Project-6

Chitra Karki[*] University of Texas at El Paso (UTEP)

November 22, 2022

Contents

We consider a human resource data set concerning employee retention from one Kaggle data analytics competition. The data set contains 14,999 observations and 10 variables. The binary target left indicates whether a employee left the company.

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:

[*]cbkarki@miners.utep.edu

hr$salary <- factor(hr$salary, levels=c("low", "medium", "high"), ordered=TRUE)
Change the column name for variable sales to department. Make sure that the target variable left is categorical, i.e., factor in R. Inspect if there is any missing values and, if so, handle them with imputation.

```
setwd("C:/Users/chitr/OneDrive - University of Texas at El Paso/data_science/semesters/sem3-fa
hr = read.csv(file = "HR_comma_sep.csv")

str(hr)
```

```
## '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 ...
##  $ left                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ promotion_last_5years: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ sales                : chr  "sales" "sales" "sales" "sales" ...
##  $ salary               : chr  "low" "medium" "medium" "low" ...
```

```
# sales to department
names(hr)[which(names(hr)=="sales")] = "department"

# salary to ordinal
hr$salary <- factor(hr$salary, levels=c("low", "medium",
"high"), ordered=TRUE)
#hr$salary = as.numeric(hr$salary)

# missing values
sum(is.na(hr))
```

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

```
str(hr)
```

```
## '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 ...
##  $ left                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ 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 1 ...
```

No missing values in the data set hr.

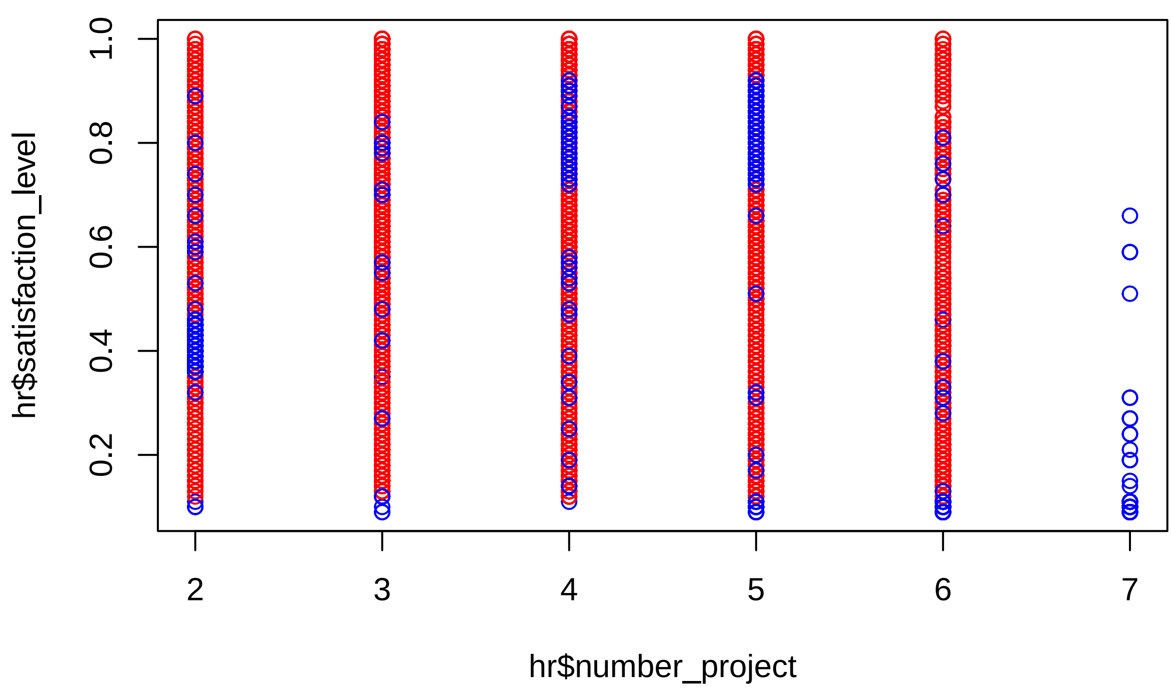
2 Exploratory Data Analysis (EDA)

Explore the data with EDA. If you type the key word 'Human Resources Analytics + Kaggle' On Google, you can find many R/Python examples posted by other experts with different EDA and supervised learning methods. Please study their approach and feel free to reproduce some of the results in this project. Nevertheless, make sure that you understand what you are doing and interpret the results appropriately. In particular,

(a) Make a scatterplot of satisfaction level versus number project and color the points differently according to the target variable left. Interpret the results.

(b) Optionally, you may compute and visualize the correlation matrix among the variables. This is part of the reason that we make sure that salary is ordinal. Since the data contain different types of variables, Pearson correlation may not be a good choice. Besides the above, present at least THREE more interesting findings from your EDA.

```
# scatter plot
par(mfrow=c(1,1))
plot(y=hr$satisfaction_level,x=hr$number_project,col=ifelse(hr$left==0,
                                                           "red","blue"))
```

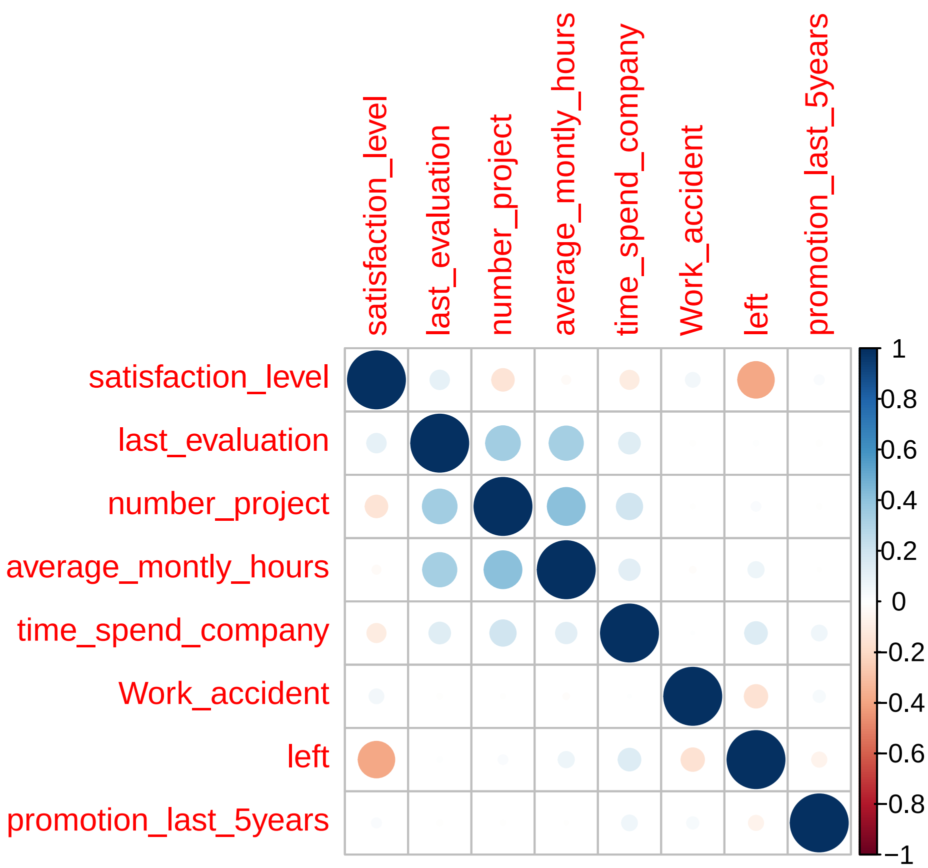


For project number 7, all left the company, it make sense because people will have low satisfaction with heavy amout or work.

```
str(hr)
```

```
## '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 ...
##  $ left                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ 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 1 1 ...
```

```
cor = cor(hr[,-c(9,10)])
suppressPackageStartupMessages(library("corrplot"))
corrplot(cor)
```



left is negatively correlated with satisfaction level. More people tend to leave if satisfacation is less. Also evaluation is high for people we do more projects and will have more average working hours.

