Project VI

Quaye E. George

Due: 11/16/2020

Contents

1 Data Preparation

Bring in the data D and name it as, say, hr. Change the categorical variable salary in the data set to ordinal:

```
# Bring in the Data
hr.Data <- read.table(file="HR_comma_sep.csv",sep=",", header = TRUE)
colnames(hr.Data)[9]<-"department"
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
```

```
hr.Data<-hr.Data%>%
  select(-left,left)
head(hr.Data)
```

```
##   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 promotion_last_5years department salary left
## 1                  3             0                     0      sales    low    1
## 2                  6             0                     0      sales medium    1
## 3                  4             0                     0      sales medium    1
## 4                  5             0                     0      sales    low    1
## 5                  3             0                     0      sales    low    1
## 6                  3             0                     0      sales    low    1
```

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

Inspect if there is any missing values and, if so, handle them with imputation.

```
# INSPECT THE DISTINCT VALUES OF EACH X
for (j in 1:NCOL(hr.Data)){
  x <- hr.Data[,j]
  print(table(x, useNA="ifany"))
}
```

```
# 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)
}
miss.info(hr.Data)
```

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

Given the output, there are no missing values from the dataset (hr.Data)

2 Exploratory Data Analysis

2.1 Preliminary Statistical Analysis

```
# Checking the dimension of data
dim(hr.Data)
```

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

The dataset (hr.Data) contains 10 columns and 14999 observations.

```
# Check the type of our features.
str(hr.Data)
```

```
## 'data.frame':    14999 obs. of  10 variables:
##  $ satisfaction_level   : num  0.38 0.8 0.11 0.72 0.37 0.41 0.1 0.92 0.89 0.42 ...
##  $ last_evaluation      : num  0.53 0.86 0.88 0.87 0.52 0.5 0.77 0.85 1 0.53 ...
##  $ number_project       : int  2 5 7 5 2 2 6 5 5 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 ...
##  $ 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 ...
##  $ department           : chr  "sales" "sales" "sales" "sales" ...
##  $ salary               : Ord.factor w/ 3 levels "low"<"medium"<..: 1 2 2 1 1 1 1 1
##  $ left                 : int  1 1 1 1 1 1 1 1 1 1 ...
```

From the output above, it is indicated that among all the predictors, 2 are continuous; 5 are categorical and the remaining other 3 variables are integer counts.

```
#Percentage of employees who stayed and those who left
prop.table(table(hr.Data$left))*100
```

```
##
##        0        1
## 76.19175 23.80825
```

Given the above output, it is observed that about 76% of employees stayed and 24% of employees left.

```
# Overview of summary (Turnover V.S. Non-turnover)

cor_vars<-hr.Data[,c("satisfaction_level","last_evaluation","number_project","average_mo

aggregate(cor_vars[,c("satisfaction_level","last_evaluation","number_project","average_m
```

```
##   Category satisfaction_level last_evaluation number_project
## 1        0          0.6668096       0.7154734       3.786664
## 2        1          0.4400980       0.7181126       3.855503
##   average_montly_hours time_spend_company Work_accident promotion_last_5years
## 1             199.0602           3.380032    0.17500875           0.026251313
## 2             207.4192           3.876505    0.04732568           0.005320638
```

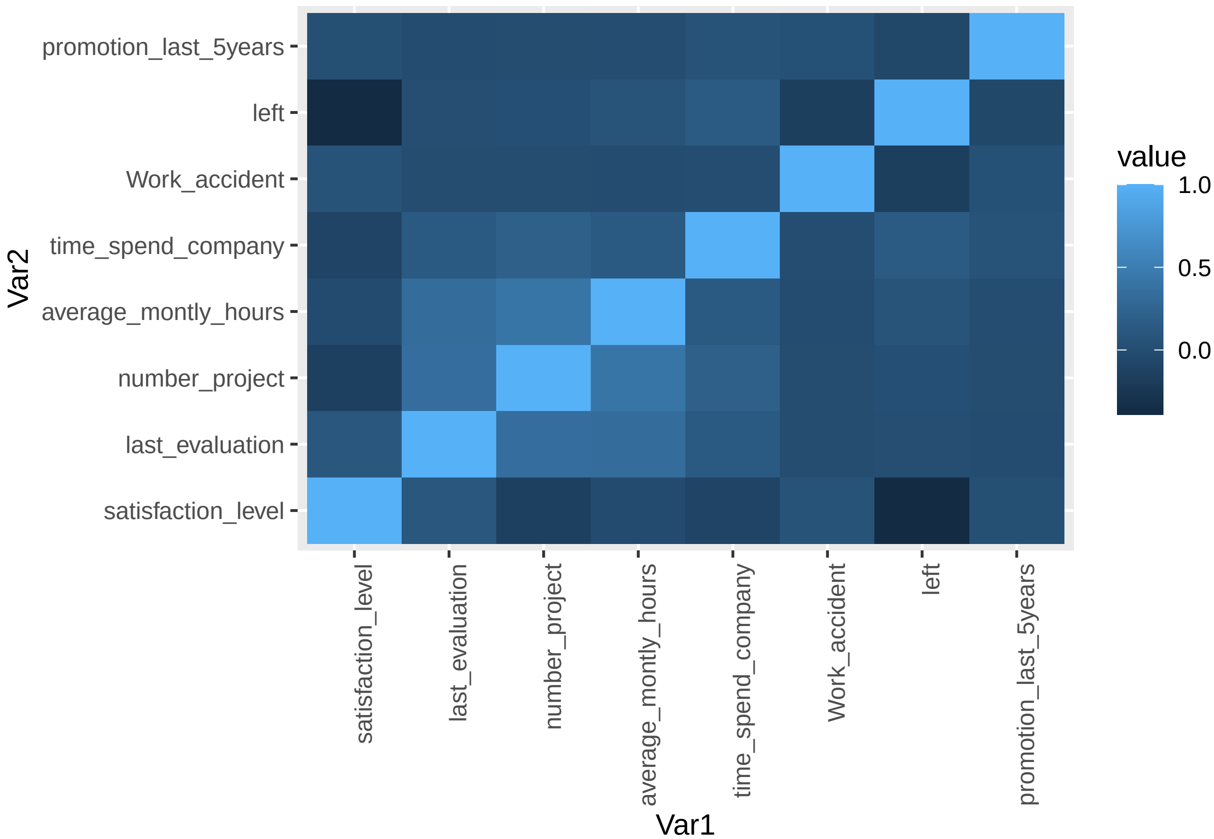
It is observed that the mean satisfaction of employees is 0.66 as against 0.44.

2.2 Correlation Matrix and Heat Map

```
#Correlation Matrix
library(reshape2)
library(ggplot2)
cor_vars<-hr.Data[,c("satisfaction_level","last_evaluation","number_project","average_mo
cor(cor_vars)
```

```
##                       satisfaction_level last_evaluation number_project
## satisfaction_level            1.00000000     0.105021214   -0.142969586
## last_evaluation               0.10502121     1.000000000    0.349332589
## number_project               -0.14296959     0.349332589    1.000000000
## average_montly_hours         -0.02004811     0.339741800    0.417210634
## time_spend_company           -0.10086607     0.131590722    0.196785891
## Work_accident                 0.05869724    -0.007104289   -0.004740548
## left                         -0.38837498     0.006567120    0.023787185
## promotion_last_5years         0.02560519    -0.008683768   -0.006063958
##                       average_montly_hours time_spend_company Work_accident
## satisfaction_level            -0.020048113       -0.100866073   0.058697241
## last_evaluation                0.339741800        0.131590722  -0.007104289
## number_project                 0.417210634        0.196785891  -0.004740548
## average_montly_hours           1.000000000        0.127754910  -0.010142888
## time_spend_company             0.127754910        1.000000000   0.002120418
## Work_accident                 -0.010142888        0.002120418   1.000000000
## left                           0.071287179        0.144822175  -0.154621634
## promotion_last_5years         -0.003544414        0.067432925   0.039245435
##                              left promotion_last_5years
## satisfaction_level    -0.38837498           0.025605186
## last_evaluation        0.00656712          -0.008683768
## number_project         0.02378719          -0.006063958
## average_montly_hours   0.07128718          -0.003544414
## time_spend_company     0.14482217           0.067432925
## Work_accident         -0.15462163           0.039245435
## left                   1.00000000          -0.061788107
## promotion_last_5years -0.06178811           1.000000000
```

```
trans<-cor(cor_vars)
melted_cormat <- melt(trans)

ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() +theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Given the correlation matrix output table, it is observed that;

i. Number of projects and average monthly hours are moderately positive correlated features (0.417210634).
ii. Turnover(left) and satisfaction level are moderately negative correlated features (-0.38837498).
iii. Last evaluation and number of project are moderately positive correlated features (0.349332589).
iv. Last evaluation and average monthly hours are moderately positive correlated features (0.339741800).

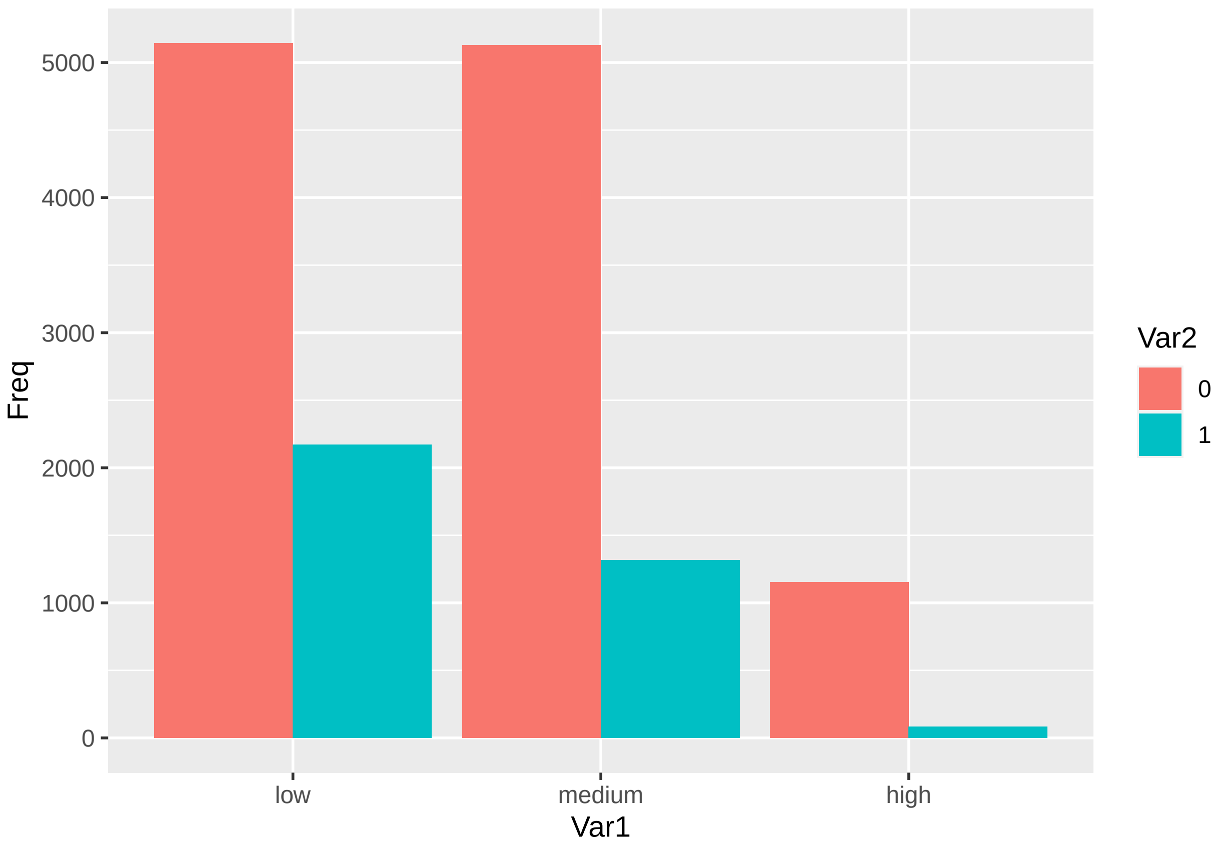
Also by the heat map, there is a positive correlation between number of project, average monthly hours, and evaluation. This may indicate that the employees who spent more hours and did more projects were evaluated on high. Again, For the negative relationships, turnover and satisfaction are highly correlated. This implies that people tend to leave the company more when they are less satisfied.

2.3 Salary V.S. Turnover

