0.7983041  RSq 0.8016191  mean.oof.RSq 0.7877309 (sd 0.0207)
##
## pmethod="backward" would have selected:
##     30 terms 5 preds,  GRSq 0.7986453  RSq 0.801555  mean.oof.RSq 0.7852753
```

Model selection

```
# MODEL SELECTION
par(mfrow=c(1, 2), mar=rep(4,4))
q1 <- plot(fit.mars, which = 1, col.mean.infold.rsq="blue",
           col.infold.rsq="lightblue",col.grsq=0, col.rsq=0,
           col.vline=0, col.oof.vline=0)
plotres(fit.mars, which=1, info = TRUE)
```



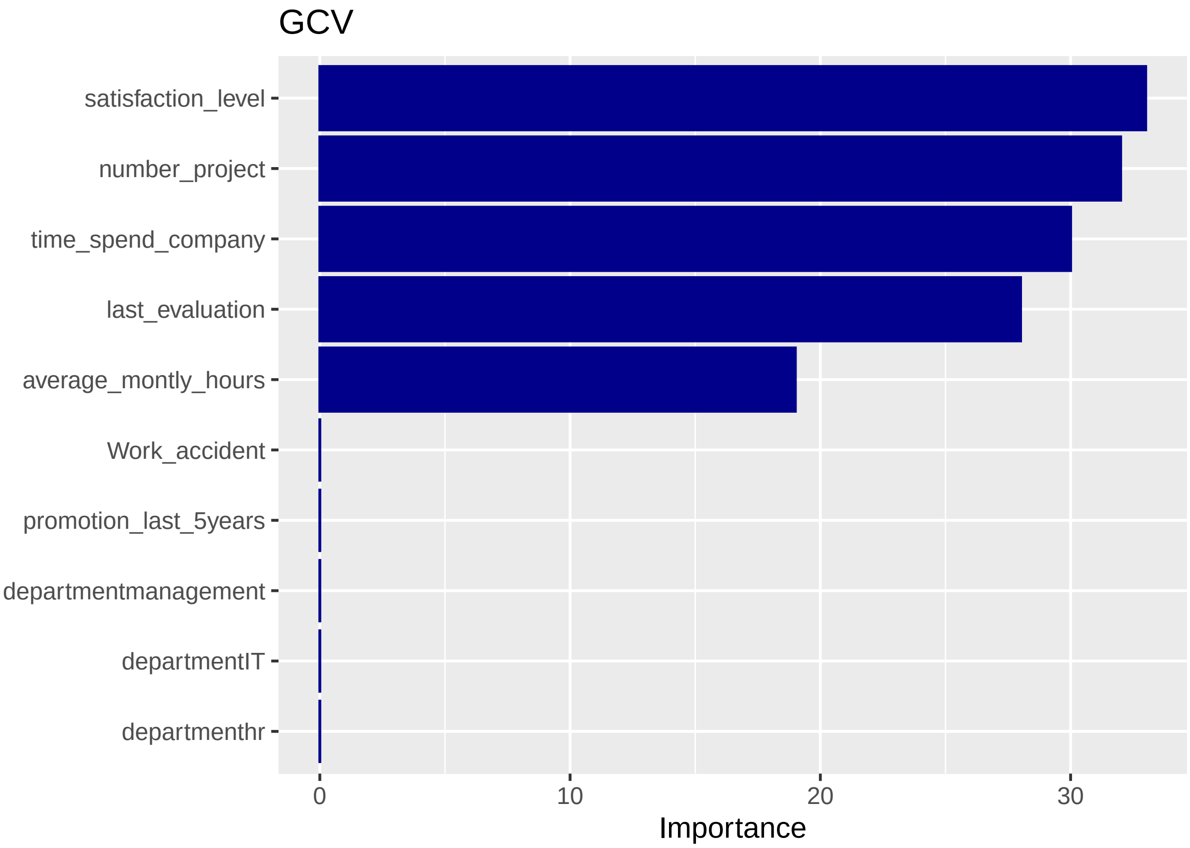
Comment:

The plot on the left indicates training and testing performance (Rsq on Y axis) obtained from the 10 fold cross validation performed thrice. The performance on training data (blue curve) increases as we increase model complexity; on independent data the performance (pink curve) increases as well.

The plot on the right shows the best model selection (33 of 34 terms, 5 of 18 predictors using **pmethod="cv"**) using green line, indicating optimal terms as 33.

Variable importance plot