3 Data Partitioning

Randomly split the data D into the training set D1 and the test set D2 with a ratio of approximately 2:1 on the sample size. Always use set.seed() in order to have reproducible results. 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.

```
set.seed(123)
D1.index = sample(1:nrow(hr),size = (2/3)*nrow(hr))
D1 = hr[D1.index,]
D2 = hr[-D1.index,]
```

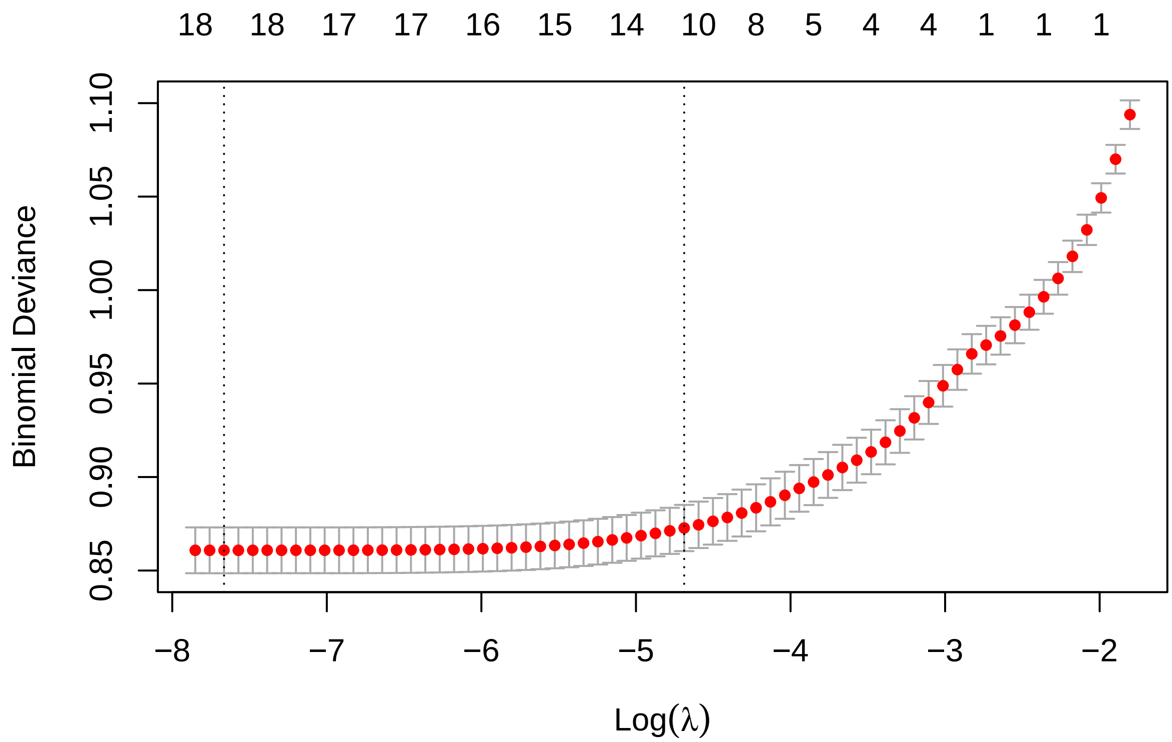
4 Logistic Regression

Fit a regularized logistic regression model as one baseline classifier for comparison. You may use either LASSO or SCAD or any other penalty function of your choice. Explain how you determine the optimal tuning parameter. Remember that logistic regression model is highly interpretable; present your final model and interpret the results.

```
#install.packages("glmnet")
suppressPackageStartupMessages(library(glmnet))
```

```
## Warning: package 'glmnet' was built under R version 4.2.2
```

```
x = model.matrix(left~.,data = D1)
y = D1[,7]
set.seed(123)
lasso.cv = cv.glmnet(x = x,y = y,alpha = 1,family = "binomial")
plot(lasso.cv)
```




```
#min lambda
lambda_min = lasso.cv$lambda.min
#1se lambda
lambda_1se = lasso.cv$lambda.1se
#regression coefficients
coef(lasso.cv,s=lambda_min)
```

```
## 20 x 1 sparse Matrix of class "dgCMatrix"
##                                 s1
## (Intercept)           -0.275868415
## (Intercept)            .
## satisfaction_level    -4.087998993
## last_evaluation        0.684734644
## number_project        -0.310007917
## average_montly_hours   0.004243868
## time_spend_company     0.268414821
## Work_accident         -1.533569699
## promotion_last_5years -1.439107258
## departmenthr           0.227625396
## departmentIT          -0.171936794
## departmentmanagement  -0.375624877
## departmentmarketing   -0.063642754
```

```
## departmentproduct_mng -0.201428210
## departmentRandD       -0.544944238
## departmentsales       -0.025334925
## departmentsupport      0.032444315
## departmenttechnical    0.076813758
## salary.L              -1.249040275
## salary.Q              -0.317075355
```

```
coef(lasso.cv,s=lambda_1se) # lets use this one for the model fitting as this
```

```
## 20 x 1 sparse Matrix of class "dgCMatrix"
##                                s1
## (Intercept)            0.110706508
## (Intercept)            .
## satisfaction_level    -3.591710511
## last_evaluation        0.124191177
## number_project        -0.162086355
## average_montly_hours   0.002284828
## time_spend_company     0.197362604
## Work_accident         -1.171910614
## promotion_last_5years -0.589377619
## departmenthr           0.018763458
## departmentIT           .
## departmentmanagement  -0.096682743
## departmentmarketing    .
## departmentproduct_mng  .
## departmentRandD       -0.200772866
## departmentsales        .
## departmentsupport      .
## departmenttechnical    .
## salary.L              -0.735886040
## salary.Q               .
```

```
# less number of variables.

x_test = model.matrix(left~.,data = D2)

# fitting the moded with lambda_1se
lasso.model = glmnet(x = x,y=y,alpha = 1,family = "binomial",
                     lambda = lambda_1se)


# prediction
lasso_pred = predict(lasso.model,newx = x_test,s=lambda_1se,type = "response")
```

The lasso regression was used. The tuning parameter lambda was obtained via cross validation and with 1.se. The dummy variables were created for the categorical variable like department. If we look at the regression coefficients with lambda minimum and lambda 1se, lambda 1se has move coefficients knocked out to zero.