```
vis_1<-table(hr.Data$salary,hr.Data$left)
d_vis_1<-as.data.frame(vis_1)
print(d_vis_1)
```

```
##     Var1 Var2 Freq
## 1    low    0 5144
## 2 medium    0 5129
## 3   high    0 1155
## 4    low    1 2172
## 5 medium    1 1317
## 6   high    1   82
```

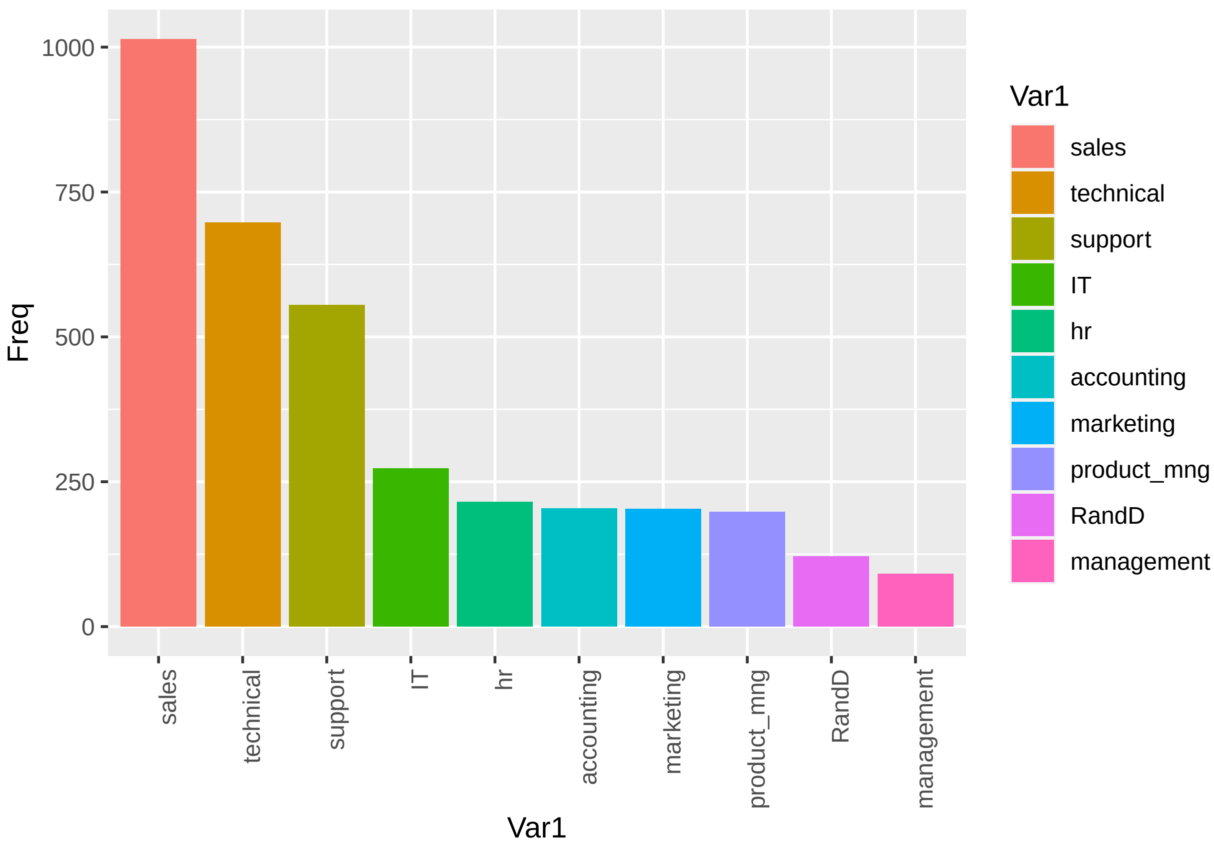
```
library(ggplot2)
p<-ggplot(d_vis_1, aes(x=Var1,y=Freq,fill=Var2)) +
 geom_bar(position="dodge",stat='identity')

print(p)
```



Given the plot above, it is observed that majority of employees who left either had low or medium salary. It is awkward for any employee with high salary to leave. Therefore employees with low to average salaries tend to leave the company mostly.

2.4 Department V.S. Turnover

```
vis_2<-table(hr.Data$department,hr.Data$left)
d_vis_2<-as.data.frame(vis_2)
d_vis_2<-subset(d_vis_2,Var2==1)
library(ggplot2)
d_vis_2$Var1 <- factor(d_vis_2$Var1, levels = d_vis_2$Var1[order(-d_vis_2$Freq)])
p<-ggplot(d_vis_2, aes(x=Var1,y=Freq,fill=Var1)) +
 geom_bar(stat='identity') +theme(axis.text.x = element_text(angle = 90, hjust = 1))
print(p)
```



By the graphical output above sales, technical, and support department were the top 3 departments to have employee turnover whiles the management department had the smallest amount of turnover.

3 Data Partitioning

```
#Partitioning of Data
partition.scale <- function(dat, xcols, percent.train=0.67, seed=0, scale=FALSE){
  set.seed(seed)
```

```
  n <- NROW(dat)
  id.train <- sample(1:n, trunc(n*percent.train), replace=FALSE)
  train <- dat[id.train,]; test <- dat[-id.train,]

  if (scale) {
    X.train <- train[, xcols]; X.test <- test[, xcols]
    scale.train <- scale(train[, xcols], center=TRUE, scale = TRUE)
    train[, xcols] <- as.data.frame(scale.train)
    test[, xcols] <- as.data.frame(scale(test[, xcols],
                          center=attributes(scale.train)$`scaled:center`,
                          scale=attributes(scale.train)$`scaled:scale`))
  }
  return(list(train=train, test=test))
}
```

```
xcols <-1:9
ps <- partition.scale(dat=hr.Data, xcols=xcols, percent.train=0.67, seed=123)
TrainData <- ps$train; TestData <- ps$test
dim(TrainData); dim(TestData)
```

```
## [1] 10049    10
```

```
## [1] 4950   10
```

The dataset is partitioned with the TrainData having 10049 observations whiles the TestData had 4950.

4 Methodology

In the steps to follow, we will train several classifiers with D1 and then apply each trained model on D2 to predict whether an employee will quit his/her current position or its likelihood. For each approach, obtain the ROC curve and the corresponding AUC based on the prediction on D2:

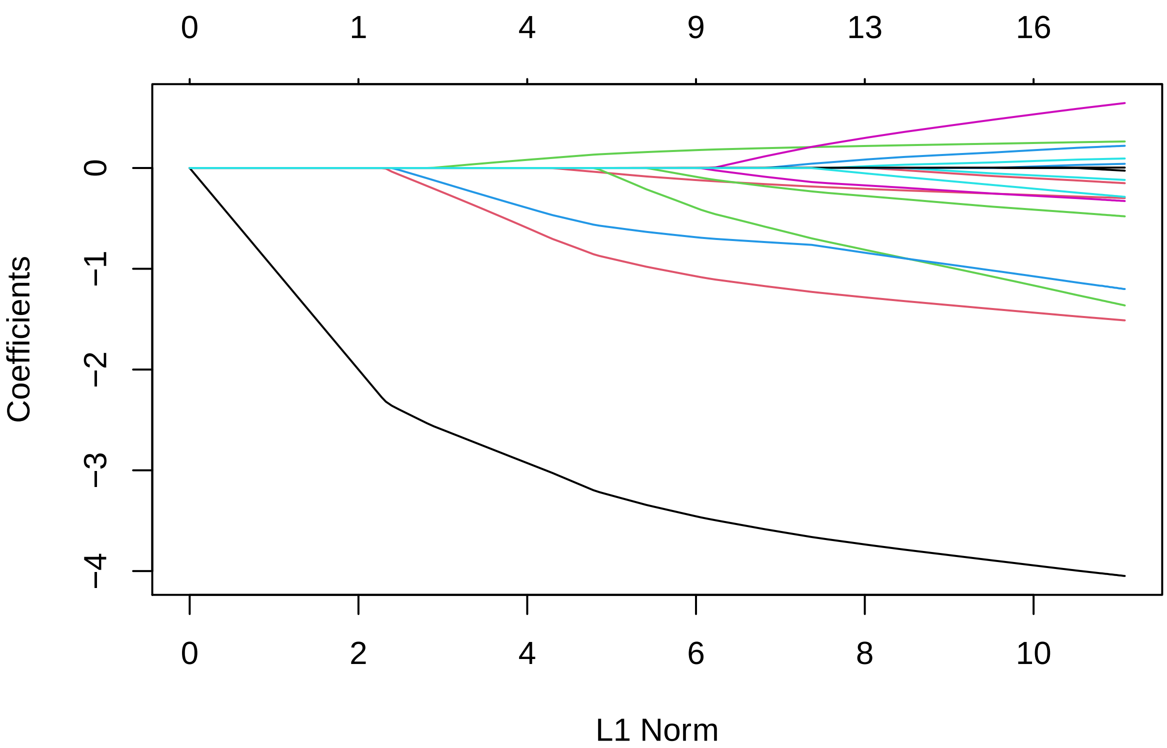
4.1 Logistic Regression