```
vip(fit.mars, num_features = 10, aesthetics = list(color = "darkblue", fill = "darkblue")) + ggtitle(
```



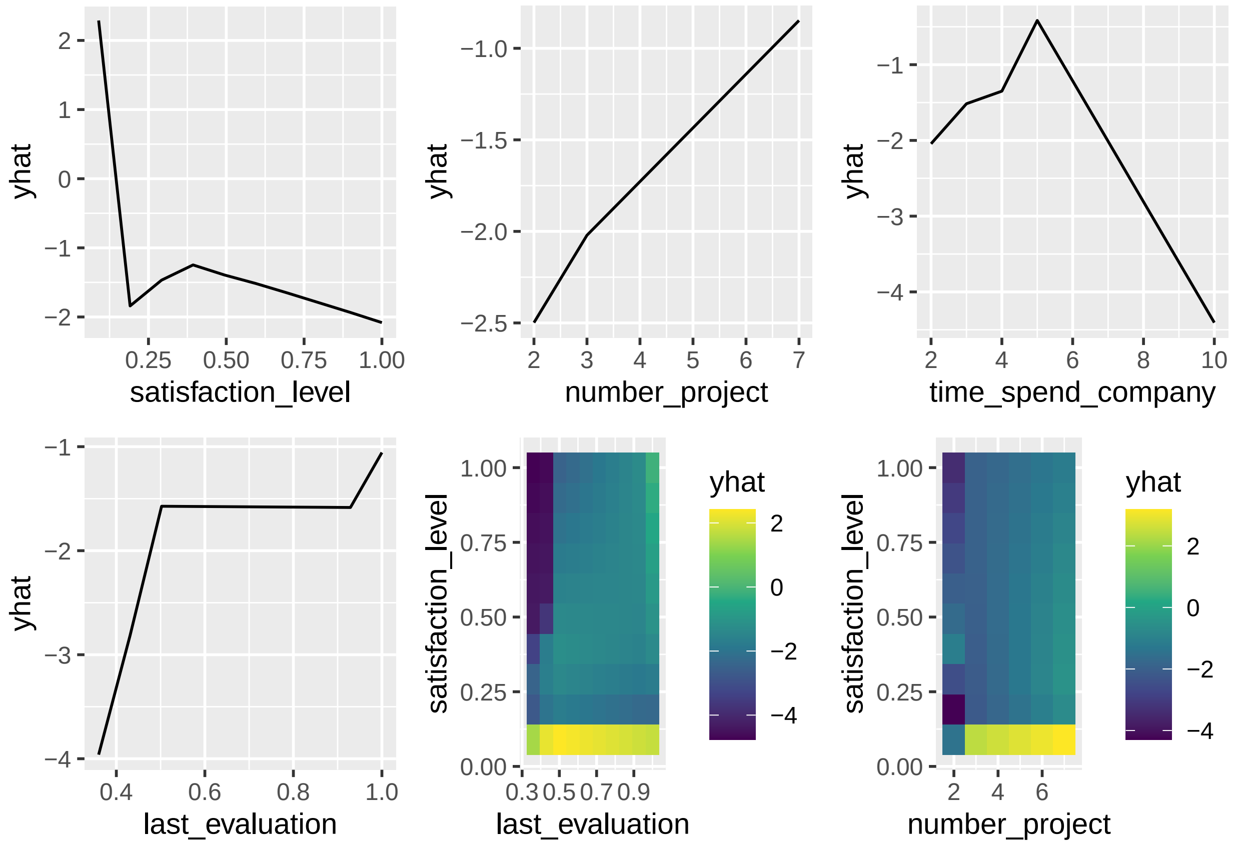
Comment:

Fromt the **vip**, we see that **satisfaction_level, number_project, time_spend_company** and **last_evaluation** are the four most influential variables; however, variable importance does not tell us how our model is treating the non-linear patterns for each feature. Also, if we look at the interaction terms our model retained, we see interactions between different hinge functions.

Partial dependence plot