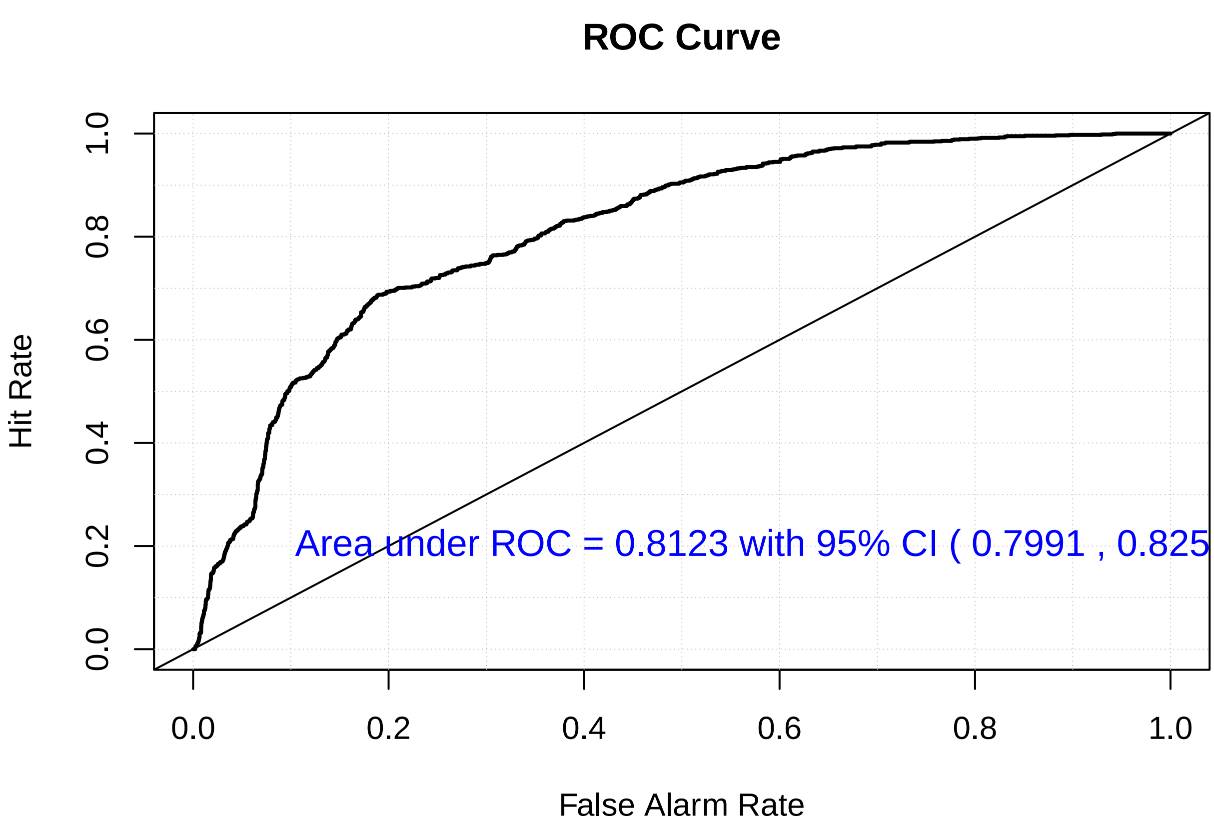
```
suppressPackageStartupMessages(library(cvAUC))
AUC <- ci.cvAUC(predictions=lasso_pred, labels=D2[,7], confidence=0.95)
auc.ci <- round(AUC$ci, digits=4)
suppressPackageStartupMessages(library(verification))
mod.glm <- verify(obs=D2[,7], pred=lasso_pred)
```

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

```
roc.plot(mod.glm, plot.thres = NULL)
```

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



5 RF

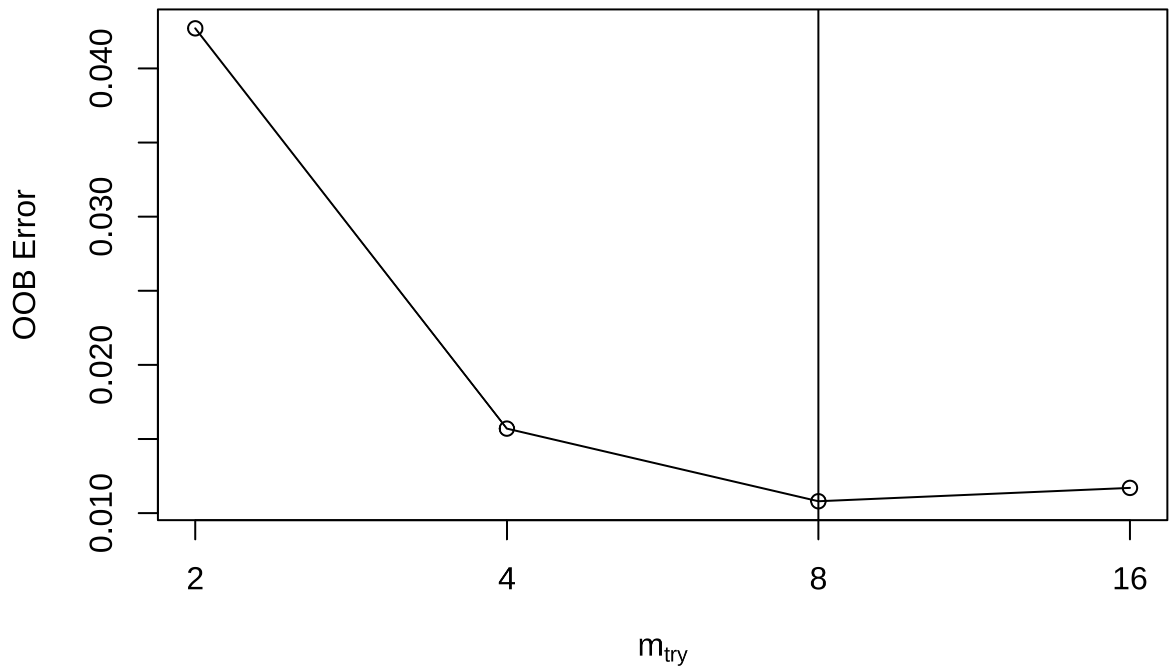
Fit random forests as another baseline for comparison. RF is one top performer. Also, obtain partial dependence plots and variable importance ranking from RF; these results should be interpreted as well.

```
suppressPackageStartupMessages(library("randomForest"))
set.seed(123)
tuneRF(x=x,y=as.factor(y))
```

```
## mtry = 4  OOB error = 1.57%
## Searching left ...
## mtry = 2     OOB error = 4.27%
## -1.719745 0.05
## Searching right ...
## mtry = 8     OOB error = 1.08%
## 0.3121019 0.05
## mtry = 16    OOB error = 1.17%
## -0.08333333 0.05
```

```
##         mtry   OOBError
## 2.OOB      2 0.04270427
## 4.OOB      4 0.01570157
## 8.OOB      8 0.01080108
## 16.OOB    16 0.01170117
```

```
abline(v=8)
```