```
# Using LASSO
set.seed(123)
library(glmnet)
```

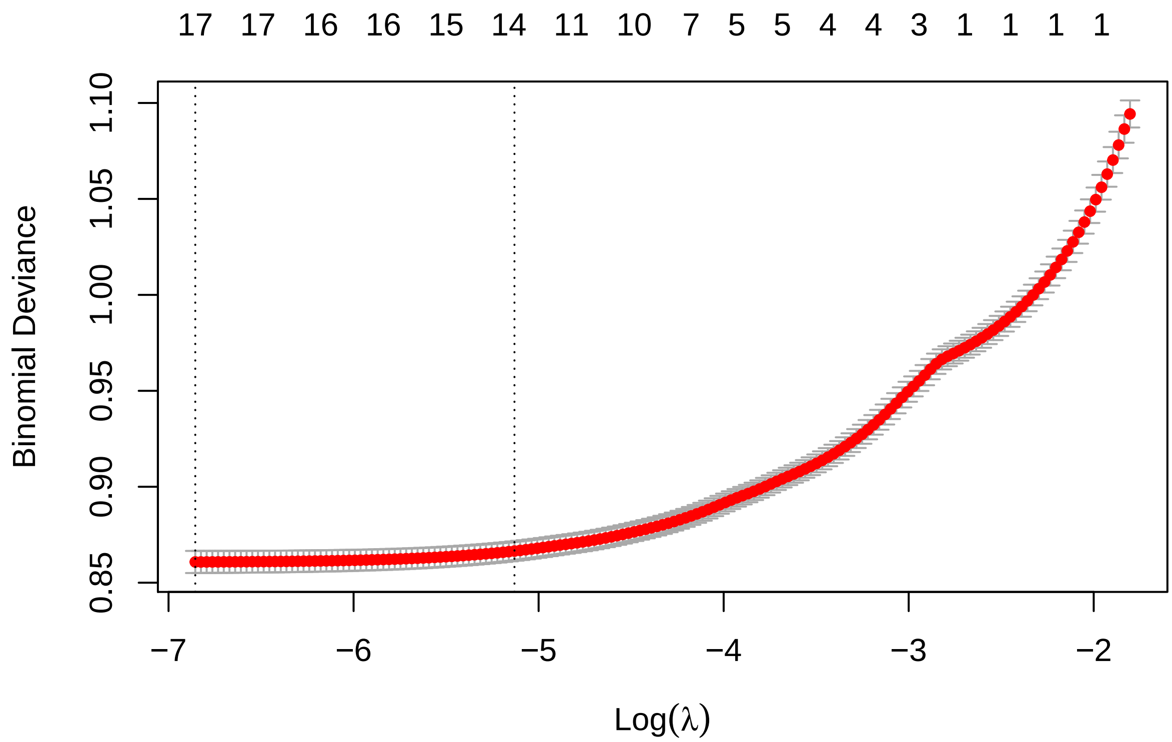
```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.0-2
```

```
X <- model.matrix(object=~ satisfaction_level + number_project +  time_spend_company +
factor(department) + last_evaluation +  average_montly_hours + Work_accident + promotion
y <- TrainData$left
fit.lasso <- glmnet(x=X, y=y, family="binomial", alpha=1,
    lambda.min = 1e-4, nlambda = 300, standardize=T, thresh = 1e-07,
    maxit=3000)
plot(fit.lasso)
```




```
CV <- cv.glmnet(x=X, y=y, family="binomial", alpha = 1,
    lambda.min = 1e-4, nlambda = 300, standardize = T, thresh = 1e-07,
    maxit=3000)
CV
plot(CV)
```



Given the output graph of the LASSO, two models were found to be statistically significant but due to the law of parsimony the model with 14 variables is chosen.

```
# SELECTING THE BEST TUNING PARAMETER
b.lambda <- CV$lambda.1se; b.lambda  # THE BEST lamdba WITH 1SE RULE
```

```
## [1] 0.00591304
```

```
fit.lasso <- glmnet(x=X, y=y, family="binomial", alpha = 1,
    lambda=b.lambda, standardize = T, thresh = 1e-07,
    maxit=1000)
names(fit.lasso)
```

```
##  [1] "a0"         "beta"       "df"         "dim"        "lambda"
##  [6] "dev.ratio"  "nulldev"    "npasses"    "jerr"       "offset"
## [11] "classnames" "call"       "nobs"
```

```
fit.lasso$beta
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                                   s0
## (Intercept)                    .
## satisfaction_level          -3.757965768
```

```
## number_project              -0.213756849
## time_spend_company           0.221864599
## factor(department)hr         0.095637767
## factor(department)IT          .
## factor(department)management -0.182870504
## factor(department)marketing   .
## factor(department)product_mng -0.008437540
## factor(department)RandD      -0.292144828
## factor(department)sales       .
## factor(department)support     .
## factor(department)technical   0.024440072
## last_evaluation              0.325606985
## average_montly_hours         0.002972647
## Work_accident               -1.299648328
## promotion_last_5years       -0.848138360
## factor(salary).L            -0.866217251
## factor(salary).Q            -0.069462175
```

```r
fit.pen.lasso <- glm(factor(left) ~ satisfaction_level + number_project   + time_spend_c
department + last_evaluation +  average_montly_hours + Work_accident + promotion_last_5y
family = binomial, data=TrainData)
```

The tuning parameter was selected by using the largest value of lambda such that error is within 1 standard error of the minimum. From the graph above we observe that 14 variables are selected with this choice of lamdba (obtained via cross validation).

```r
summary(fit.pen.lasso)
```

```
##
## Call:
## glm(formula = factor(left) ~ satisfaction_level + number_project +
##     time_spend_company + department + last_evaluation + average_montly_hours +
##     Work_accident + promotion_last_5years + salary, family = binomial,
##     data = TrainData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2448  -0.6625  -0.4021  -0.1213   3.0979
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -0.2510011  0.1880453  -1.335 0.181945
## satisfaction_level    -4.1198868  0.1194741 -34.484  < 2e-16 ***
## number_project        -0.3190040  0.0258427 -12.344  < 2e-16 ***
```

```
## time_spend_company     0.2737836  0.0191794  14.275  < 2e-16 ***
## departmenthr           0.1655528  0.1597722   1.036 0.300118
## departmentIT          -0.2316803  0.1469363  -1.577 0.114855
## departmentmanagement  -0.4451795  0.1892858  -2.352 0.018678 *
## departmentmarketing   -0.1424082  0.1606205  -0.887 0.375287
## departmentproduct_mng -0.2665583  0.1583850  -1.683 0.092379 .
## departmentRandD       -0.6073774  0.1781076  -3.410 0.000649 ***
## departmentsales       -0.0966579  0.1234719  -0.783 0.433725
## departmentsupport     -0.0245019  0.1321732  -0.185 0.852933
## departmenttechnical    0.0275679  0.1288409   0.214 0.830571
## last_evaluation        0.7185780  0.1816964   3.955 7.66e-05 ***
## average_montly_hours   0.0043616  0.0006263   6.965 3.30e-12 ***
## Work_accident         -1.5643620  0.1099761 -14.225  < 2e-16 ***
## promotion_last_5years -1.5124724  0.3234141  -4.677 2.92e-06 ***
## salary.L              -1.2972885  0.1088966 -11.913  < 2e-16 ***
## salary.Q              -0.3453375  0.0712525  -4.847 1.26e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 11004.2  on 10048  degrees of freedom
## Residual deviance:  8607.9  on 10030  degrees of freedom
## AIC: 8645.9
##
## Number of Fisher Scoring iterations: 5
```

Obtaining the 95% confidence intervals for coeffcients β_j 's:

```
confint(fit.pen.lasso, level=0.95)
```

```
## Waiting for profiling to be done...
```

```
##                              2.5 %       97.5 %
## (Intercept)           -0.620714131  0.116588446
## satisfaction_level    -4.355714126 -3.887323243
## number_project        -0.369878341 -0.268562465
## time_spend_company     0.236201165  0.311406279
## departmenthr          -0.147694113  0.478878087
## departmentIT          -0.519337927  0.056882253
## departmentmanagement  -0.819763918 -0.077104769
## departmentmarketing   -0.457784868  0.172142445
## departmentproduct_mng -0.577532709  0.043631764
## departmentRandD       -0.959327676 -0.260623981
```

```
## departmentsales       -0.337029030  0.147199320
## departmentsupport     -0.282298755  0.236026616
## departmenttechnical   -0.223520182  0.281743749
## last_evaluation        0.362909236  1.075253269
## average_montly_hours   0.003136496  0.005591739
## Work_accident         -1.784621650 -1.353168486
## promotion_last_5years -2.195828120 -0.917957511
## salary.L              -1.517370966 -1.089835488
## salary.Q              -0.487873193 -0.208208762
```

Estimating(Obtaining) the associated odds ratio and the 95% confidence intervals for the odds ratio:

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