```
#par(mfrow=c(2,2), mar = rep(4,4))
q1 <- partial(fit.mars, pred.var = "satisfaction_level", grid.resolution = 10)%>%autoplot()
q2 <- partial(fit.mars, pred.var = "number_project", grid.resolution = 10)%>%autoplot()
q3 <- partial(fit.mars, pred.var = "time_spend_company", grid.resolution = 10)%>%autoplot()
q4 <- partial(fit.mars, pred.var = "last_evaluation", grid.resolution = 10)%>%autoplot()
q5 <- partial(fit.mars, pred.var = c("last_evaluation","satisfaction_level"),
              grid.resolution = 10)%>%autoplot()
```

```
q6 <- partial(fit.mars, pred.var = c("number_project","satisfaction_level"),
              grid.resolution = 10)%>%autoplot()

grid.arrange(q1,q2,q3,q4,q5,q6, ncol = 3)
```



Comment:

From the first plot on the upper panel, we see that the is a sharp increase in odds of an employee quit until a certain threshold of satisfaction level, the odds start decreasing as satisfaction level increases. Also, the variables **number_project and log(odds(left))** seems to be a directly proportional.

This 2-D plot gives us an idea about how the association of two variables at a time on **log(odds(left))**. On the X-Axis we have **satisfaction_level** and on the Y-Axis we have **last_evaluation**. The variation is **log(odds(left))** is indicated with the help of color scale (yellow indicating high odds and dark violet indicating low odds).

```
yhat.mars <- predict(fit.mars, newdata=D2, type="response")
AUC.MARS <- ci.cvAUC(predictions=as.vector(yhat.mars), labels=yobs, folds=1:length(yhat.mars), confid
```

```
## $cvAUC
## [1] 0.984323
##
## $se
## [1] 0.002308812
##
## $ci
## [1] 0.9797978 0.9888482
##
## $confidence
## [1] 0.95
```

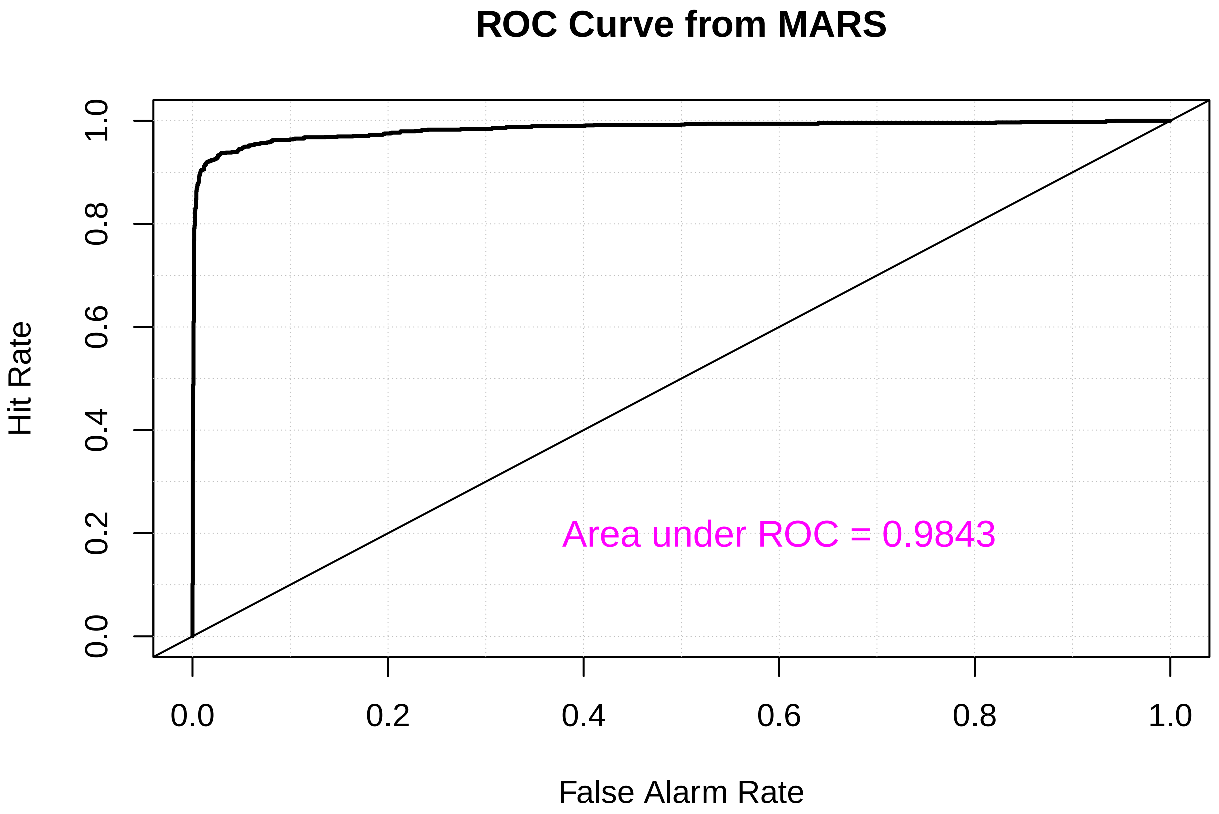
```r
auc.ci <- round(AUC.MARS$ci, digits=4)

