PROJECT06

Isaiah Thompson Ocansey

2022-11-22

DATA PREPARATION

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

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

The data set has 14999 observations and 10 variables

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

The above output changed the categorical variable salary in the data set to ordinal

```
summary(hr)
```

```
##  satisfaction_level last_evaluation  number_project  average_montly_hours
##  Min.   :0.0900     Min.   :0.3600   Min.   :2.000   Min.   : 96.0
##  1st Qu.:0.4400     1st Qu.:0.5600   1st Qu.:3.000   1st Qu.:156.0
##  Median :0.6400     Median :0.7200   Median :4.000   Median :200.0
##  Mean   :0.6128     Mean   :0.7161   Mean   :3.803   Mean   :201.1
```

```
##  3rd Qu.:0.8200     3rd Qu.:0.8700   3rd Qu.:5.000   3rd Qu.:245.0
##  Max.   :1.0000     Max.   :1.0000   Max.   :7.000   Max.   :310.0
##  time_spend_company Work_accident         left          promotion_last_5years
##  Min.   : 2.000     Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.: 3.000     1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median : 3.000     Median :0.0000   Median :0.0000   Median :0.00000
##  Mean   : 3.498     Mean   :0.1446   Mean   :0.2381   Mean   :0.02127
##  3rd Qu.: 4.000     3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.00000
##  Max.   :10.000     Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
##   department           salary
##  Length:14999       low   :7316
##  Class :character   medium:6446
##  Mode  :character   high  :1237
##
##
##
```

From the output above, it can be seen that among all the predictors, 2 of the variables are continuous; 5 are categorical and the remaining 3 variables are integers.

```
sum(is.na(hr))
```

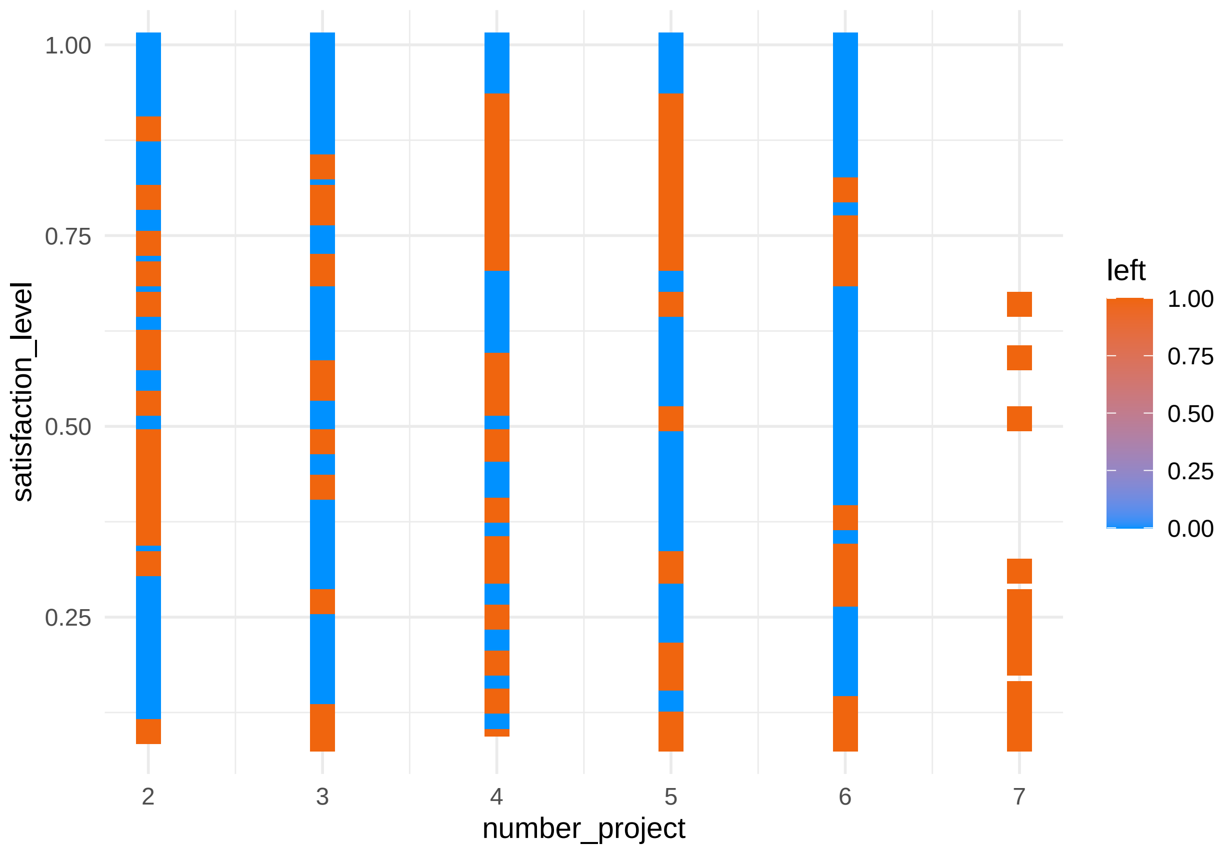
```
## [1] 0
```

From the above, it can be seen that there are no missing values

EXPOLRATORY DATA ANALYSIS

(2a) Make a scatterplot of satisfaction level versus number project and color the points differently according to the target variable left. Interpret the results.

```
library(ggplot2)
ggplot(hr, aes( number_project,satisfaction_level, color = left)) +
  geom_point(shape = 15, size =4 , show.legend = TRUE) +
  theme_minimal() +
  scale_color_gradient(low = "#0091ff", high = "#f0650e")
```