```
## Waiting for profiling to be done...
```

```
##                               OR      2.5 %     97.5 %
## (Intercept)           0.77802155 0.53756041 1.12365689
## satisfaction_level    0.01624635 0.01283327 0.02050015
## number_project        0.72687265 0.69081837 0.76447767
## time_spend_company    1.31493018 1.26642905 1.36534382
## departmenthr          1.18004530 0.86269496 1.61426232
## departmentIT          0.79319966 0.59491429 1.05853116
## departmentmanagement  0.64070925 0.44053564 0.92579285
## departmentmarketing   0.86726717 0.63268357 1.18784702
## departmentproduct_mng 0.76601132 0.56128150 1.04459763
## departmentRandD       0.54477773 0.38315040 0.77057061
## departmentsales       0.90786655 0.71388812 1.15858487
## departmentsupport     0.97579583 0.75404838 1.26620801
## departmenttechnical   1.02795146 0.79969875 1.32543903
## last_evaluation       2.05151391 1.43750538 2.93073507
## average_montly_hours  1.00437108 1.00314142 1.00560740
## Work_accident         0.20922146 0.16786056 0.25842016
## promotion_last_5years 0.22036447 0.11126638 0.39933384
## salary.L              0.27327178 0.21928764 0.33627181
## salary.Q              0.70798139 0.61393072 0.81203750
```

From the above, All the variables which excludes 1 in the CI are significant.

Interpretation of odds ratio for satisfaction level: The estimated odds for satisfaction_level is exp(-4.1198868) = 0.01624635 . For each increase in 1 unit of satisfaction_level, the estimated odds of an employee turnover decreases by a factor of 0.016 regardless of the other predictors.

ROC Curve:

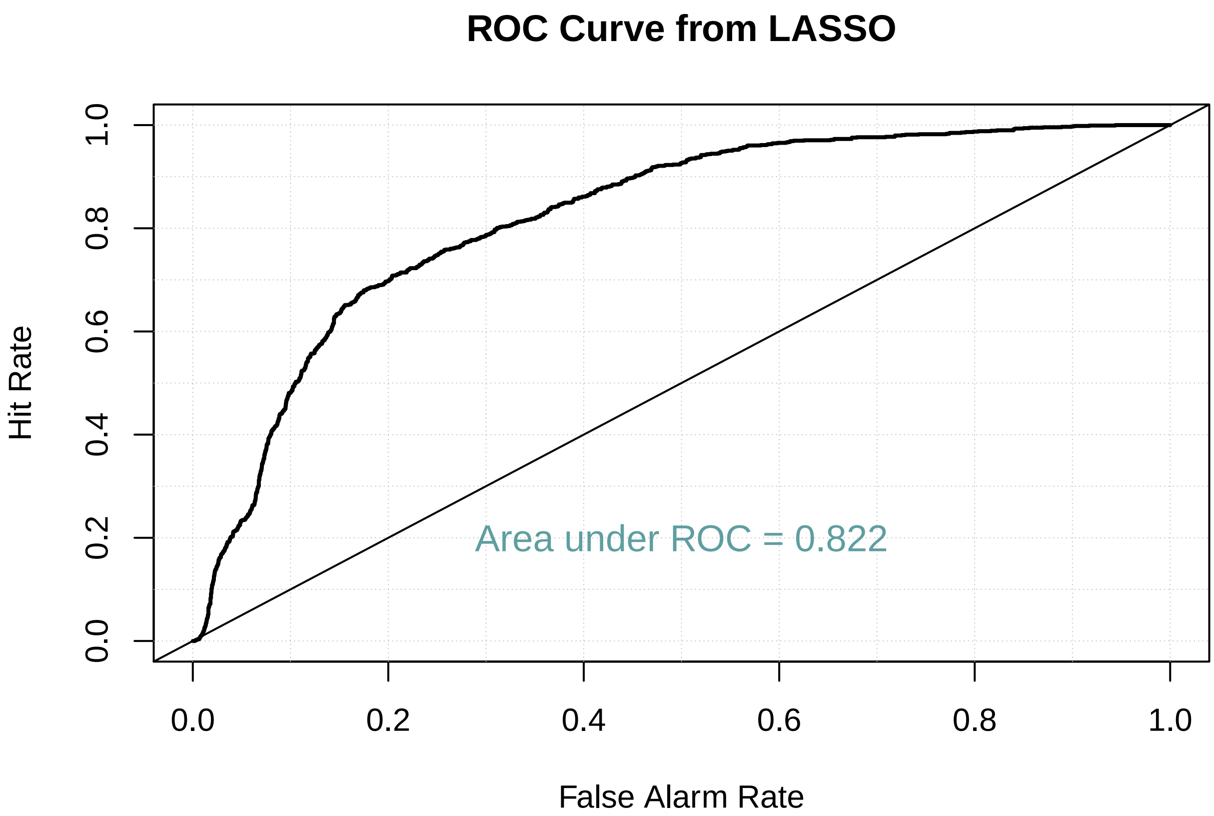
```
library(cvAUC)
library(verification)
n <- NROW(TestData)
yobs <- TestData$left
yhat.lasso <- predict(fit.pen.lasso, newdata=TestData, type="response")
AUC.lasso  <- ci.cvAUC(predictions=yhat.lasso, labels=yobs, folds=1:n, confidence=0.95);
```

```
## $cvAUC
## [1] 0.8217913
##
## $se
## [1] 0.006508279
##
## $ci
## [1] 0.8090353 0.8345473
##
## $confidence
## [1] 0.95
```

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

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

```
roc.plot(mod.glm, plot.thres = NULL, main="ROC Curve from LASSO")
text(x=0.5, y=0.2, paste("Area under ROC =", round(AUC.lasso$cvAUC, digits=3),
    sep=" "), col="cadetblue", cex=1.2)
```



The LASSO gives the area under ROC value of 0.822.

4.2 Random Forest

```
library(randomForest)
fit.rf <- randomForest(factor(left) ~., data=TrainData,importance=TRUE, proximity=TRUE,
fit.rf;
```

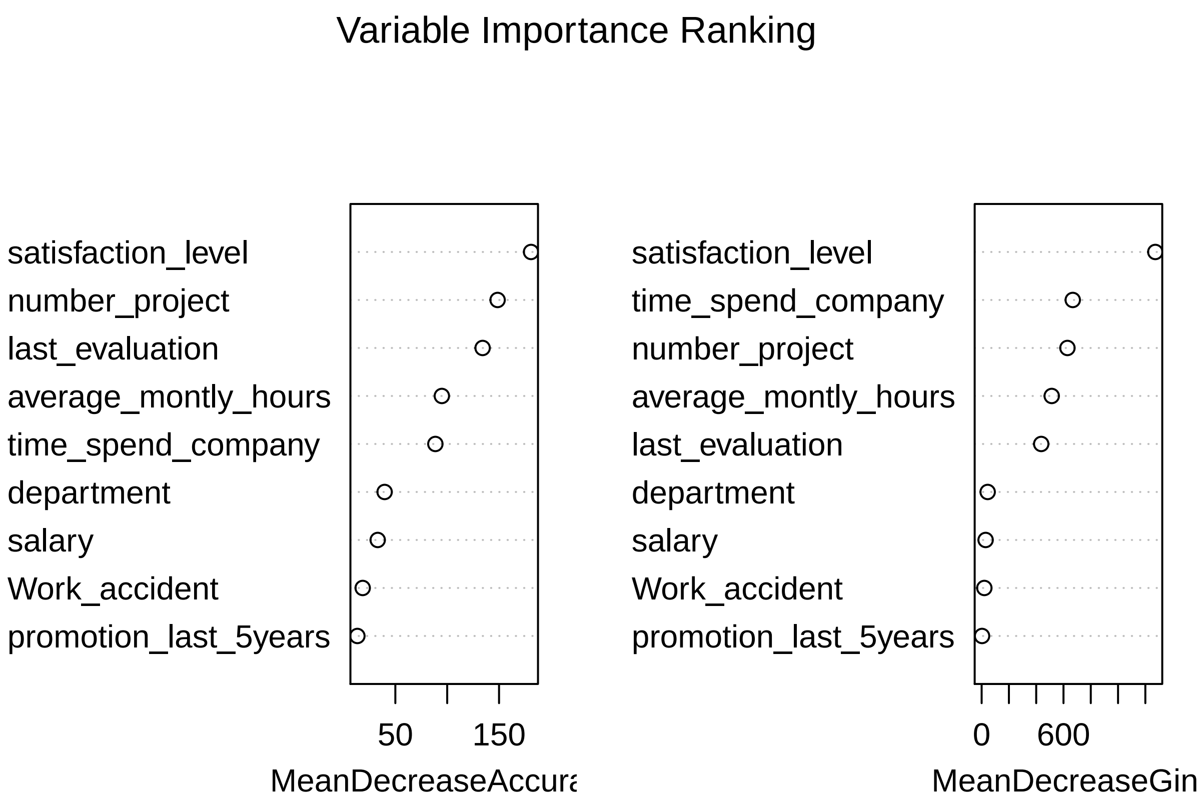
```
##
## Call:
##  randomForest(formula = factor(left) ~ ., data = TrainData, importance = TRUE,      p
##                Type of random forest: classification
##                      Number of trees: 400
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 0.94%
## Confusion matrix:
##      0    1 class.error
## 0 7655   13 0.001695357
## 1   81 2300 0.034019320
```

```
yhat.Random <- predict(fit.rf, newdata=TestData, type="prob")[, 2]
```

```
# VARIABLE IMPORTANCE RANKING
round(importance(fit.rf), 2)
```

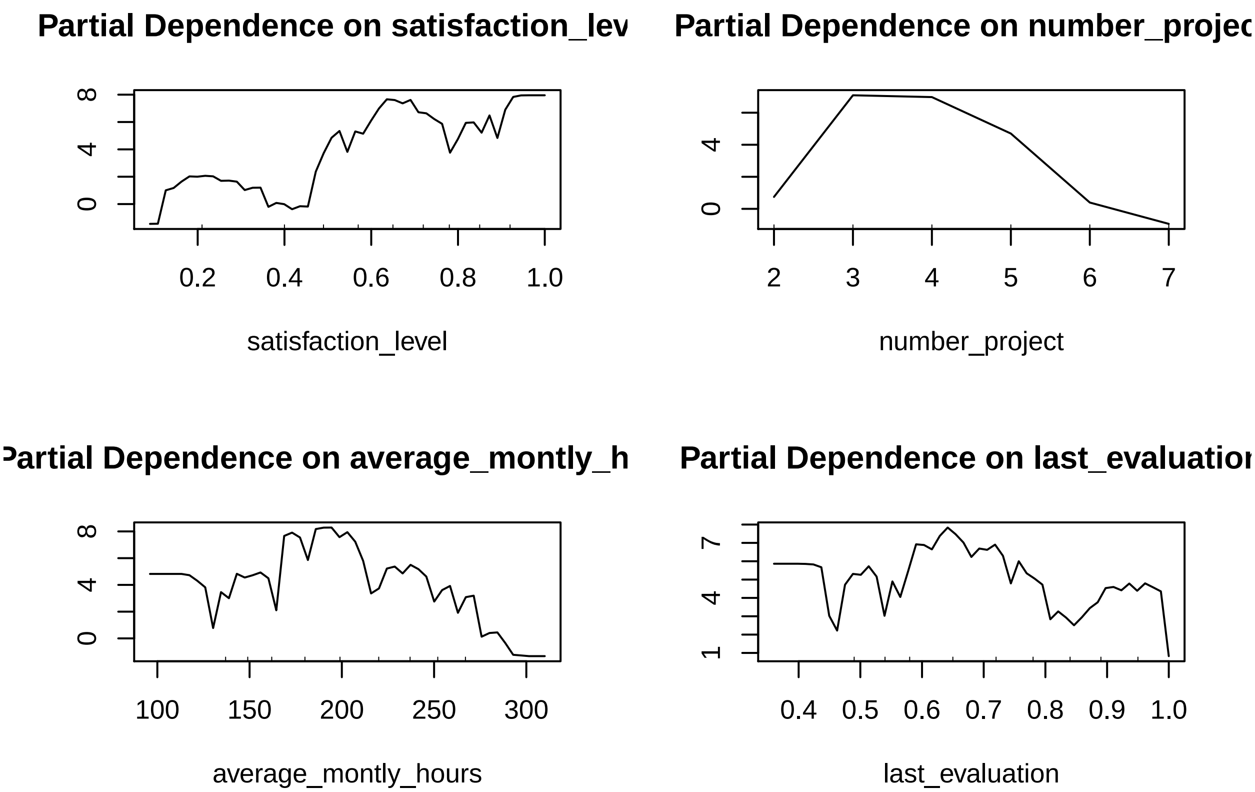
```
##                            0      1 MeanDecreaseAccuracy MeanDecreaseGini
## satisfaction_level     63.58 192.69               180.86          1273.01
## last_evaluation        23.55 135.85               134.15           436.76
## number_project         38.58 150.15               148.60           628.90
## average_montly_hours   48.23  86.31                94.84           513.80
## time_spend_company     52.59  80.74                88.60           667.89
## Work_accident           8.65  18.60                18.50            19.78
## promotion_last_5years   7.80  11.54                13.37             2.96
## department             11.13  54.18                39.64            44.32
## salary                 11.73  37.18                32.99            29.23
```

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