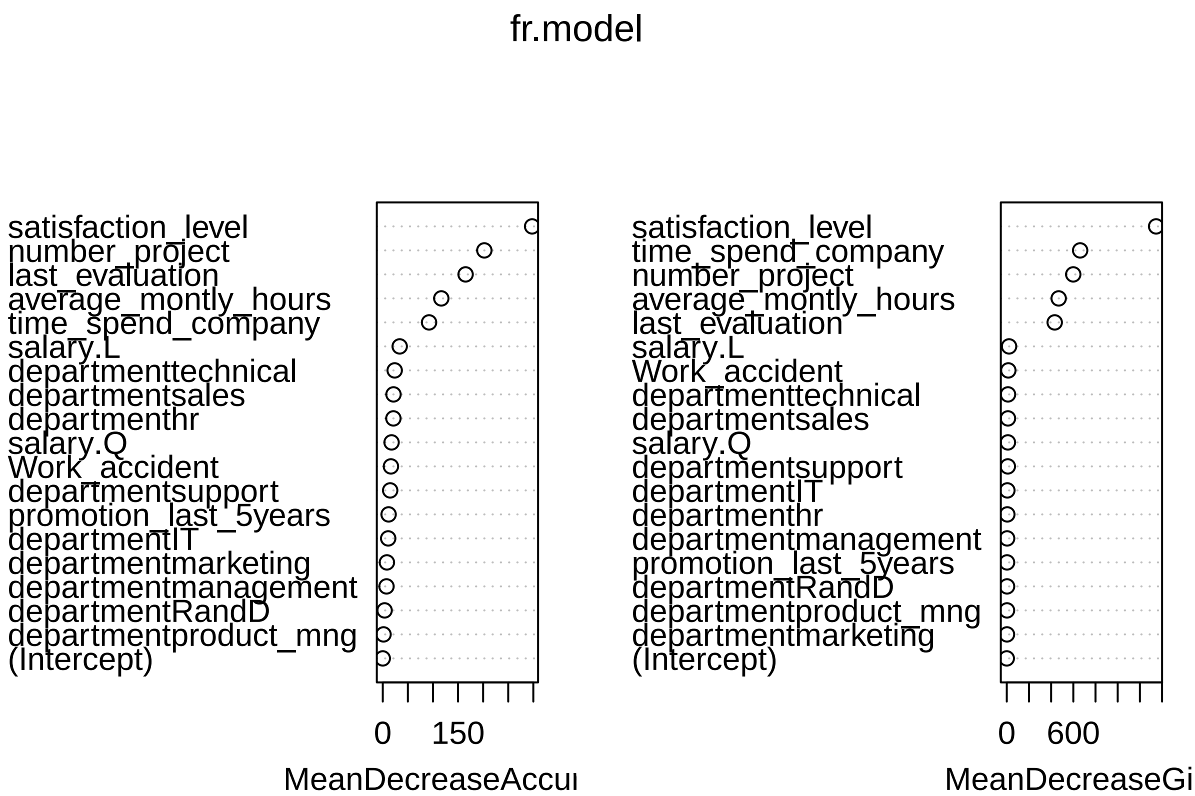



```r
# fitting random forest model with mtry=8
fr.model = randomForest(x=x,y=as.factor(y),importance=T,mtry=8)
# which.min(fr.model$err.rate[,3])
# plot(fr.model)
rf.model.pred = predict(fr.model,newdata = x_test,type = "prob")
importance(fr.model)
```

```
##                                0          1 MeanDecreaseAccuracy
## (Intercept)            0.0000000   0.000000             0.000000
## satisfaction_level    91.2991703 319.827878           297.595349
## last_evaluation       28.8244794 168.173070           165.010782
## number_project        48.3397355 210.280522           202.382487
## average_montly_hours  63.6863271 104.518182           116.627051
## time_spend_company    60.4981994  84.814980            92.260232
## Work_accident          7.4632865  16.680426            16.244915
## promotion_last_5years  4.0639339  11.678535            11.640105
## departmenthr           0.3676654  39.089293            21.161876
## departmentIT           5.3059174  12.636912            10.848377
## departmentmanagement   3.0396140   9.689375             7.459884
## departmentmarketing    1.8329557  11.827591             8.260137
## departmentproduct_mng  0.1389113   3.031081             1.543339
## departmentRandD       -1.8274952  10.539413             3.697032
```

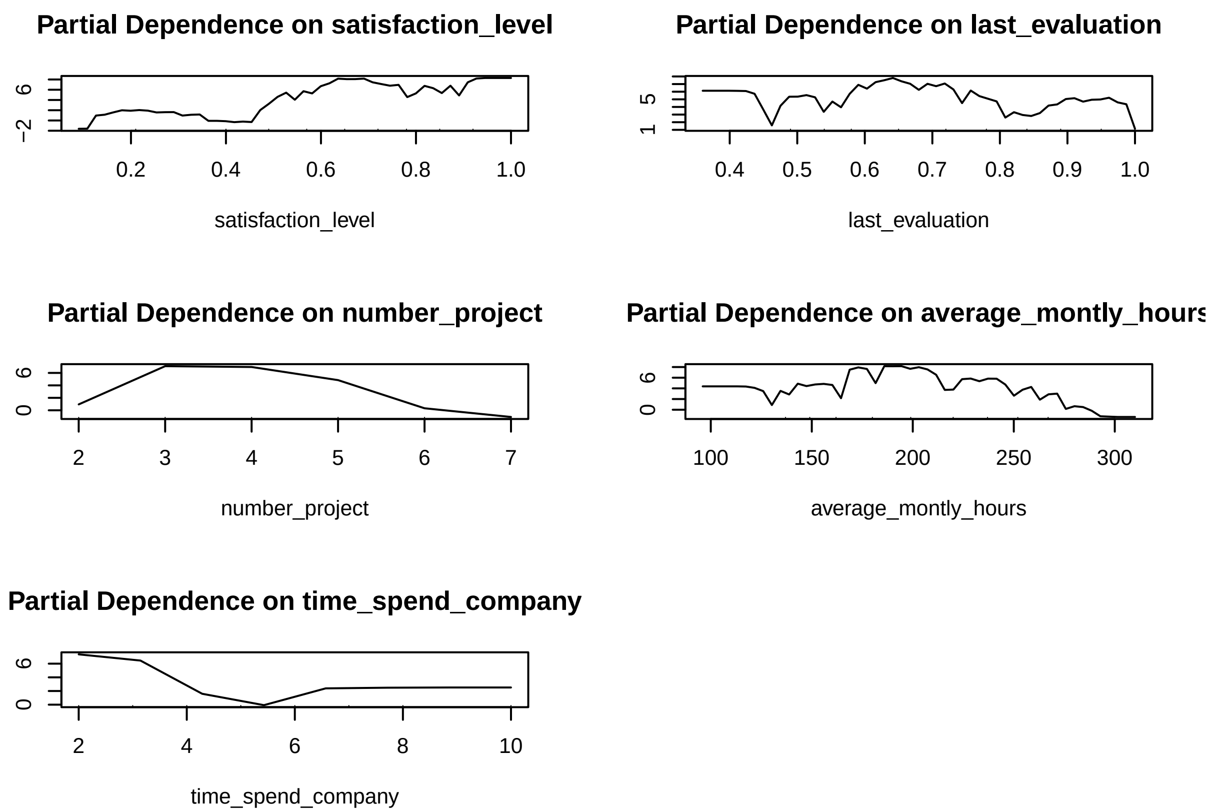
```
## departmentsales        3.0656592  34.259038              21.384931
## departmentsupport      3.0029451  24.536569              14.783145
## departmenttechnical    1.7909788  40.329525              23.776540
## salary.L              16.2754985  33.344749              33.736027
## salary.Q               5.9225409  21.286998              17.532073
##                       MeanDecreaseGini
## (Intercept)                   0.000000
## satisfaction_level         1346.351239
## last_evaluation             432.107556
## number_project              599.167327
## average_montly_hours        467.933504
## time_spend_company          661.441969
## Work_accident                14.291361
## promotion_last_5years         2.633916
## departmenthr                  5.097804
## departmentIT                  5.514650
## departmentmanagement          3.480322
## departmentmarketing           2.052349
## departmentproduct_mng         2.279616
## departmentRandD               2.578635
## departmentsales              10.949253
## departmentsupport             9.723005
## departmenttechnical          11.305711
## salary.L                     20.764849
## salary.Q                     10.427139
```

```
varImpPlot(fr.model)
```