The above plot shows the scatter plot of satisfaction level versus number project.It can been observed that there is a high chance of people leaving on the project number 7

***DATA PARTITION*

(3) 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.

```
set.seed(123)
sample <- sample(nrow(hr), (2/3)*nrow(hr), replace = FALSE)
# training set
D1 <- hr[sample, ]
#test set
D2 <- hr[-sample, ]
dim(D1); dim(D2)
```

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

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

The data set is split into training set and testing set in the ratio 2:1. After the split the training set has 9999 observations and 10 variables whiles the testing set has 5000 observations and 10 variables

LOGISTIC REGRESSION

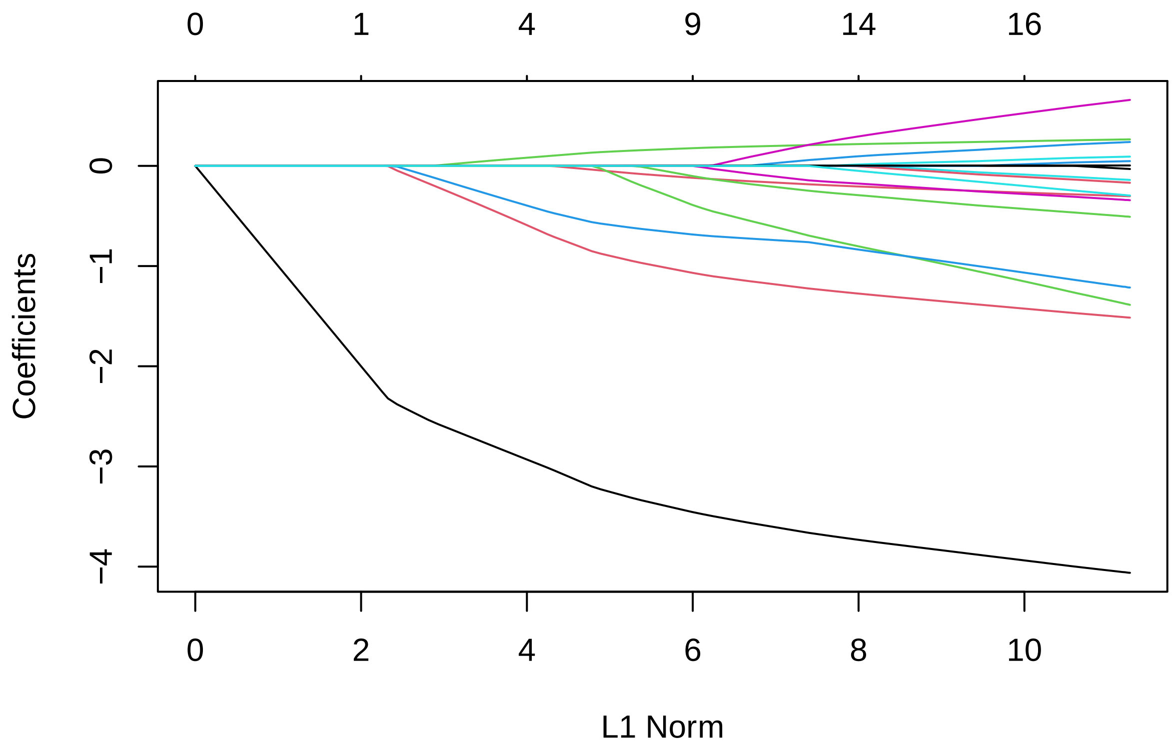
(4)Fit a regularized logistic regression model as one baseline classifier for comparison. You may use either LASSO or SCAD or any other penalty function of your choice. Explain how you determine the optimal tuning parameter. Remember that logistic regression model is highly interpretable present your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

```
library(glmnet)
```

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

```
## Loaded glmnet 4.1-4
```

```
Base_Model <- model.matrix(object=~ satisfaction_level + number_project + time_spend_company +
factor(department) + last_evaluation + average_montly_hours + Work_accident + promotion_last_5years + fa
y <- D1$left
fit.lasso <- glmnet(x=Base_Model, y=y, family="binomial", alpha=1,
lambda.min = 1e-4, nlambda = 200, standardize=T, thresh = 1e-07,
maxit=2000)
plot(fit.lasso)
```