library(verification)
mod.mars <- verify(obs=yobs, pred=yhat.mars)
```

```
## If baseline is not included, baseline values  will be calculated from the  sample obs.
```

```r
roc.plot(mod.mars, plot.thres = NULL, main="ROC Curve from MARS")
```

```
## Warning in roc.plot.default(c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, : Large
## amount of unique predictions used as thresholds. Consider specifying
## thresholds.
```

```r
text(x=0.6, y=0.2, paste("Area under ROC =", round(AUC.MARS$cvAUC, digits=4),
    sep=" "), col="magenta", cex=1.2)
```



Comment:

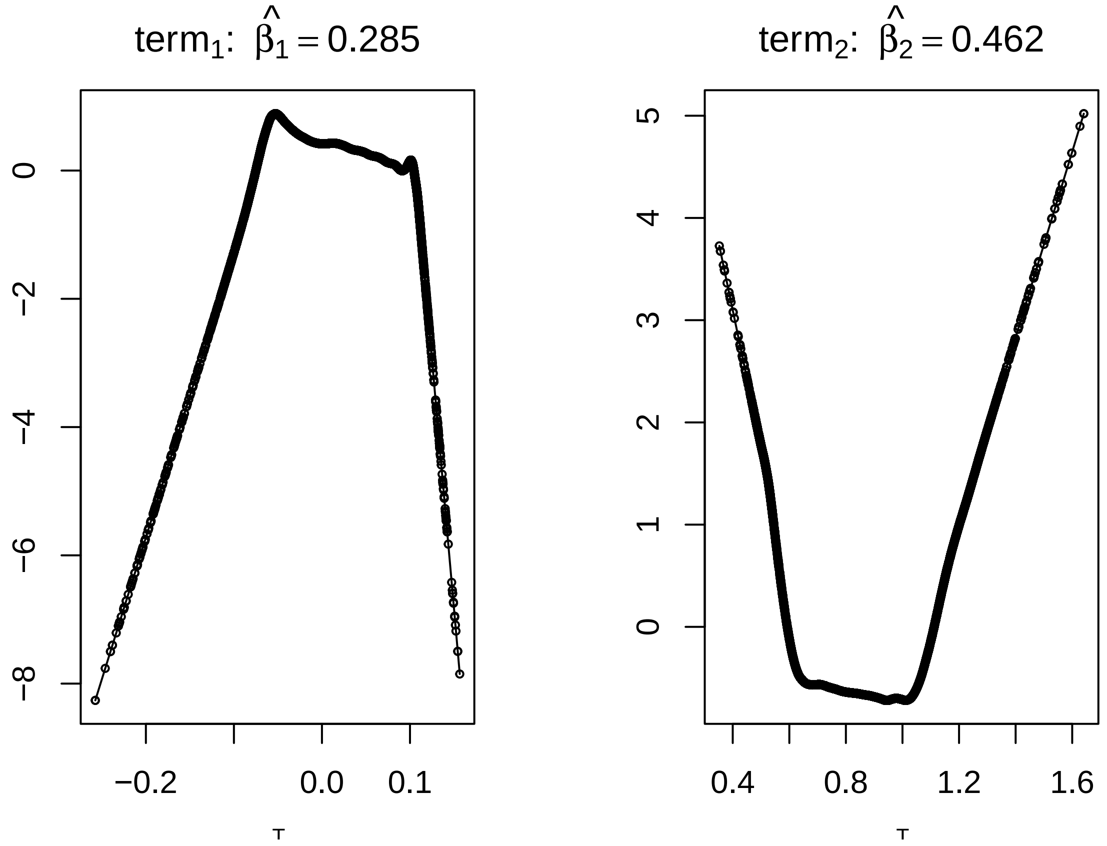
The AUC from the MARS is 0.98 which is high and it tells us that the model is a good fit and has good prediction accuracy.

8 Question 8 - Project Pursuit Regression