From tuning mtry we observed that mtry = 8 produces minimum out of bag error.Form the graphs, its observed that the continuous variables like satisfication level, last _evaluation, number_project,average_monthly_hrs,time_spend_company are more important in predicting the response variable.

```
par(mfrow=c(3,2))
partialPlot(fr.model,x,satisfaction_level)
partialPlot(fr.model,x,last_evaluation)
partialPlot(fr.model,x,number_project)
partialPlot(fr.model,x,average_montly_hours)
partialPlot(fr.model,x,time_spend_company)
```

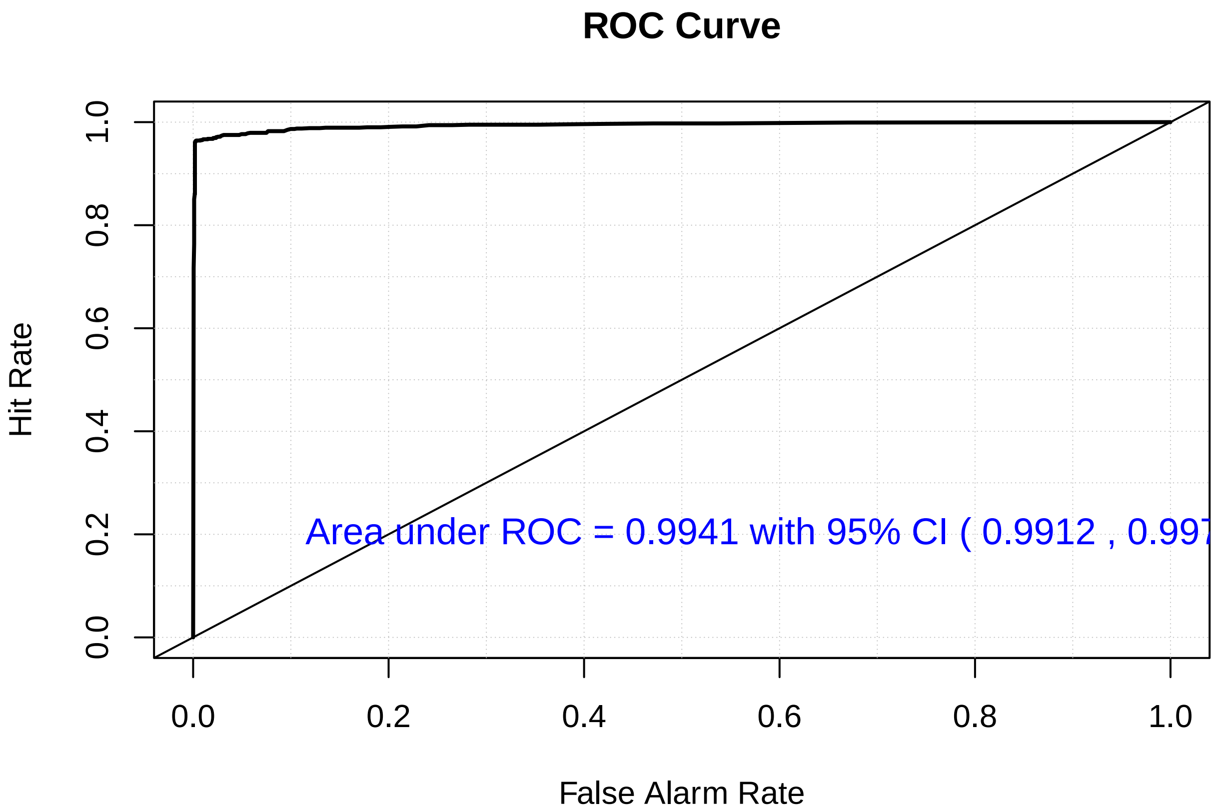


The partial plots for continuous variables are plotted above. If the curve goes up for increasing variables then there is high probability of left and ,if curve goes down, the probability of left is low.

```
AUC <- ci.cvAUC(predictions=rf.model.pred[,2], labels=D2[,7], confidence=0.95)
auc.ci <- round(AUC$ci, digits=4)
mod.rf <- verify(obs=D2[,7], pred=rf.model.pred[,2])
```

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

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



6 GAM

Fit a generalized additive model. Explain how you determine the smoothing param- eters and variable/model selection involved in fitting GAM. Present your final model. Plots the (nonlinear) functional forms for continuous predictors and comment on the adequacy of the (linear) logistic regression in Part 4.

```
suppressPackageStartupMessages(library(gam))
```

```
## Warning: package 'gam' was built under R version 4.2.2
```