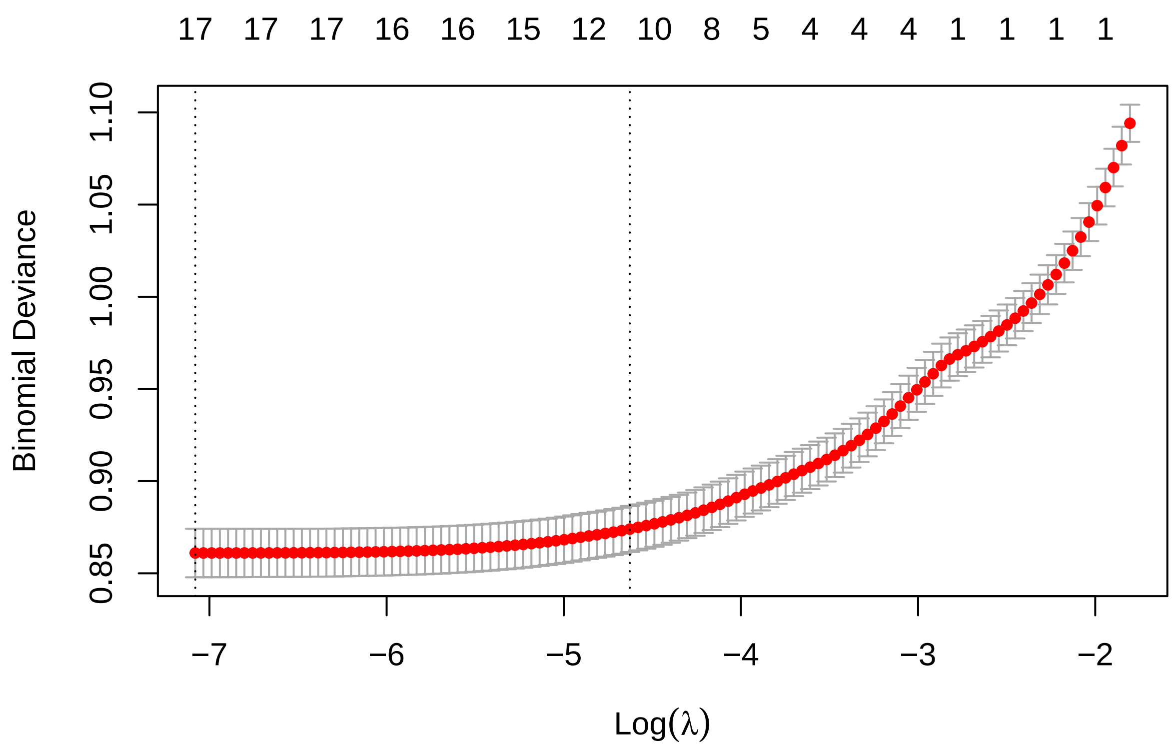



```
mod_cv <- cv.glmnet(x=Base_Model, y=y, family="binomial", alpha = 1,
lambda.min = 1e-4, nlambda = 200, standardize = T, thresh = 1e-07,
maxit=3000)
mod_cv
```

```
##
## Call:  cv.glmnet(x = Base_Model, y = y, family = "binomial", alpha = 1,      lambda.min = 1e-04, nlam
```

```
##
## Measure: Binomial Deviance
##
##       Lambda Index Measure      SE Nonzero
## min 0.000842   115   0.861 0.01316      17
## 1se 0.009782    62   0.874 0.01247      11
```

```r
plot(mod_cv)
```



From the graph of the logistic model with LASSO penalty above, two models; one with 17 variables and the other with 10 variables are significant however we chose the model with 10 variables due to the law of parsimony

```r
best_lamda<-mod_cv$lambda.min #minimum error lamda
best_lamda
```

```
## [1] 0.0008416
```

The best lamda that produces the minimum error is 0.0008416

```r
b.fit.lasso <- glmnet(x=Base_Model, y=y, family="binomial", alpha = 1,
lambda= best_lamda, standardize = T, thresh = 1e-07, maxit=1000)
b.fit.lasso$beta
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
```

```
##                                        s0
## (Intercept)                  .
## satisfaction_level          -4.063195073
## number_project              -0.302844968
## time_spend_company           0.264806607
## factor(department)hr         0.238266736
## factor(department)IT        -0.140516457
## factor(department)management -0.342319595
## factor(department)marketing  -0.031259439
## factor(department)product_mng -0.168904983
## factor(department)RandD      -0.508170425
## factor(department)sales       .
## factor(department)support    0.047693609
## factor(department)technical  0.092480080
## last_evaluation              0.658919285
## average_montly_hours         0.004149665
## Work_accident               -1.515367144
## promotion_last_5years       -1.387768372
## factor(salary).L            -1.217022292
## factor(salary).Q            -0.297085885
```