```
# FIT PPR MODELS
# nterms = number of terms to include in the final model.
# max.terms = maximum number of terms to choose from when building the model.

D1$left <- ifelse(D1$left=="stayed", 0, 1)
fit0.ppr <- ppr(left ~ ., data = D1,
    nterms = 2, max.terms = 10,
    sm.method = "supsmu", bass=3, spen=0)
summary(fit0.ppr)
```

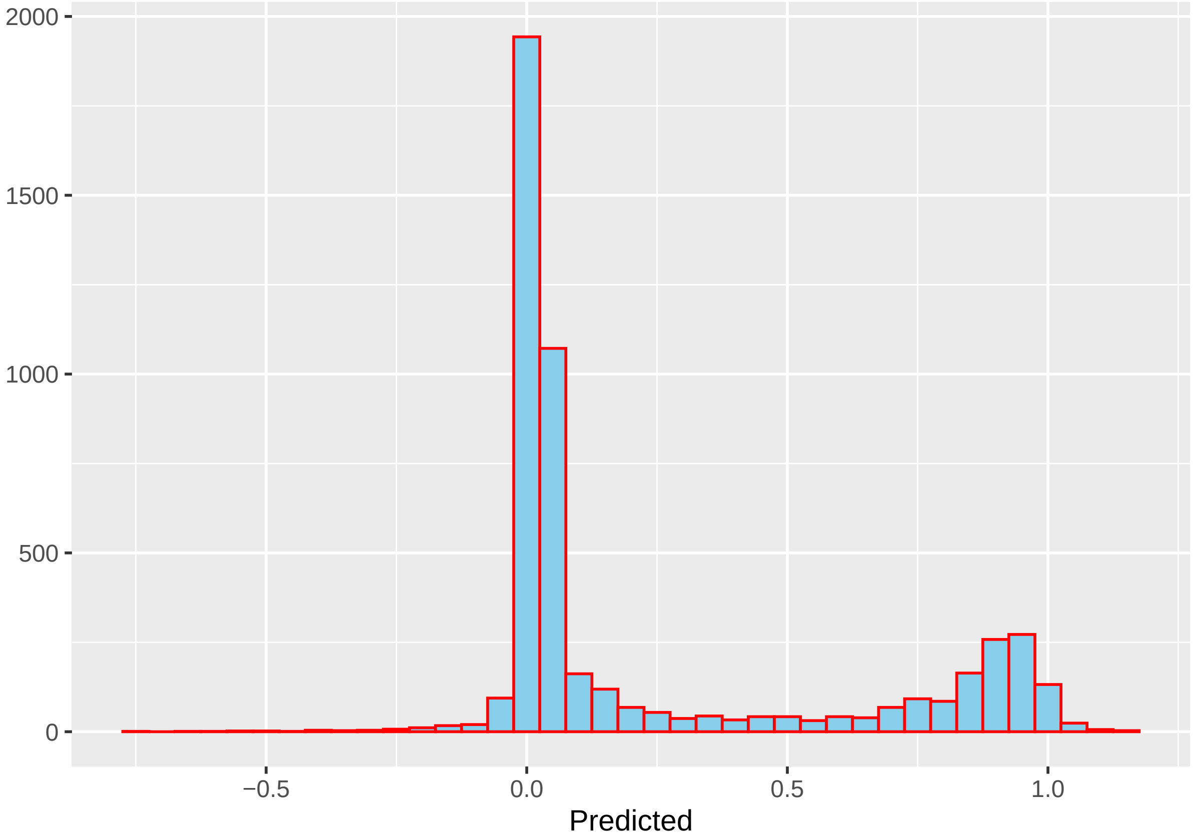
```
## Call:
## ppr(formula = left ~ ., data = D1, nterms = 2, max.terms = 10,
##     sm.method = "supsmu", bass = 3, spen = 0)
##
## Goodness of fit:
##  2 terms  3 terms  4 terms  5 terms  6 terms  7 terms  8 terms  9 terms
## 451.1104 450.2731 423.4635 467.2904 436.1631 438.1591   0.0000   0.0000
## 10 terms
##   0.0000
##
## Projection direction vectors ('alpha'):
##                        term 1        term 2
## satisfaction_level      0.0012110979  0.0629132411
## last_evaluation        -0.0672644688  0.3751051150
## number_project         -0.0186586682  0.0821277029
## average_montly_hours   -0.0002477212  0.0012572350
## time_spend_company     -0.0343755565  0.0801642535
## Work_accident           0.0023366510 -0.0126586556
## promotion_last_5years   0.0070076745 -0.0366993631
## departmentaccounting    0.3144994785 -0.2894724412
## departmenthr            0.3148302260 -0.2834694747
## departmentIT            0.3149965761 -0.2913515756
## departmentmanagement    0.3135734074 -0.2862039390
## departmentmarketing     0.3145167534 -0.2897002318
## departmentproduct_mng   0.3146254501 -0.2886926810
## departmentRandD         0.3185114448 -0.2992944758
## departmentsales         0.3151921676 -0.2926202789
## departmentsupport       0.3164398008 -0.2900974043
## departmenttechnical     0.3153881345 -0.2882338908
## salary.L               -0.0009592769  0.0003331341
## salary.Q               -0.0003236076  0.0012576295
##
## Coefficients of ridge terms ('beta'):
##    term 1    term 2
## 0.2848580 0.4617535
```