```
# PARTIAL DEPENDENCE PLOT
par(mfrow=c(2,2))
partialPlot(fit.rf, pred.data=TrainData, x.var=satisfaction_level, rug=TRUE)
```

```r
partialPlot(fit.rf, pred.data=TrainData, x.var=number_project, rug=TRUE)
partialPlot(fit.rf, pred.data=TrainData, x.var=average_montly_hours, rug=TRUE)
partialPlot(fit.rf, pred.data=TrainData, x.var=last_evaluation, rug=TRUE)
```

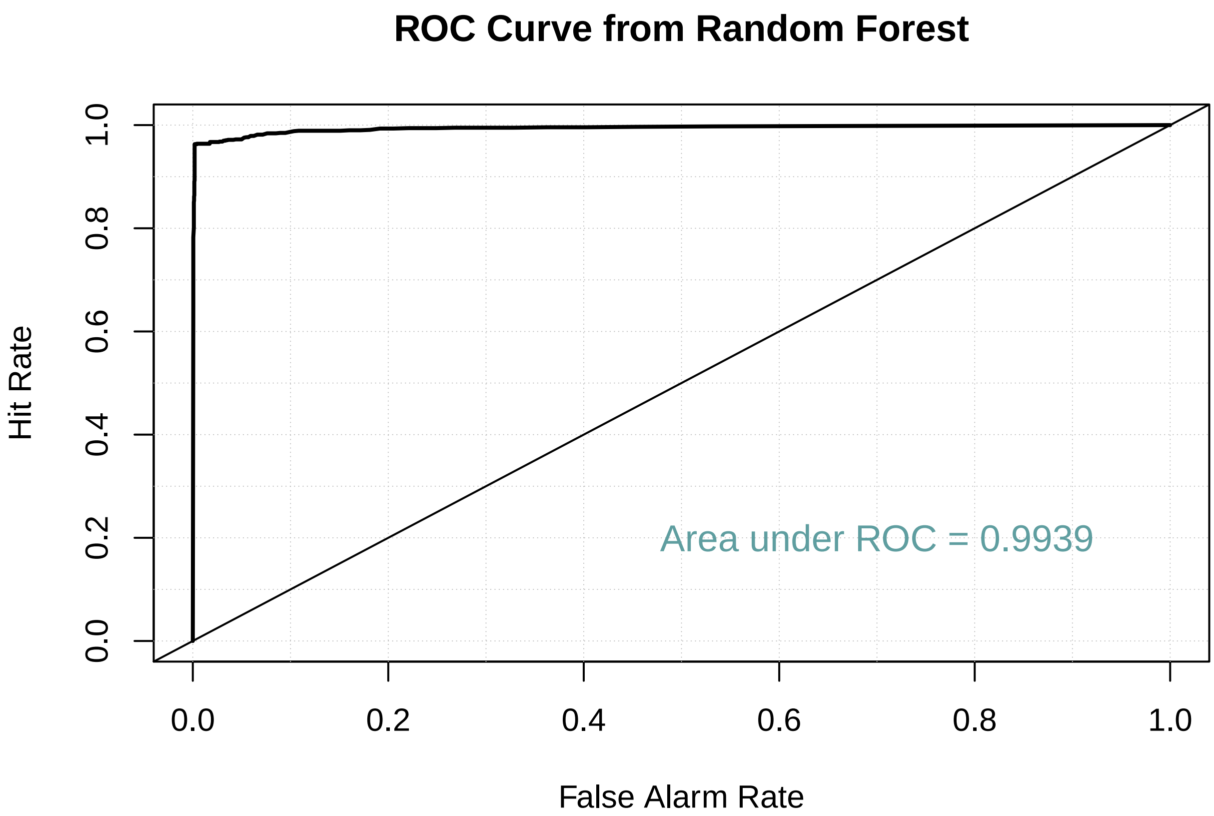


Based on the MeanDecreaseAccuracy, the top two variables according to the variable importance ranking for random forest are satisfaction_level and number_project. The least significant variable is promotion_last_5years.

```r
AUC.RF <- roc.area(obs=yobs, pred=yhat.Random)$A
mod.rf <- verify(obs=yobs, pred=yhat.Random)
```

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

```r
roc.plot(mod.rf, plot.thres = NULL, col="red", main="ROC Curve from Random Forest")
text(x=0.7, y=0.2, paste("Area under ROC =", round(AUC.RF, digits=4),
    sep=" "), col="cadetblue", cex=1.2)
```



The RF gives the area under ROC value of 0.9939.

4.3 Generalized Additive Model

```
library(gam)
fit.gam <- gam( left ~ satisfaction_level + number_project +   + time_spend_company +
department + last_evaluation +  average_montly_hours + Work_accident + promotion_last_5y
+ salary , family = binomial,
    data=TrainData, trace=TRUE,
    control = gam.control(epsilon=1e-04, bf.epsilon = 1e-04, maxit=50, bf.maxit = 50))
summary(fit.gam)
```

```
##
## Call: gam(formula = left ~ satisfaction_level + number_project + +time_spend_company
##     department + last_evaluation + average_montly_hours + Work_accident +
##     promotion_last_5years + salary, family = binomial, data = TrainData,
##     control = gam.control(epsilon = 1e-04, bf.epsilon = 1e-04,
##         maxit = 50, bf.maxit = 50), trace = TRUE)
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2447 -0.6625 -0.4021 -0.1213  3.0974
##
```

```
## (Dispersion Parameter for binomial family taken to be 1)
##
##     Null Deviance: 11004.17 on 10048 degrees of freedom
## Residual Deviance: 8607.933 on 10030 degrees of freedom
## AIC: 8645.933
##
## Number of Local Scoring Iterations: 4
##
## Anova for Parametric Effects
##                          Df Sum Sq Mean Sq  F value    Pr(>F)
## satisfaction_level        1  979.9  979.87 1040.456 < 2.2e-16 ***
## number_project            1   28.6   28.58   30.344 3.707e-08 ***
## time_spend_company        1  124.1  124.08  131.755 < 2.2e-16 ***
## department                9   45.0    5.00    5.311 2.903e-07 ***
## last_evaluation           1   30.2   30.24   32.110 1.497e-08 ***
## average_montly_hours      1   43.6   43.58   46.270 1.089e-11 ***
## Work_accident             1  189.6  189.56  201.284 < 2.2e-16 ***
## promotion_last_5years     1   27.9   27.86   29.580 5.491e-08 ***
## salary                    2  193.1   96.56  102.533 < 2.2e-16 ***
## Residuals             10030 9445.9    0.94
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
yhat.gam <- predict(fit.gam, newdata=TestData, type="response", se.fit=FALSE)
```

Model Selection:

```
# STEPWISE SELECTION
fit.step <- step.Gam(fit.gam, scope=list("satisfaction_level"=~1 +satisfaction_level + l
                "last_evaluation"=~1+ last_evaluation + lo(last_evaluation)+ s(last_eval
                "number_project"=~1 + number_project + s(number_project, 2) + s(number_p
                    "average_montly_hours"=~1 + average_montly_hours + s(average_montly_
    "time_spend_company"=~1 + time_spend_company + s(time_spend_company, 2) + s(time_spe
            scale =2, steps=1000, parallel=TRUE, direction="both")
```

```
## Start:  left ~ satisfaction_level + number_project + +time_spend_company +      depar
```

```
## Warning: executing %dopar% sequentially: no parallel backend registered
```

```
## Step:1 left ~ salary + satisfaction_level + last_evaluation + s(number_project,
## Step:2 left ~ salary + satisfaction_level + last_evaluation + s(number_project,
## Step:3 left ~ salary + lo(satisfaction_level) + last_evaluation + s(number_project,
## Step:4 left ~ salary + lo(satisfaction_level) + lo(last_evaluation) +      s(number_p
```

```
## Step:5 left ~ salary + lo(satisfaction_level) + lo(last_evaluation) +     s(number_p
## Step:6 left ~ salary + lo(satisfaction_level) + lo(last_evaluation) +     s(number_p
## Step:7 left ~ salary + lo(satisfaction_level) + lo(last_evaluation) +     s(number_p
## Step:8 left ~ salary + lo(satisfaction_level) + lo(last_evaluation) +     s(number_p
```

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

```
##
## Call: gam(formula = left ~ salary + lo(satisfaction_level) + lo(last_evaluation) +
##     s(number_project, 4) + s(average_montly_hours, 4) + s(time_spend_company,
##     4), family = binomial, data = TrainData, control = gam.control(epsilon = 1e-04,
##     bf.epsilon = 1e-04, maxit = 50, bf.maxit = 50), trace = FALSE)
## Deviance Residuals:
##       Min        1Q    Median        3Q       Max
## -3.158039 -0.326463 -0.137658 -0.004606  3.563409
##
## (Dispersion Parameter for binomial family taken to be 1)
##
##     Null Deviance: 11004.17 on 10048 degrees of freedom
## Residual Deviance: 4273.194 on 10027.14 degrees of freedom
## AIC: 4316.917
##
## Number of Local Scoring Iterations: 1
##
## Anova for Parametric Effects
##                               Df Sum Sq Mean Sq F value    Pr(>F)
## salary                         2   81.3   40.64  41.939 < 2.2e-16 ***
## lo(satisfaction_level)         1   18.8   18.84  19.444 1.047e-05 ***
## lo(last_evaluation)            1  140.7  140.73 145.232 < 2.2e-16 ***
## s(number_project, 4)           1   47.1   47.11  48.612 3.316e-12 ***
## s(average_montly_hours, 4)     1   95.9   95.94  99.006 < 2.2e-16 ***
## s(time_spend_company, 4)       1  340.8  340.76 351.660 < 2.2e-16 ***
## Residuals                  10027 9716.3    0.97
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                            Npar Df Npar Chisq    P(Chi)
## (Intercept)
## salary
## lo(satisfaction_level)         2.4     393.63 < 2.2e-16 ***
## lo(last_evaluation)            2.5     358.02 < 2.2e-16 ***
## s(number_project, 4)           3.0     945.58 < 2.2e-16 ***
## s(average_montly_hours, 4)     3.0     370.13 < 2.2e-16 ***
```

```
## s(time_spend_company, 4)       3.0     297.88 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