We can observe that using the best lamda that produces the minimum error, 10 variables are selected

```
fit.pen.lasso <- glm(factor(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.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 = D1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2451  -0.6634  -0.4023  -0.1223   3.0926
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -0.2621245  0.1889728  -1.387 0.165411
## satisfaction_level    -4.1200803  0.1197800 -34.397  < 2e-16 ***
## number_project        -0.3192202  0.0259128 -12.319  < 2e-16 ***
## time_spend_company     0.2731001  0.0192109  14.216  < 2e-16 ***
## departmenthr           0.1957121  0.1605847   1.219 0.222940
## departmentIT          -0.2301334  0.1477726  -1.557 0.119388
## departmentmanagement  -0.4357787  0.1903441  -2.289 0.022055 *
## departmentmarketing   -0.1228640  0.1612418  -0.762 0.446068
## departmentproduct_mng -0.2607839  0.1591776  -1.638 0.101355
## departmentRandD       -0.6097666  0.1791081  -3.404 0.000663 ***
## departmentsales       -0.0758714  0.1242306  -0.611 0.541378
## departmentsupport     -0.0040647  0.1329174  -0.031 0.975604
```

```
## departmenttechnical    0.0401815  0.1296296   0.310 0.756582
## last_evaluation        0.7181540  0.1821124   3.943 8.03e-05 ***
## average_montly_hours   0.0043630  0.0006276   6.952 3.61e-12 ***
## Work_accident         -1.5570035  0.1100485 -14.148  < 2e-16 ***
## promotion_last_5years -1.5072990  0.3234655  -4.660 3.16e-06 ***
## salary.L              -1.2927700  0.1089529 -11.865  < 2e-16 ***
## salary.Q              -0.3439180  0.0713233  -4.822 1.42e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 10946.7  on 9998  degrees of freedom
## Residual deviance:  8565.2  on 9980  degrees of freedom
## AIC: 8603.2
##
## Number of Fisher Scoring iterations: 5
```

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

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

```
##                              2.5 %       97.5 %
## (Intercept)           -0.633697706  0.107243015
## satisfaction_level    -4.356514976 -3.886924834
## number_project        -0.370233102 -0.268642448
## time_spend_company     0.235455406  0.310784005
## departmenthr          -0.119064154  0.510696668
## departmentIT          -0.519393711  0.060111607
## departmentmanagement  -0.812491466 -0.065664300
## departmentmarketing   -0.439398726  0.192968396
## departmentproduct_mng -0.573280401  0.050998229
## departmentRandD       -0.963701212 -0.261062394
## departmentsales       -0.317659249  0.169549929
## departmentsupport     -0.263255822  0.257992981
## departmenttechnical   -0.212390444  0.295971343
## last_evaluation        0.361669970  1.075645336
## average_montly_hours   0.003135321  0.005595921
## Work_accident         -1.777399669 -1.345663872
## promotion_last_5years -2.190703648 -0.912636265
## salary.L              -1.512956603 -1.085201567
## salary.Q              -0.486586196 -0.206645166
```

The above output shows the 95% confidence interval for the coefficients

```
exp(cbind(OR = coef(fit.pen.lasso), confint(fit.pen.lasso))) ##obtaining the odds ratio and the conf in
```

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

```
##                               OR     2.5 %     97.5 %
## (Intercept)           0.76941524 0.5306261 1.11320475
## satisfaction_level    0.01624321 0.0128230 0.02050832
```

```
## number_project        0.72671551 0.6905733 0.76441653
## time_spend_company    1.31403172 1.2654849 1.36449446
## departmenthr          1.21617673 0.8877508 1.66645175
## departmentIT          0.79442765 0.5948811 1.06195506
## departmentmanagement  0.64676082 0.4437511 0.93644518
## departmentmarketing   0.88438392 0.6444238 1.21284446
## departmentproduct_mng 0.77044739 0.5636733 1.05232103
## departmentRandD       0.54347772 0.3814783 0.77023286
## departmentsales       0.92693542 0.7278508 1.18477150
## departmentsupport     0.99594354 0.7685453 1.29432973
## departmenttechnical   1.04099974 0.8086489 1.34443163
## last_evaluation       2.05064422 1.4357250 2.93188434
## average_montly_hours  1.00437257 1.0031402 1.00561161
## Work_accident         0.21076668 0.1690772 0.26036680
## promotion_last_5years 0.22150746 0.1118380 0.40146446
## salary.L              0.27450933 0.2202578 0.33783368
## salary.Q              0.70898703 0.6147214 0.81330819
```