```
par(mfrow=c(1,2), mar=rep(3,4))
plot(fit0.ppr)
```



```r
yhat1.pp <- predict(fit0.ppr, newdata = D2)

qplot(yhat1.pp, geom="histogram", xlab="Predicted",
      fill=I("sky blue"), col=I("red"), binwidth=0.05)
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



```r
# SCALE PREDICTED Y INTO [0,1]
phat1.pp <- as.vector(scale(yhat1.pp, center=min(yhat1.pp),
                   scale = diff(range(yhat1.pp))))
```

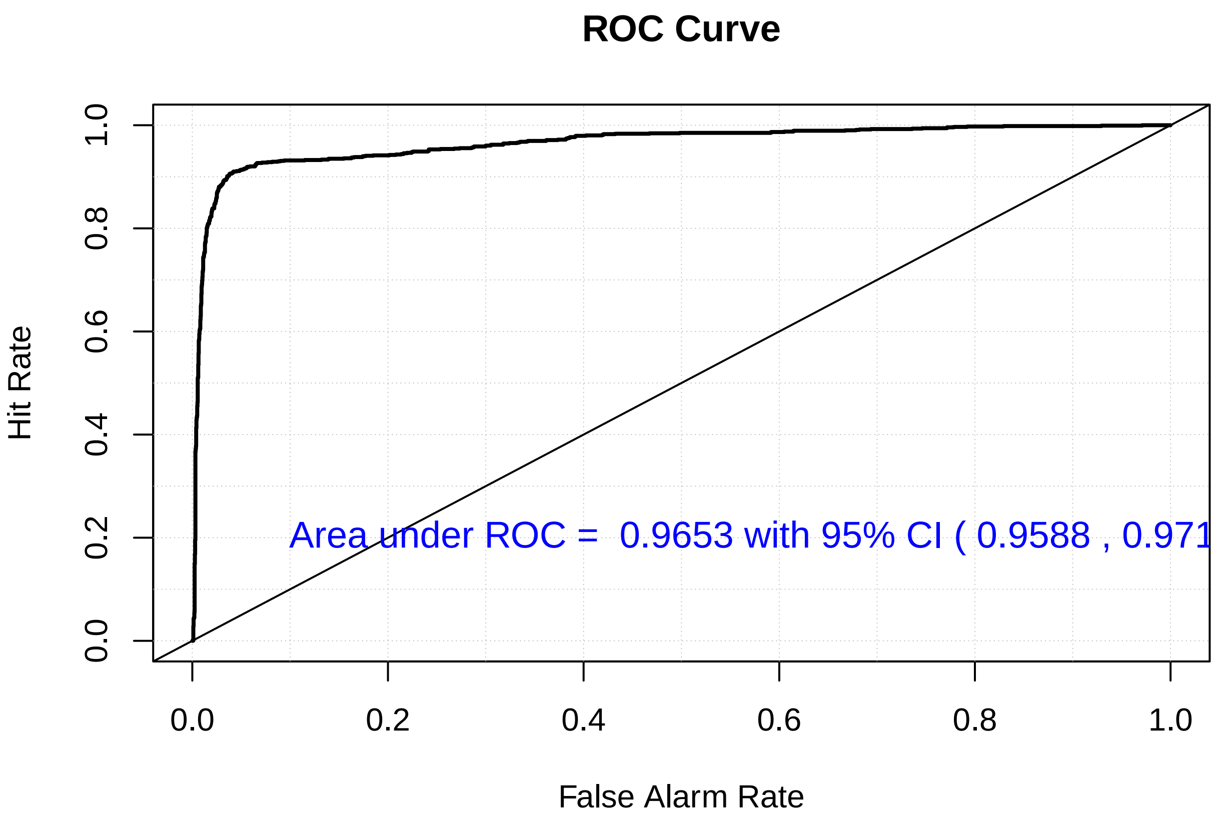
```r
ppr.AUC <- ci.cvAUC(predictions=phat1.pp, labels=yobs,
                    folds=1:length(phat1.pp), confidence=0.95)
ppr.auc.ci <- round(ppr.AUC$ci, digits = 4)

library(verification)
mod.ppr <- verify(obs = yobs, pred = phat1.pp)
```

```
## If baseline is not included, baseline values  will be calculated from the  sample obs.
```

```r
roc.plot(mod.ppr, plot.thres=NULL)
```

```
## Warning in roc.plot.default(c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, : Large
## amount of unique predictions used as thresholds. Consider specifying
## thresholds.
```

```r
text(x=0.6, y=0.2, paste("Area under ROC = ",
                         round(ppr.AUC$cvAUC, digits = 4),
                         "with 95% CI (",
                         ppr.auc.ci[1], ",", ppr.auc.ci[2],
                         ").", sep = " "), col="blue", cex =1.2)
```



The AUC from the PPR is 0.97 which is high and it tells us that the model is a good fit and has good prediction accuracy.

9 Model Comparison

```r
library(kableExtra)
Measure <- c(AUC.LASSO$cvAUC, AUC.RF$cvAUC, gam.AUC$cvAUC, AUC.MARS$cvAUC,
             ppr.AUC$cvAUC)
mod <- data.frame("Method"= c("LASSO","Random Forest","GAM","MARS","PPR"),
                       "AUC"= Measure)

knitr::kable(mod, booktabs = T, format = "markdown", digits = 3) %>%
  kable_paper("hover", full_width = F)%>%
  kable_styling(font_size = 12,bootstrap_options = "striped",
                   full_width = F, latex_options = c("HOLD_position"))
```

Method	AUC
LASSO	0.811
Random Forest	0.992
GAM	0.953
MARS	0.984
PPR	0.965

Comment:

From the output above, we can conclude that among all the five supervised learning approaches used, the **Random Forest** gives favorable result since its model produced relatively high area under the receiver operating characteristic curve, that is $AUC = 0.99$.