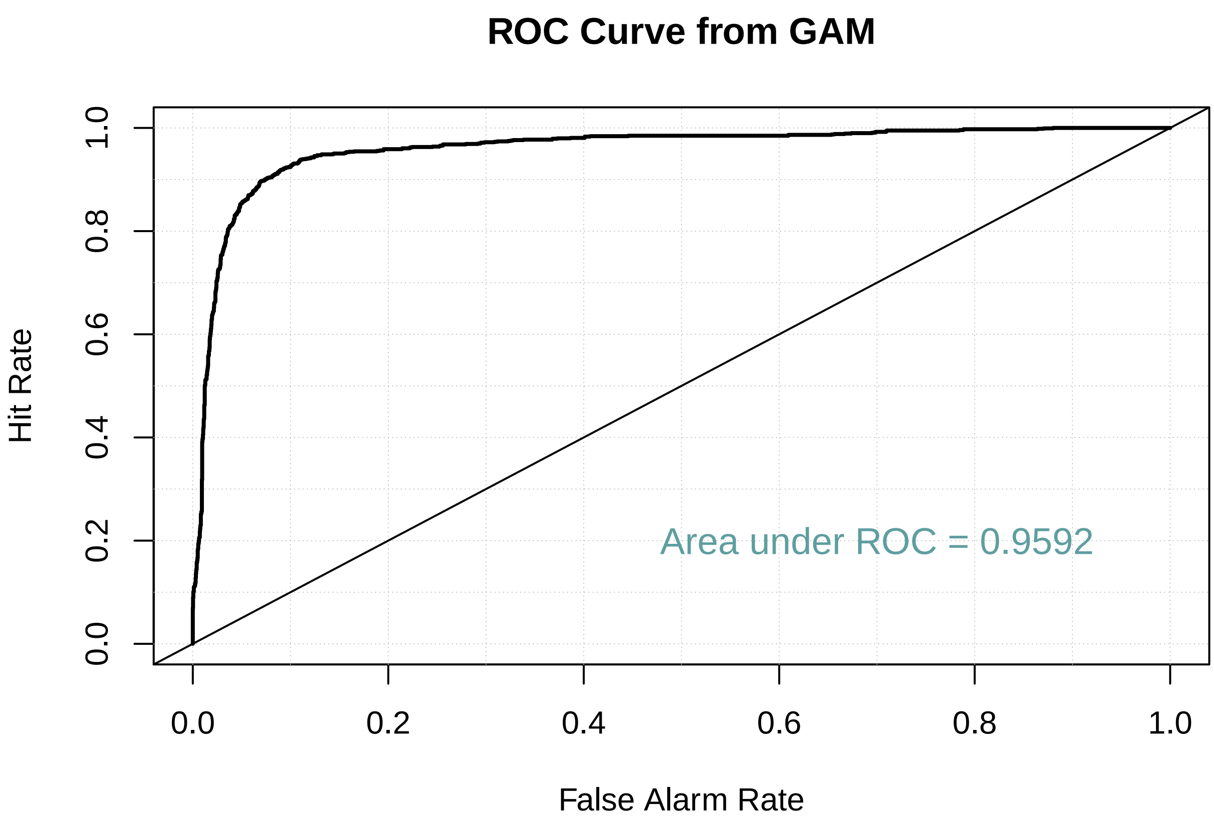
```
yhat.gam <- predict(fit.step, newdata=TestData, type="response", se.fit=FALSE)
AUC.GAM <- roc.area(obs=yobs, pred=yhat.gam)$A
mod.gam <- verify(obs=yobs, pred=yhat.gam)
```

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

```
roc.plot(mod.gam, plot.thres = NULL, col="red", main="ROC Curve from GAM")
```

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

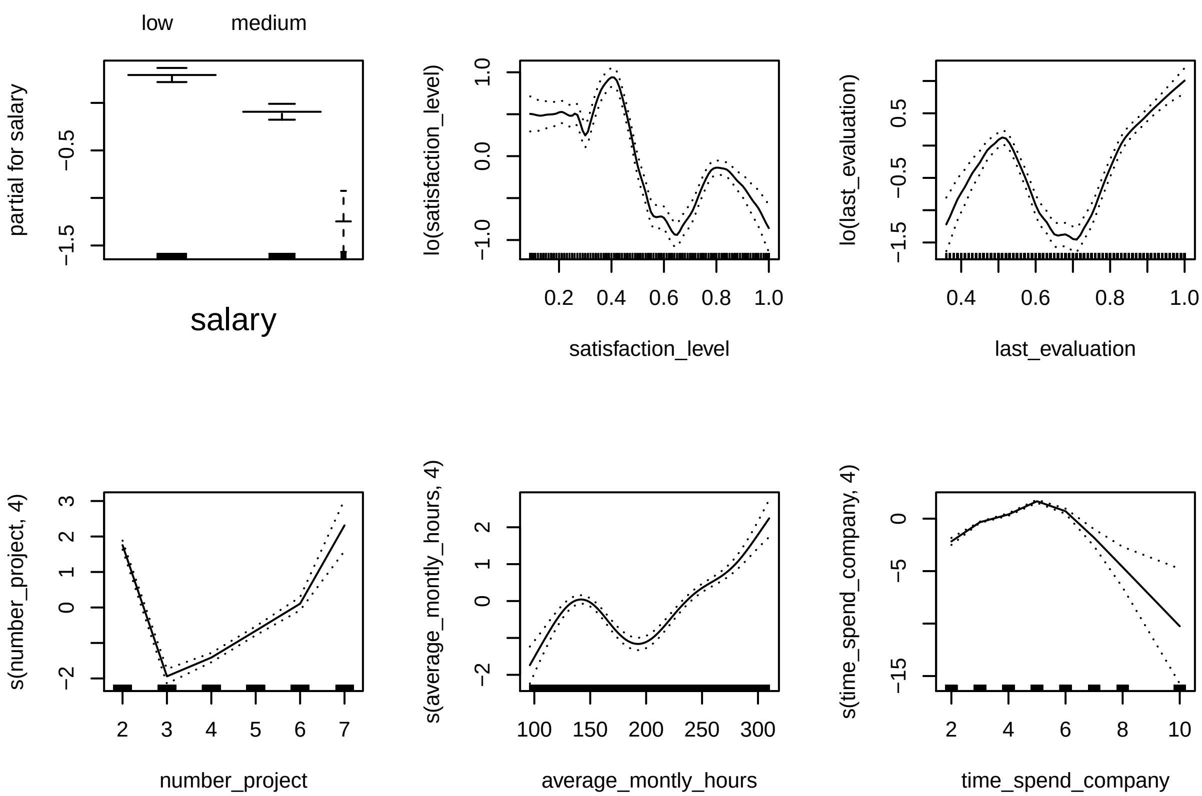
```
text(x=0.7, y=0.2, paste("Area under ROC =", round(AUC.GAM, digits=4),
    sep=" "), col="cadetblue", cex=1.2)
```



The GAM gives the area under ROC value of 0.9592.

Plotting the (nonlinear) functional forms for continuous predictors.

```r
par(mfrow=c(2,3))
plot(fit.step, se =TRUE)
```



Each smoothing parameter was determined adaptively in the backfitting algorithm. In this scenario since smoothing splines are used, optimization of the tuning parameter is automatically done via minimum GCV.Also Stepwise selection with AIC was used to do the variable selection.

4.4 Multivariate Adaptive Regression Splines

```r
library("earth")
library(ggplot2)    # plotting
library(caret)      # automating the tuning process
library(vip)        # variable importance
library(pdp)        # variable relationships
fit.mars <- earth(left ~ .,  data = TrainData, degree=3,
    glm=list(family=binomial(link = "logit")))
print(fit.mars)
```

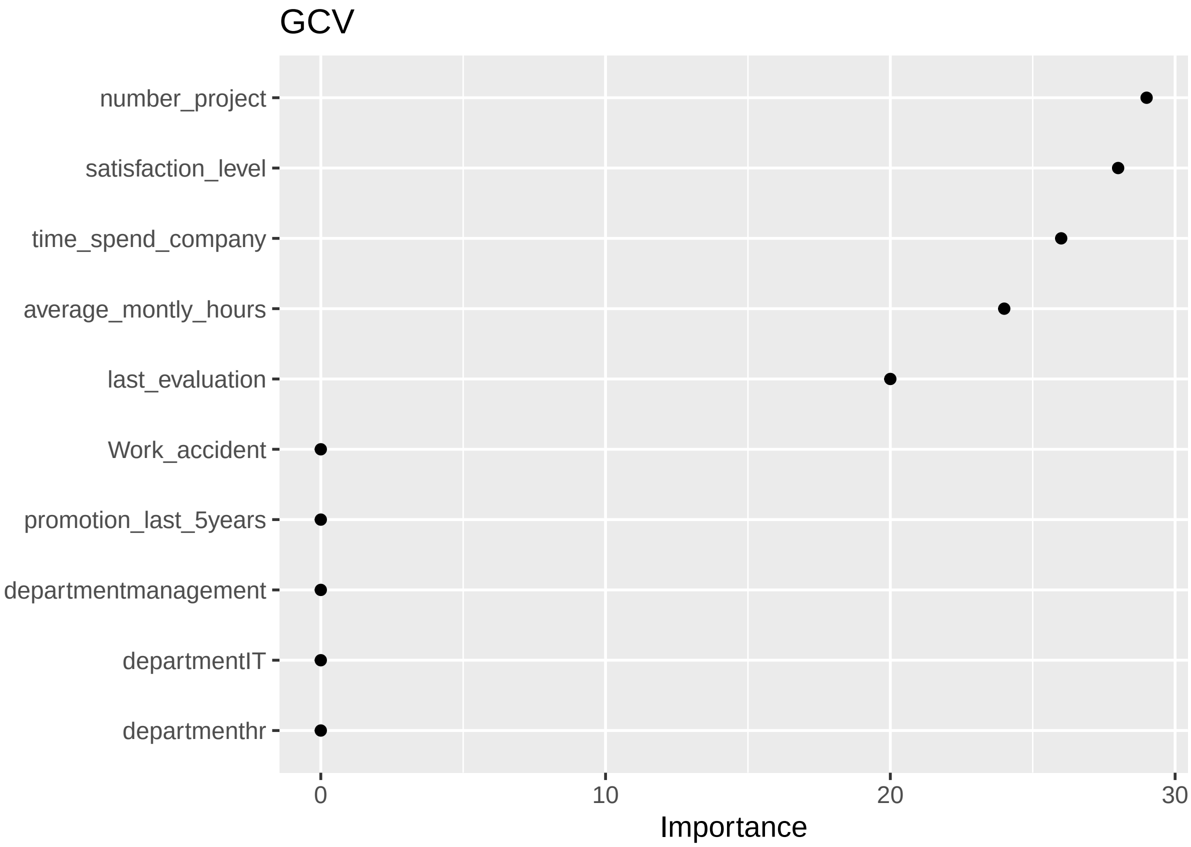
```
## GLM (family binomial, link logit):
##  nulldev    df       dev    df   devratio     AIC iters converged
```

```
##  11004.2 10048   2306.77 10019       0.79    2367    18          1
##
## Earth selected 30 of 34 terms, and 5 of 18 predictors
## Termination condition: Reached nk 37
## Importance: number_project, satisfaction_level, time_spend_company, ...
## Number of terms at each degree of interaction: 1 4 13 12
## Earth GCV 0.03658696    RSS 362.3037    GRSq 0.7976776    RSq 0.8005867
```

```
summary(fit.mars) %>% .$coefficients %>% head(10)
```