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

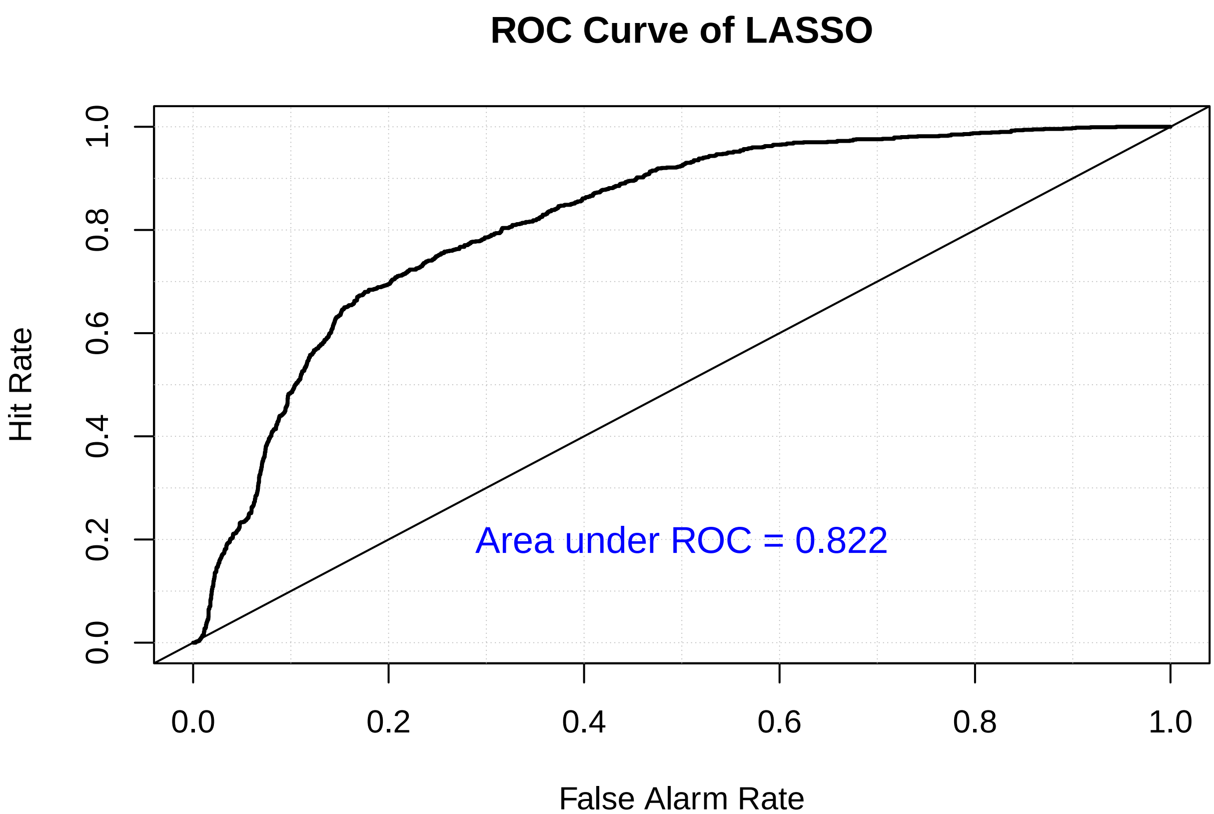
```
library(cvAUC)
library(verification)
n <- NROW(D2)
yobs <- D2$left
yhat.lasso <- predict(fit.pen.lasso, newdata=D2, type="response")
AUC.lasso <- ci.cvAUC(predictions=yhat.lasso, labels=yobs, folds=1:n, confidence=0.95)
AUC.lasso
```

```
## $cvAUC
## [1] 0.8217438
##
## $se
## [1] 0.006486044
##
## $ci
## [1] 0.8090314 0.8344562
##
## $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 of LASSO")
text(x=0.5, y=0.2, paste("Area under ROC =", round(AUC.lasso$cvAUC, digits=3),
sep=" "), col="blue", cex=1.2)
```



The area under the curve of the logistic model with LASSO penalty is 82.2%

```
**RANDOM FOREST**
```

(5) 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. • One common error in previous classes is that many students fit random forests as a regression problem, instead of classification; same for the MARS model below. Please try to avoid this error.