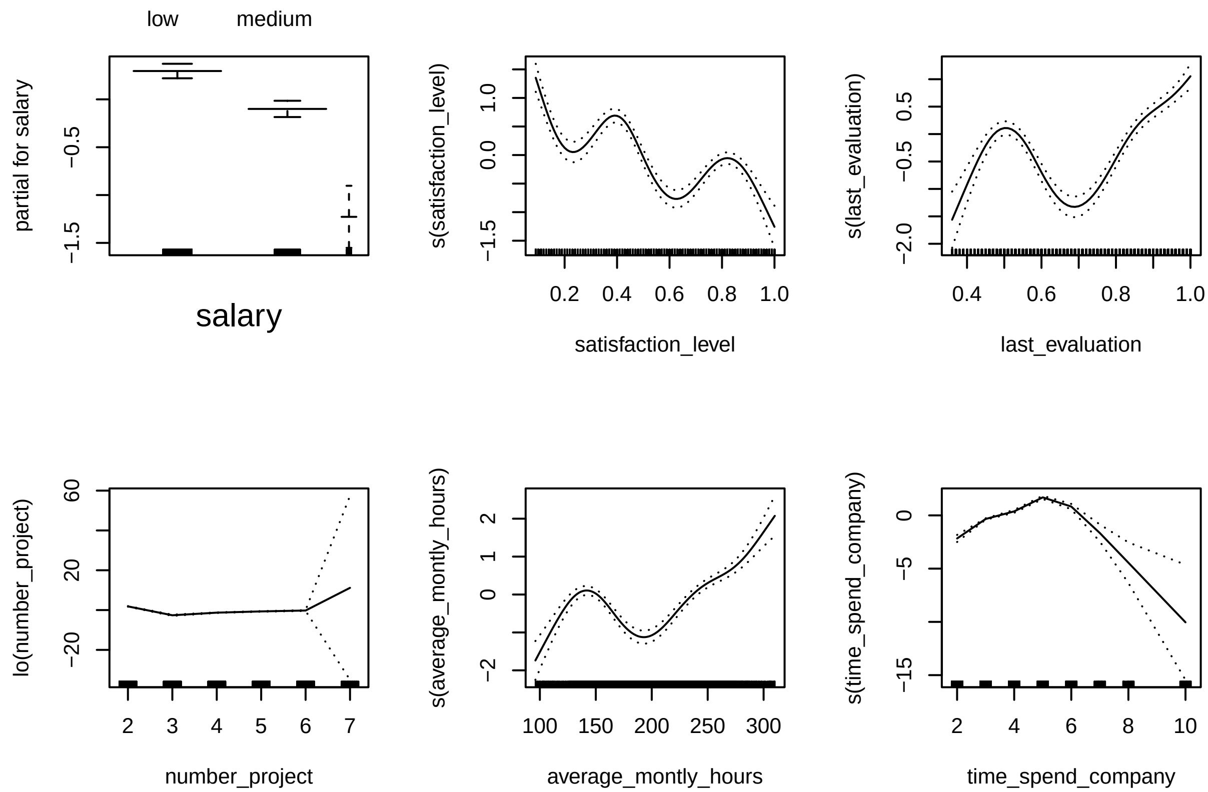
```
gam.model <- gam(left ~ satisfaction_level + number_project +
                   time_spend_company +
department + last_evaluation + average_montly_hours + Work_accident +
  promotion_last_5years + salary , family = binomial,data=D1, trace=T, control = gam.control(e
```

```
summary(gam.model.step)
```

```
##
## Call: gam(formula = left ~ salary + s(satisfaction_level) + s(last_evaluation) +
```

```
##     lo(number_project) + s(average_montly_hours) + s(time_spend_company),
##     family = binomial, data = D1, control = gam.control(epsilon = 1e-04,
##         bf.epsilon = 1e-04, maxit = 50, bf.maxit = 50), trace = FALSE)
## Deviance Residuals:
##       Min        1Q    Median        3Q       Max
## -3.048461 -0.315705 -0.128497 -0.004693  3.628274
##
## (Dispersion Parameter for binomial family taken to be 1)
##
##     Null Deviance: 10946.72 on 9998 degrees of freedom
## Residual Deviance: 4086.844 on 9975.001 degrees of freedom
## AIC: 4134.841
##
## Number of Local Scoring Iterations: 1
##
## Anova for Parametric Effects
##                             Df  Sum Sq Mean Sq F value    Pr(>F)
## salary                       2    73.1   36.53  34.277 1.461e-15 ***
## s(satisfaction_level)        1    29.6   29.60  27.775 1.391e-07 ***
## s(last_evaluation)           1    47.3   47.28  44.359 2.877e-11 ***
## lo(number_project)           1   128.5  128.46 120.523 < 2.2e-16 ***
## s(average_montly_hours)      1    67.1   67.10  62.958 2.339e-15 ***
## s(time_spend_company)        1   351.1  351.14 329.440 < 2.2e-16 ***
## Residuals                 9975 10632.0    1.07
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                           Npar Df Npar Chisq    P(Chi)
## (Intercept)
## salary
## s(satisfaction_level)        3     505.23 < 2.2e-16 ***
## s(last_evaluation)           3     362.87 < 2.2e-16 ***
## lo(number_project)           4     715.94 < 2.2e-16 ***
## s(average_montly_hours)      3     351.28 < 2.2e-16 ***
## s(time_spend_company)        3     287.74 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
par(mfrow=c(2,3))
plot(gam.model.step,se=T)
```



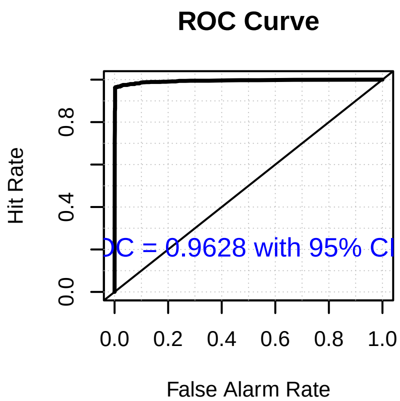

```
gm.model.pred <- predict(gam.model.step, newdata=D2, type="response",
                         se.fit=FALSE)

AUC <- ci.cvAUC(predictions=gm.model.pred, labels=D2[,7], confidence=0.95); AUC
```

```
## $cvAUC
## [1] 0.9627862
##
## $se
## [1] 0.003192223
##
## $ci
## [1] 0.9565295 0.9690428
##
## $confidence
## [1] 0.95
```

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

mod.rf <- verify(obs=D2[,7], pred=rf.model.pred[,2])
```

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

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



Stepwise selection was used to obtain best fitting model for GAM. local and smoothing splines were used for smooting. From summary table its observed that satisfication_level, last_evaluation, average_monthly_hours,time_spend_company have smoothing splines and number_project has localy weighted smpothing as optimum

7 MARS

Train a multivariate adaptive regression splines model. Present the final model if possible. Obtain variable importance ranking and partial dependence plots (for continuous predictors.

```
suppressPackageStartupMessages(library(earth))
```

```
## Warning: package 'earth' was built under R version 4.2.2
```

```
## Warning: package 'plotmo' was built under R version 4.2.2
```

```
## Warning: package 'TeachingDemos' was built under R version 4.2.2
```

```
suppressPackageStartupMessages(library("tidyverse"))

# FITTING MARS
mars.model <- earth(factor(left) ~ .,  data = D1, degree=1,
    glm=list(family=binomial(link = "logit")),
    pmethod="cv", nfold=3)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

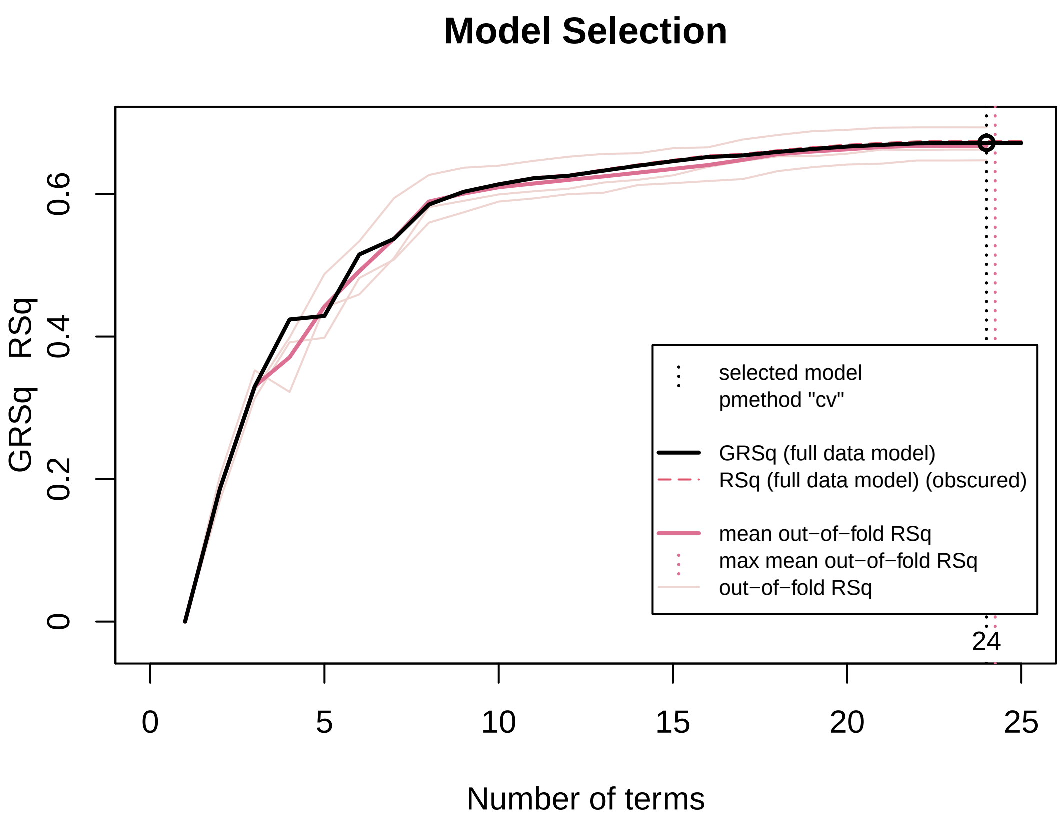
```
print(mars.model)
```

```
## GLM (family binomial, link logit):
##  nulldev   df       dev   df   devratio     AIC iters converged
##  10946.7 9998   3002.25 9975      0.726    3050    18         1
##
## Earth selected 24 of 25 terms, and 6 of 18 predictors (pmethod="cv")
## Termination condition: Reached nk 37
## Importance: number_project, satisfaction_level, time_spend_company, ...
## Number of terms at each degree of interaction: 1 23 (additive model)
## Earth GRSq 0.6717126  RSq 0.6747265  mean.oof.RSq 0.667659 (sd 0.0236)
##
## pmethod="backward" would have selected the same model:
##     24 terms 6 preds,  GRSq 0.6717126  RSq 0.6747265  mean.oof.RSq 0.667659
```