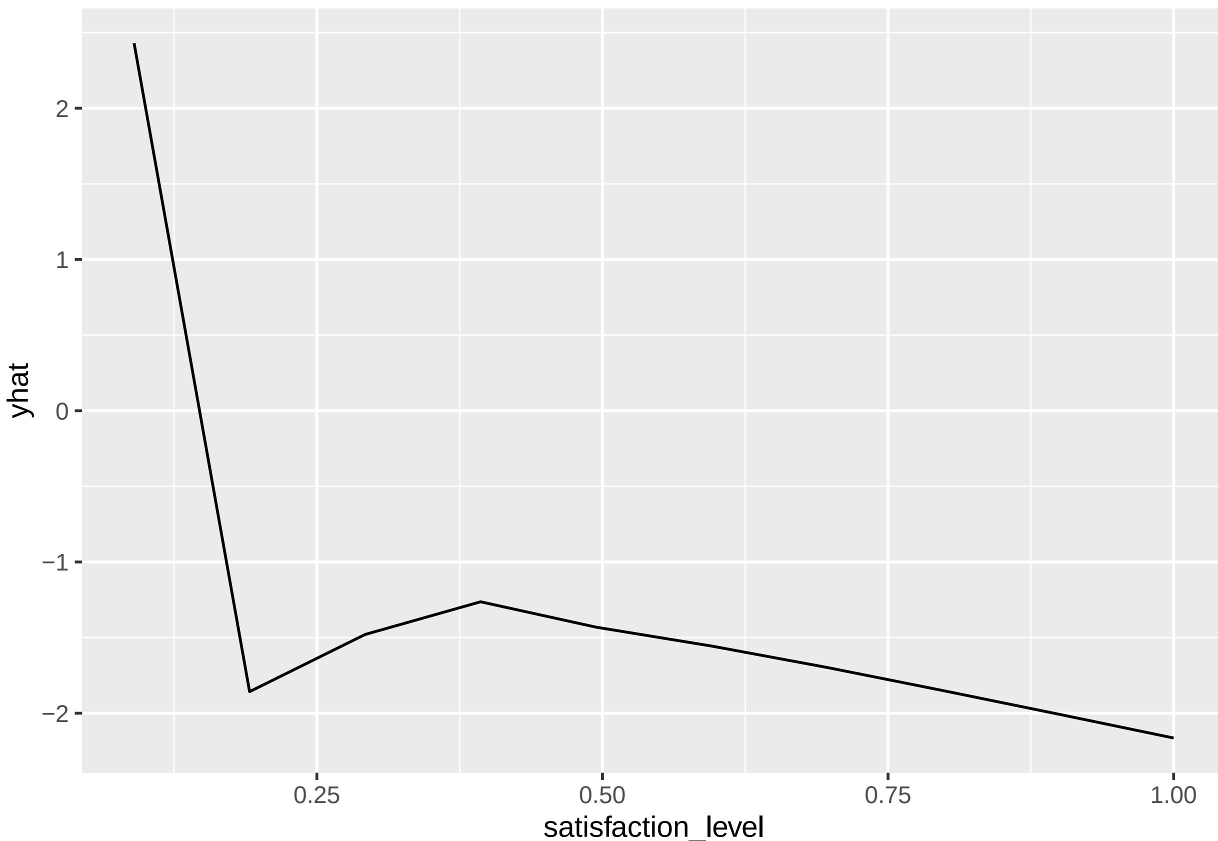
```
##                                                                         left
## (Intercept)                                                      -0.01568823
## h(number_project-3)                                               0.03297497
## h(3-number_project)                                               1.12381379
## h(number_project-3)*h(time_spend_company-5)                      -0.02043810
## h(number_project-3)*h(5-time_spend_company)                       0.02756807
## h(satisfaction_level-0.38)*h(3-number_project)                   -2.09913147
## h(0.38-satisfaction_level)*h(3-number_project)                   -2.23879040
## h(satisfaction_level-0.23)*h(number_project-3)                    0.14422928
## h(0.23-satisfaction_level)*h(number_project-3)                    0.37482872
## h(satisfaction_level-0.23)*h(last_evaluation-0.99)*h(number_project-3) 11.76623858
```

```
# VARIABLE IMPORTANCE PLOT
vip(fit.mars, num_features = 10, bar = FALSE) + ggtitle("GCV")
```

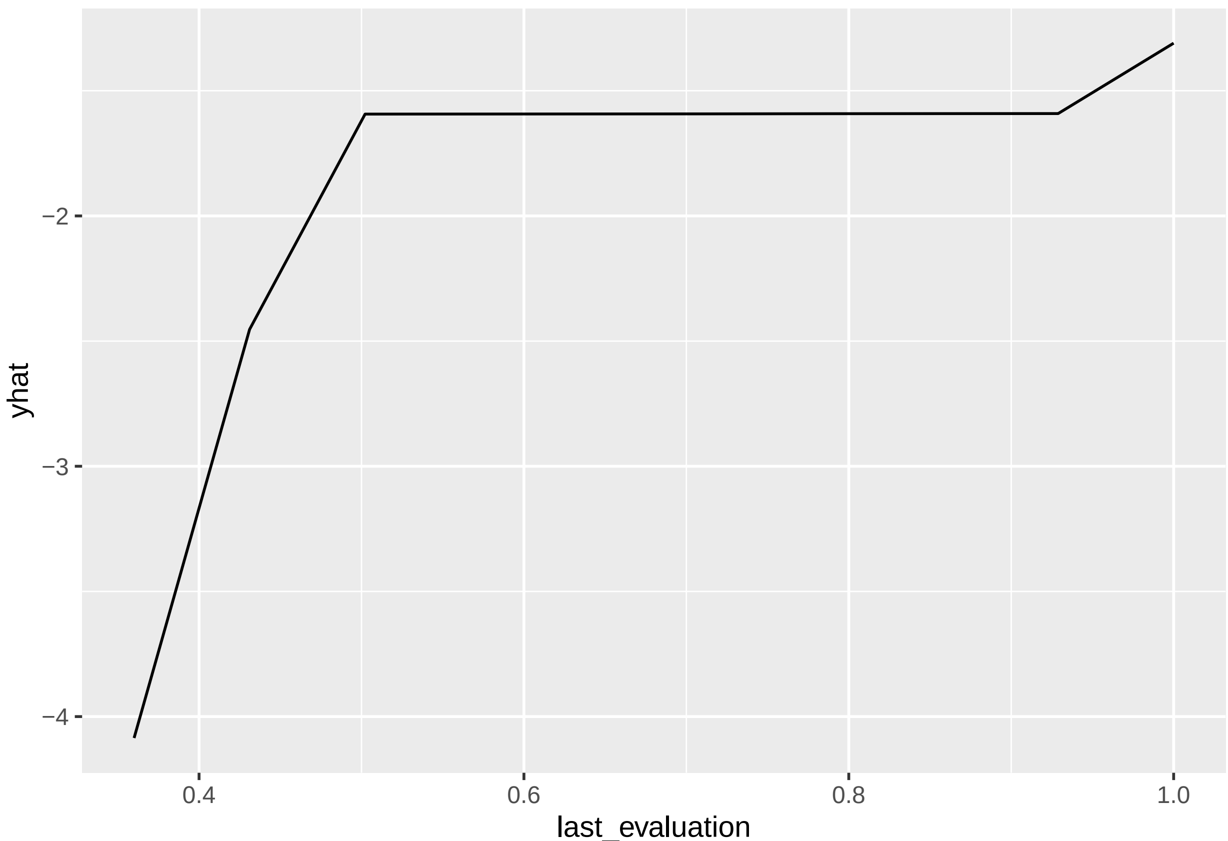


Given the graph, the two top important variables is satisfaction level and number of projects. This implies satisfaction level and number of projects are the two top variables that predict employee detention or turnover.

```
# PARTIAL DEPENDENCE PLOT
par(mfrow=c(1,2))
partial(fit.mars, pred.var = "satisfaction_level", grid.resolution = 10)%>%autoplot()
```




```
partial(fit.mars, pred.var = "last_evaluation", grid.resolution = 10)%>%autoplot()
```



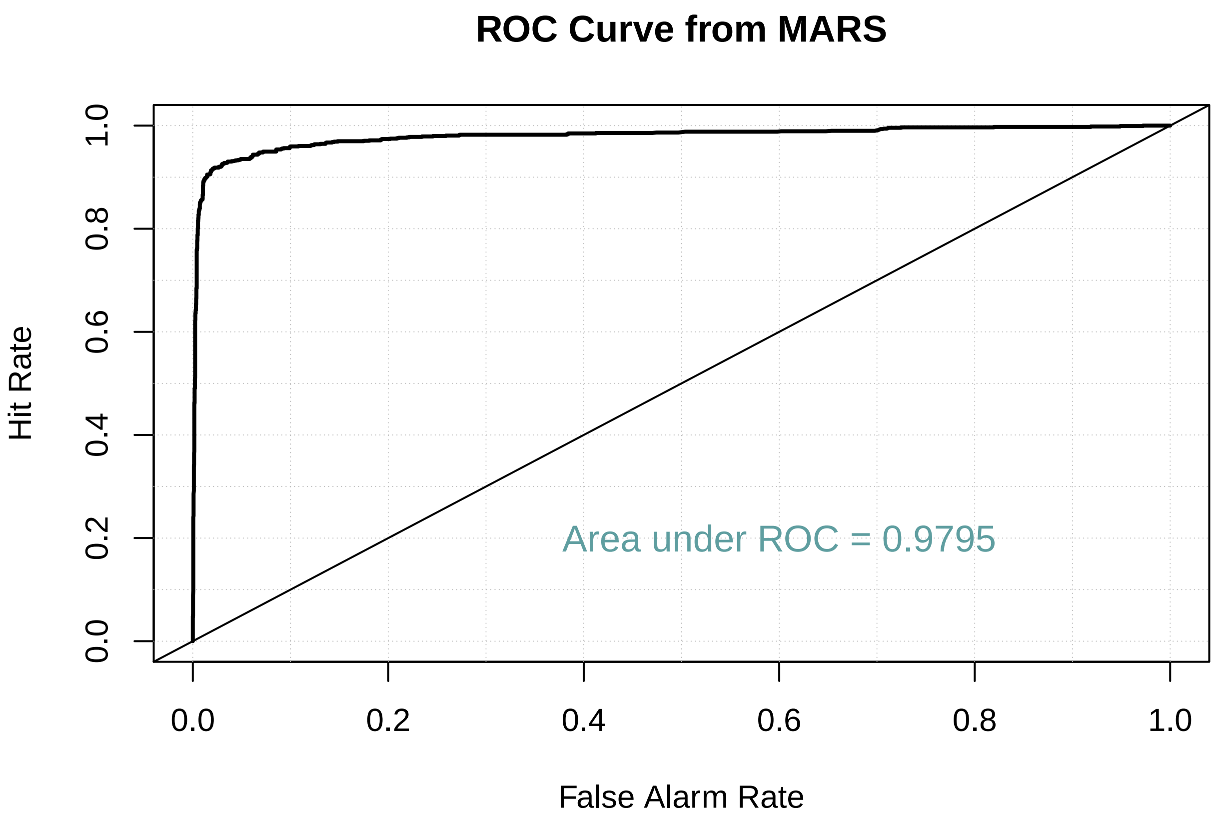

```r
# PREDICTION
yhat.mars <- predict(fit.mars, newdata=TestData, type="response")
AUC.MARS <- ci.cvAUC(predictions=yhat.mars, labels=yobs, folds=1:length(yhat.mars), conf
```

```
## $cvAUC
## [1] 0.9794854
##
## $se
## [1] 0.002764741
##
## $ci
## [1] 0.9740666 0.9849042
##
## $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
```

```
roc.plot(mod.mars, plot.thres = NULL, main="ROC Curve from MARS")
text(x=0.6, y=0.2, paste("Area under ROC =", round(AUC.MARS$cvAUC, digits=4),
    sep=" "), col="cadetblue", cex=1.2)
```



The GAM gives the area under ROC value of 0.9795.