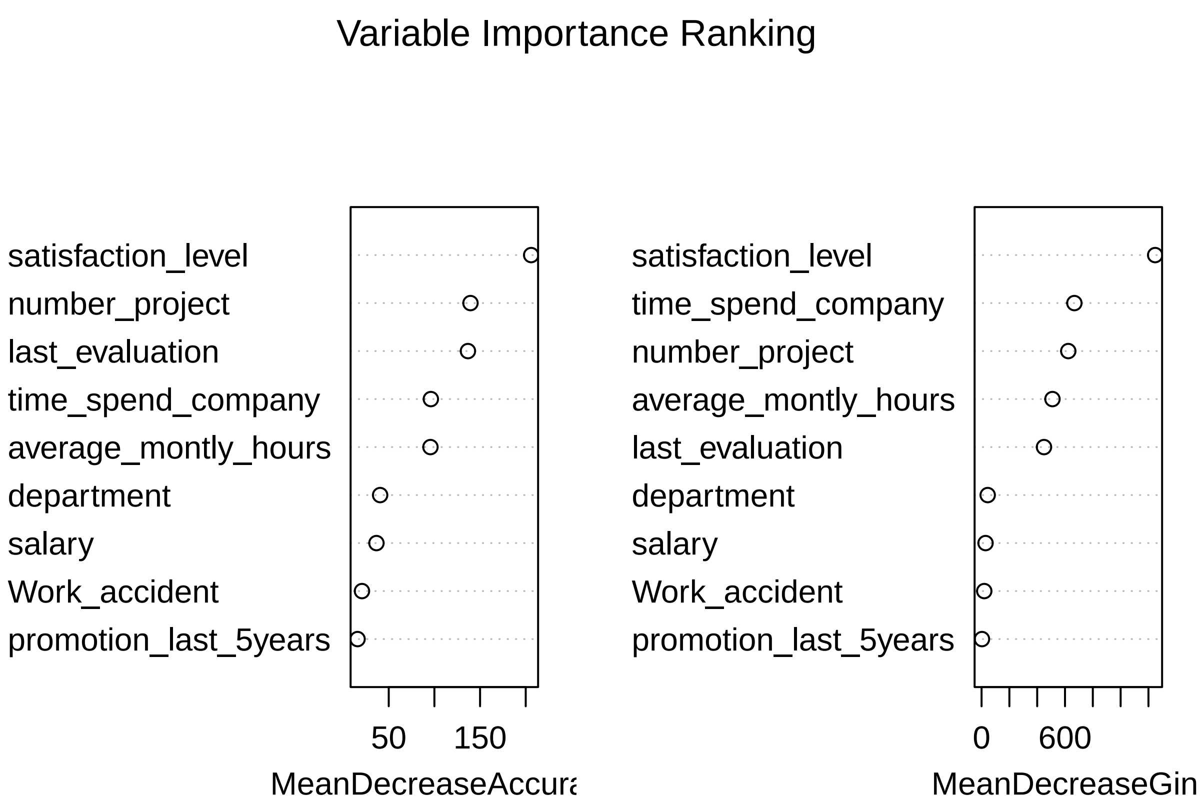
```
library(randomForest)
fit.rf <- randomForest(factor(left) ~., data=D1,importance=TRUE, proximity=TRUE, ntree=500)
fit.rf
```

```
##
## Call:
##  randomForest(formula = factor(left) ~ ., data = D1, importance = TRUE,      proximity = TRUE, ntree
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 0.95%
## Confusion matrix:
##      0    1 class.error
## 0 7617   14 0.001834622
## 1   81 2287 0.034206081
```

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

```
##                           0      1 MeanDecreaseAccuracy MeanDecreaseGini
## satisfaction_level    67.92 235.13               205.93          1248.20
## last_evaluation       27.34 135.44               136.73           448.86
## number_project        46.74 134.23               139.49           623.55
## average_montly_hours  53.19  87.69                95.66           509.71
## time_spend_company    58.82  88.41                96.11           667.32
## Work_accident          8.98  20.43                20.70            19.43
## promotion_last_5years  8.44  14.16                15.86             3.04
## department            10.87  57.00                40.57            44.21
## salary                14.99  39.23                36.53            28.74
```