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

```
##                                           1
## (Intercept)                      2.91303346
## h(number_project-3)              0.03211808
## h(3-number_project)              0.30360936
## h(time_spend_company-5)         -0.51806939
## h(5-time_spend_company)         -0.03804961
## h(0.24-satisfaction_level)       1.52422582
## h(satisfaction_level-0.38)     -13.25593391
## h(satisfaction_level-0.51)       7.55419753
## h(last_evaluation-0.99)         24.78733405
## h(0.99-last_evaluation)         -4.15125997
```

```
# MODEL SELECTION
par(mfrow=c(1, 1), mar=rep(4,4))
plot(mars.model, which = 1)
```



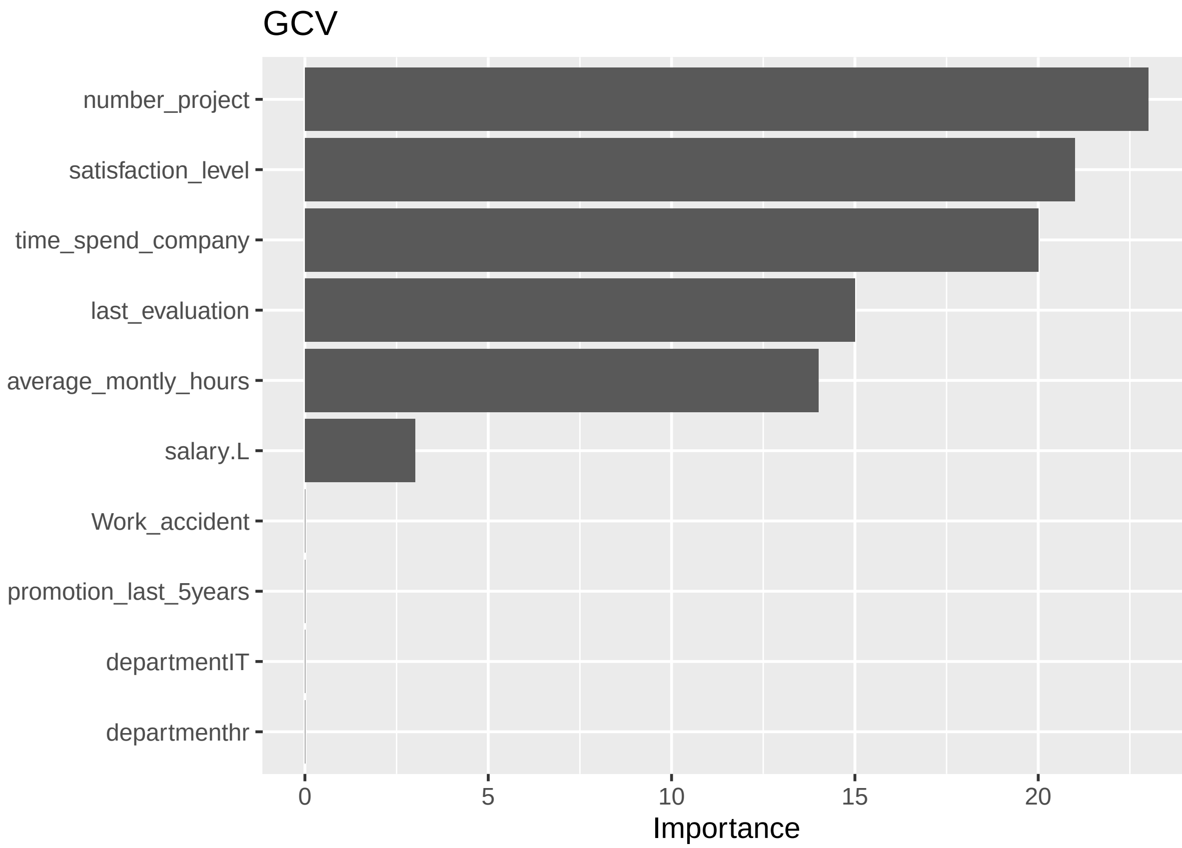

```
# VARIABLE IMPORTANCE PLOT
library("vip")
```

```
## Warning: package 'vip' was built under R version 4.2.2
```

```
##
## Attaching package: 'vip'
```

```
## The following object is masked from 'package:utils':
##
##     vi
```