4.5 Project Pursuit Regression

```
fit.ppr <- ppr(left ~ ., sm.method = "supsmu",
    data = TrainData, nterms = 2, max.terms = 10, bass=3)
summary(fit.ppr)
```

```
## Call:
## ppr(formula = left ~ ., data = TrainData, sm.method = "supsmu",
##     nterms = 2, max.terms = 10, bass = 3)
##
## Goodness of fit:
##  2 terms  3 terms  4 terms  5 terms  6 terms  7 terms  8 terms  9 terms
## 561.8795 451.9724 491.5016 490.6019   0.0000   0.0000   0.0000   0.0000
## 10 terms
##   0.0000
```

```
##
## Projection direction vectors ('alpha'):
##                         term 1        term 2
## satisfaction_level       0.0934098319  0.1562410392
## last_evaluation          0.1376469723  0.4083557351
## number_project           0.0296820724  0.0859886179
## average_montly_hours     0.0004083296  0.0013369214
## time_spend_company      -0.0212869596 -0.0041713341
## Work_accident           -0.0002034402 -0.0107587760
## promotion_last_5years   -0.0055850681 -0.0328441939
## departmentaccounting    -0.3086584082 -0.2785382820
## departmenthr            -0.3116257844 -0.2771259078
## departmentIT            -0.3132751979 -0.2843281361
## departmentmanagement    -0.3101856300 -0.2831861023
## departmentmarketing     -0.3130659073 -0.2860312097
## departmentproduct_mng   -0.3114104969 -0.2830139624
## departmentRandD         -0.3093558433 -0.2809265298
## departmentsales         -0.3134887179 -0.2871618137
## departmentsupport       -0.3132803064 -0.2867054996
## departmenttechnical     -0.3116289925 -0.2807107803
## salary.L                -0.0018357663 -0.0201410221
## salary.Q                -0.0013087138 -0.0105253805
##
## Coefficients of ridge terms ('beta'):
##    term 1    term 2
## 0.1776006 0.4071109
```

```
fit1.ppr <- update(fit.ppr, bass=5, nterms=4)
summary(fit1.ppr)
```

```
## Call:
## ppr(formula = left ~ ., data = TrainData, sm.method = "supsmu",
##     nterms = 4, max.terms = 10, bass = 5)
##
## Goodness of fit:
##  4 terms  5 terms  6 terms  7 terms  8 terms  9 terms 10 terms
## 517.2908 496.3734 447.4861 421.9938 433.5522 425.4056   0.0000
##
## Projection direction vectors ('alpha'):
##                         term 1        term 2        term 3        term 4
## satisfaction_level      -0.4979390759 -0.0613894088  0.6750761477  0.0483001989
## last_evaluation         -0.2234034347  0.2048552976 -0.2251208569  0.1465278883
## number_project           0.0208954729  0.0246751495 -0.0787837392  0.0354981690
## average_montly_hours     0.0004808969  0.0009539412 -0.0026320349  0.0004733579
```

```
## time_spend_company    -0.0848060837 -0.0099543676  0.1549498258  0.0087581736
## Work_accident         -0.0460102923 -0.0189489298  0.0558399823 -0.0013323237
## promotion_last_5years  0.0445870577 -0.0362588815  0.0256496255 -0.0102998212
## departmentaccounting   0.2726562524 -0.3088416037 -0.2063011731 -0.3121011017
## departmenthr           0.2617218904 -0.3031354245 -0.2009286008 -0.3092256344
## departmentIT           0.2530804200 -0.3074977081 -0.2138527537 -0.3122946398
## departmentmanagement   0.2813707698 -0.3140668071 -0.2182680341 -0.3139963764
## departmentmarketing    0.2494198291 -0.3115467723 -0.2017645590 -0.3141845219
## departmentproduct_mng  0.2509601142 -0.3093125896 -0.2205642925 -0.3115943700
## departmentRandD        0.2683870351 -0.3101079561 -0.2420178580 -0.3103580493
## departmentsales        0.2734424464 -0.3121226567 -0.2101094252 -0.3135163229
## departmentsupport      0.2478899702 -0.3036145040 -0.2085637793 -0.3136975560
## departmenttechnical    0.2634198025 -0.3040439467 -0.2140483887 -0.3109601911
## salary.L              -0.0325131154 -0.0215346753  0.0404670046 -0.0074476397
## salary.Q              -0.0215541655 -0.0021442014  0.0160568601 -0.0052026369
##
## Coefficients of ridge terms ('beta'):
##    term 1    term 2    term 3    term 4
## 0.1374878 0.2756484 0.2442952 0.2813322
```

```r
# PREDICTION
yhat.ppr <- predict(fit1.ppr, newdata=TestData)
yhat.ppr <- scale(yhat.ppr,center = min(yhat.ppr),scale = max(yhat.ppr)-min(yhat.ppr))
AUC.PPR <- ci.cvAUC(predictions=yhat.ppr, labels=yobs, folds=1:length(yhat.ppr), confide
```

```
## Warning in if (class(predictions) == "list" | class(labels) == "list") {: the
## condition has length > 1 and only the first element will be used
```

```
## $cvAUC
## [1] 0.9656718
##
## $se
## [1] 0.003099812
##
## $ci
## [1] 0.9595963 0.9717473
##
## $confidence
## [1] 0.95
```

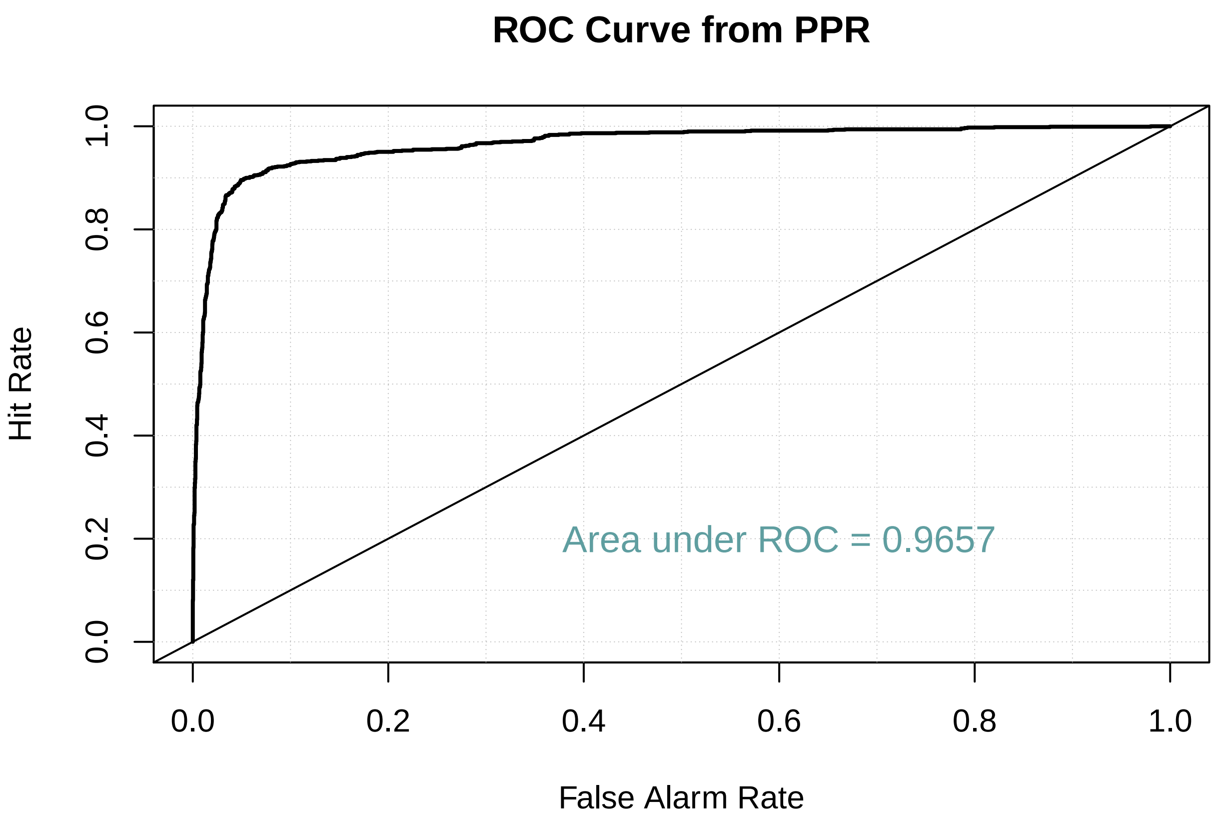
```r
auc.ci <- round(AUC.PPR$ci, digits=4)
library(verification)
mod.ppr <- verify(obs=yobs, pred=yhat.ppr)
```

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

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

```
## Warning in roc.plot.default(c(1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, :
## 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.PPR$cvAUC, digits=4),
    sep=" "), col="cadetblue", cex=1.2)
```



The PPR gives the area under ROC value of 0.9657.

5 Results and Comparison

```
Measure <- c(round(AUC.lasso$cvAUC, digits=3),round(AUC.RF, digits=4),round(AUC.GAM, dig
Measures <- data.frame("Method"= c("LASSO","Random Forest","GAM","MARS","PPR"), "AUC"= M
```

```
##          Method    AUC
## 1         LASSO 0.8220
## 2 Random Forest 0.9939
## 3           GAM 0.9592
## 4          MARS 0.9795
## 5           PPR 0.9657
```

```
knitr::kable(Measures, align = "lc")
```

Method	AUC
LASSO	0.8220
Random Forest	0.9939
GAM	0.9592
MARS	0.9795
PPR	0.9657

Given the above results, among all the five supervised learning approaches, Random forest gave the best results (since it provides the largest AUC) of correctly predicting the probability of employee turnovers in the company. Among all the methods, we see that satisfaction level and number of projects are the top two variables that predict an employees turnover or detention.