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

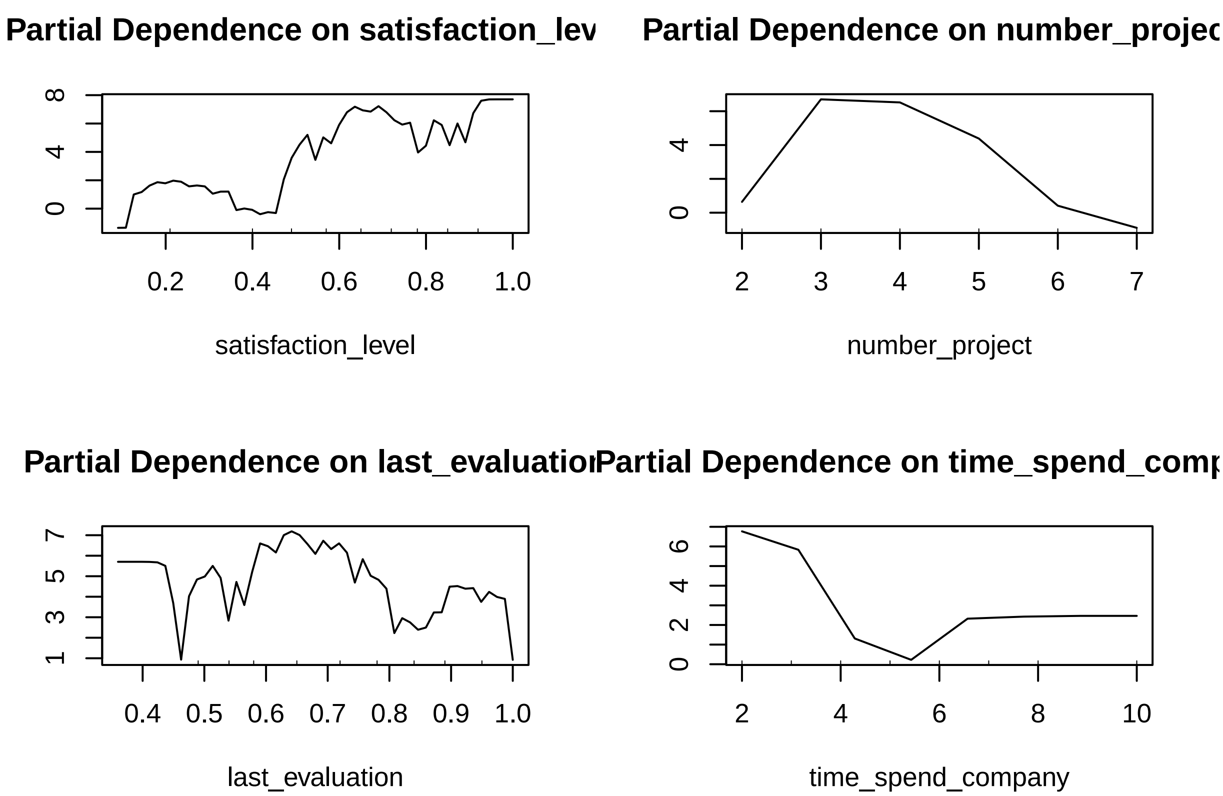


From the above output, it can be observed that satisfaction level, number project, last evaluation, time spend company and average monthly hours are the top five variables with highest association with the response variable left

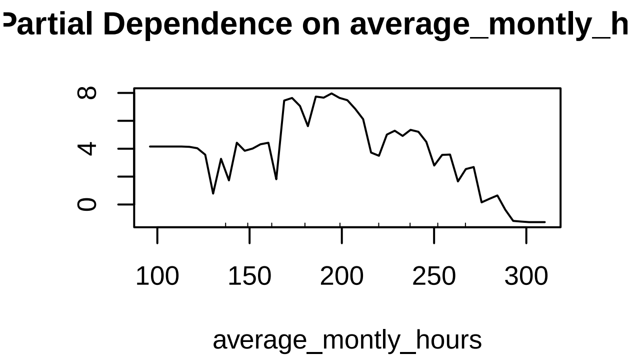
```r
yhat.rf <- predict(fit.rf, newdata=D2, type="prob")[, 2]
```

```r
#PARTIAL DEPENDENCE PLOT
par(mfrow=c(2,2))
partialPlot(fit.rf, pred.data=D1, x.var=satisfaction_level, rug=TRUE)
```

```r
partialPlot(fit.rf, pred.data=D1, x.var=number_project, rug=TRUE)
partialPlot(fit.rf, pred.data=D1, x.var=last_evaluation, rug=TRUE)
partialPlot(fit.rf, pred.data=D1, x.var=time_spend_company, rug=TRUE)
```




```r
partialPlot(fit.rf, pred.data=D1, x.var=average_montly_hours, rug=TRUE)
```



The above plots investigate the type of relationships between the top five important variables as declared by the variable importance ranking plot and the response variable, left.It can be observed that there are nonlinear relationships between each of the top five variables and the response variable.