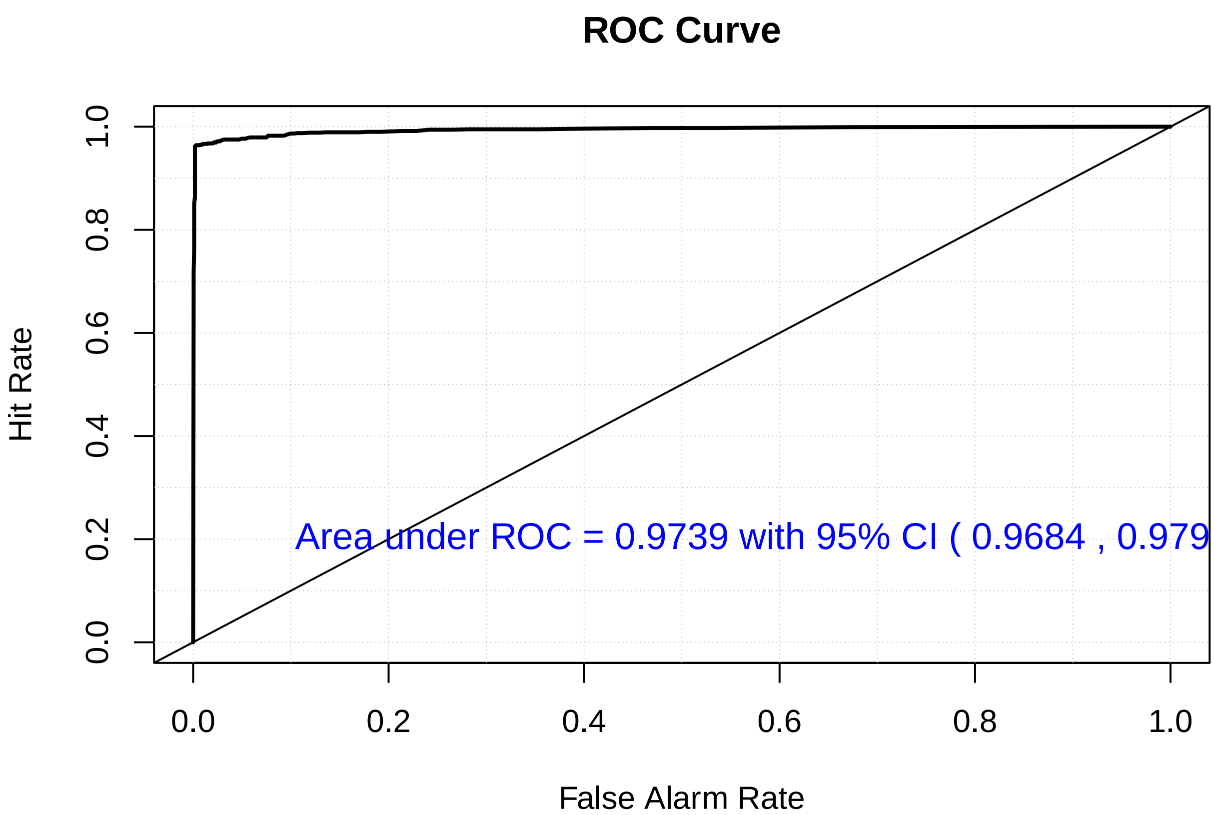
```
vip(mars.model) + ggtitle("GCV")
```




```
# PREDICTION
mars.model.pred <- predict(mars.model, newdata=D2, type="response")

AUC <- ci.cvAUC(predictions=mars.model.pred, labels=D2[,7], confidence=0.95);
auc.ci <- round(AUC$ci, digits=4)

mod.rf <- verify(obs=D2[,7], pred=rf.model.pred[,2])
```

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

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



Again the continuous variables are seems to have more importance in predicting the response variable as mention above in RF modeling.

8 PPR

Train a project pursuit regression model. This model is hard to interpret. Focus on its predictive performance only.

```
ppr.model0 <- ppr(left ~ ., data = D1,
                  nterms = 2, max.terms = 5,
                  sm.method = "supsmu", bass=0, spen=0)
summary(ppr.model0);
```

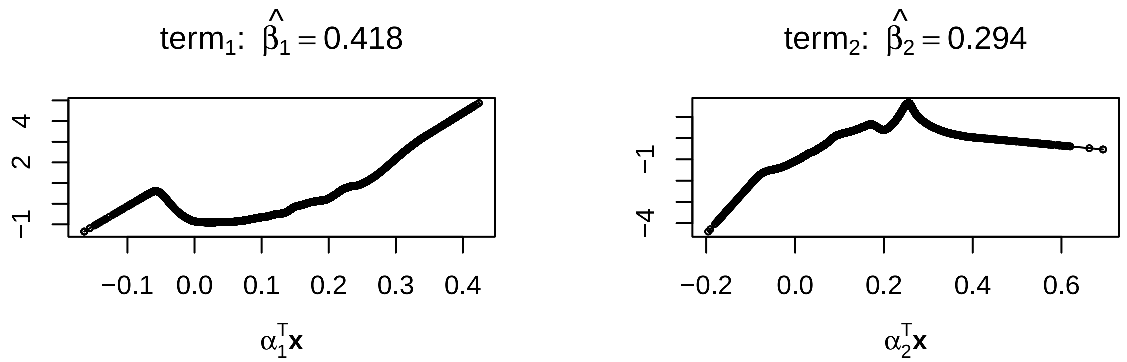
```
## Call:
## ppr(formula = left ~ ., data = D1, nterms = 2, max.terms = 5,
##     sm.method = "supsmu", bass = 0, spen = 0)
##
## Goodness of fit:
##  2 terms  3 terms  4 terms  5 terms
## 608.9823 449.9985 411.7788 386.2505
##
```

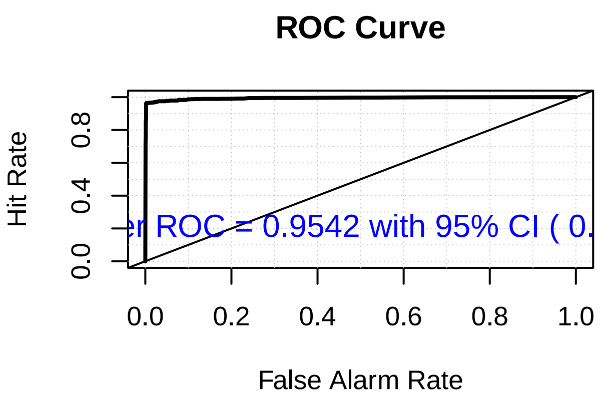
```
## Projection direction vectors ('alpha'):
##                       term 1        term 2
## satisfaction_level    -0.0204786378  0.0622208997
## last_evaluation        0.1631150644 -0.1089804947
## number_project         0.0601324293 -0.0693109992
## average_montly_hours   0.0006602845 -0.0004679684
## time_spend_company    -0.0121570925  0.0591716118
## Work_accident         -0.0048577149 -0.0053560984
## promotion_last_5years -0.0167695287  0.0025279375
## departmentaccounting  -0.3089246081  0.3111976660
## departmenthr          -0.3108493721  0.3160725093
## departmentIT          -0.3096466399  0.3110238502
## departmentmanagement  -0.3122557567  0.3133533671
## departmentmarketing   -0.3133167994  0.3129470082
## departmentproduct_mng -0.3109508605  0.3082001608
## departmentRandD       -0.3127309548  0.3102706446
## departmentsales       -0.3116481212  0.3125320643
## departmentsupport     -0.3108336686  0.3144674614
## departmenttechnical   -0.3114571736  0.3137751389
## salary.L              -0.0068014590 -0.0047169008
## salary.Q              -0.0024991430 -0.0022425432
##
## Coefficients of ridge terms ('beta'):
##    term 1    term 2
## 0.4180385 0.2943300
```

```r
par(mfrow=c(2, 2))
plot(ppr.model0)

ppr.model0.pred <- predict(ppr.model0, newdata = D2)

AUC <- ci.cvAUC(predictions=ppr.model0.pred, labels=D2[,7], confidence=0.95)
auc.ci <- round(AUC$ci, digits=4)

mod.rf <- verify(obs=D2[,7], pred=rf.model.pred[,2])
```

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

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





9 Summary

Summarize the results and compare the above five supervised learning approaches in terms of their pros and cons within this application context of employee retention.

```
data.frame(Models=c("Lasso Regression","Random Forest","GAM","MARS","PPR"),
           AUC = c(0.8110163 , 0.9938683, 0.9624532, 0.9727,0.9428))
```

```
##             Models       AUC
## 1 Lasso Regression 0.8110163
## 2    Random Forest 0.9938683
## 3              GAM 0.9624532
## 4             MARS 0.9727000
## 5              PPR 0.9428000
```

For the comparision of AUC, we observed that the lasso regression has the least AUC and Random Forest has the greatest. So, the winner based on the AUC criteria is Random Forest.