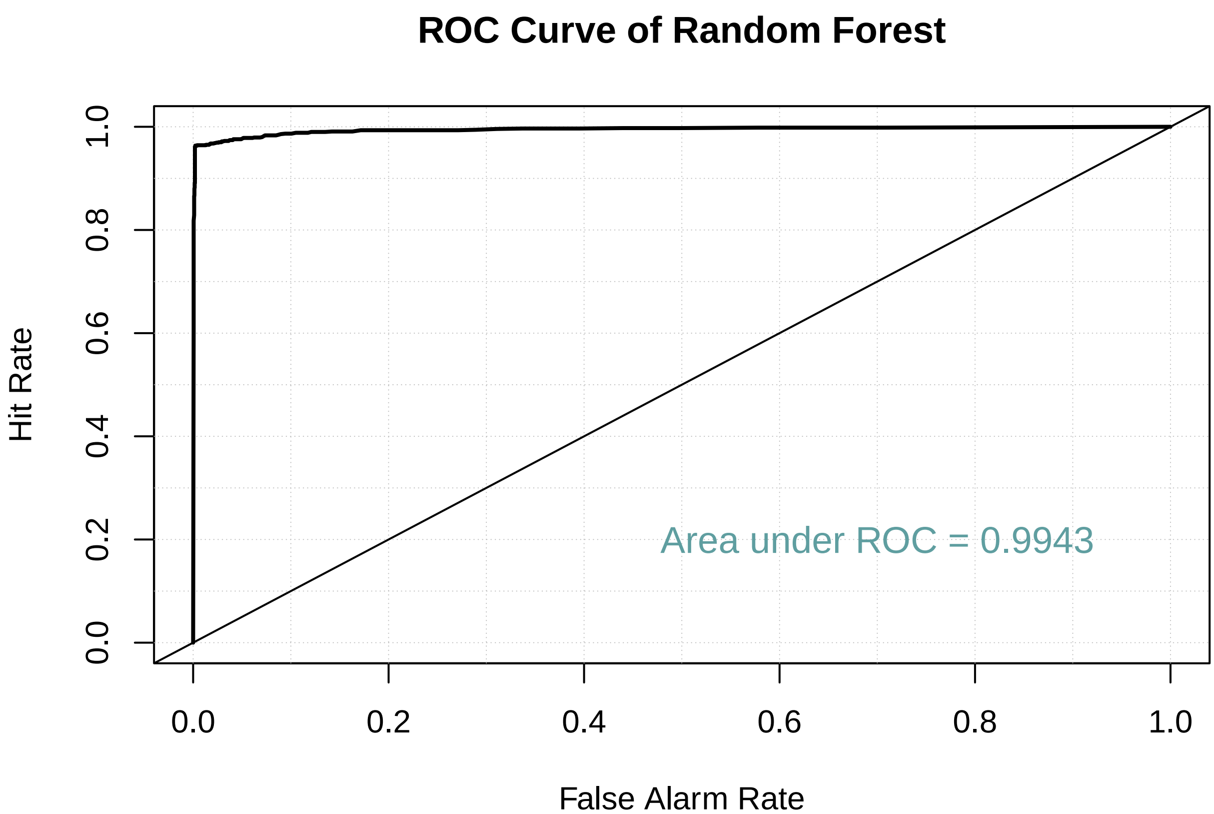
```
AUC.rf <- roc.area(obs=yobs, pred=yhat.rf)$A
mod.rf <- verify(obs=yobs, pred=yhat.rf)
```

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

```
AUC.rf
```

```
## [1] 0.9943102
```

```
roc.plot(mod.rf, plot.thres = NULL, col="green", main="ROC Curve of 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 area under the curve of the random forest model is 99.43% which is pretty good

GENERAL ADDICTIVE MODEL(GAM)

(6) Fit a generalized additive model. Explain how you determine the smoothing parameters 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

```
library(gam)
fit.gam <- gam( left ~ satisfaction_level + number_project + + time_spend_company +
department + last_evaluation + average_montly_hours + Work_accident + promotion_last_5years
+ salary , family = binomial,
data=D2, 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 = D2,
##     control = gam.control(epsilon = 1e-04, bf.epsilon = 1e-04,
##         maxit = 50, bf.maxit = 50), trace = TRUE)
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0989 -0.6630 -0.4018 -0.1016  2.8629
```

```
##
## (Dispersion Parameter for binomial family taken to be 1)
##
##     Null Deviance: 5517.706 on 4999 degrees of freedom
## Residual Deviance: 4277.395 on 4981 degrees of freedom
## AIC: 4315.395
##
## Number of Local Scoring Iterations: 4
##
## Anova for Parametric Effects
##                         Df Sum Sq Mean Sq  F value    Pr(>F)
## satisfaction_level       1  485.9  485.87 524.1893 < 2.2e-16 ***
## number_project           1    8.2    8.24   8.8947 0.0028738 **
## time_spend_company       1   58.0   57.95  62.5260 3.214e-15 ***
## department               9   27.8    3.09   3.3330 0.0004513 ***
## last_evaluation          1   16.9   16.91  18.2416 1.982e-05 ***
## average_montly_hours     1   23.9   23.88  25.7677 3.991e-07 ***
## Work_accident            1   86.3   86.32  93.1322 < 2.2e-16 ***
## promotion_last_5years    1   12.3   12.28  13.2478 0.0002757 ***
## salary                   2  129.4   64.71  69.8163 < 2.2e-16 ***
## Residuals             4981 4616.8    0.93
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
yhat.gam <- predict(fit.gam, newdata=D2, type="response", se.fit=FALSE)
```

MODEL SELECTION

```
# STEPWISE SELECTION
fit.step <- step.Gam(fit.gam, scope=list("satisfaction_level"=~1 +satisfaction_level + lo(satisfaction_l
"last_evaluation"=~1+ last_evaluation + lo(last_evaluation)+ s(last_evaluation , 2),
"number_project"=~1 + number_project + s(number_project, 2) + s(number_project, 4),
"average_montly_hours"=~1 + average_montly_hours + s(average_montly_hours, 2) + s(average_montly_hours,
"time_spend_company"=~1 + time_spend_company + s(time_spend_company, 2) + s(time_spend_company, 4)),
scale =2, steps=1000, parallel=TRUE, direction="both")
```

```
## Start:  left ~ satisfaction_level + number_project + +time_spend_company +      department + last_eva
```

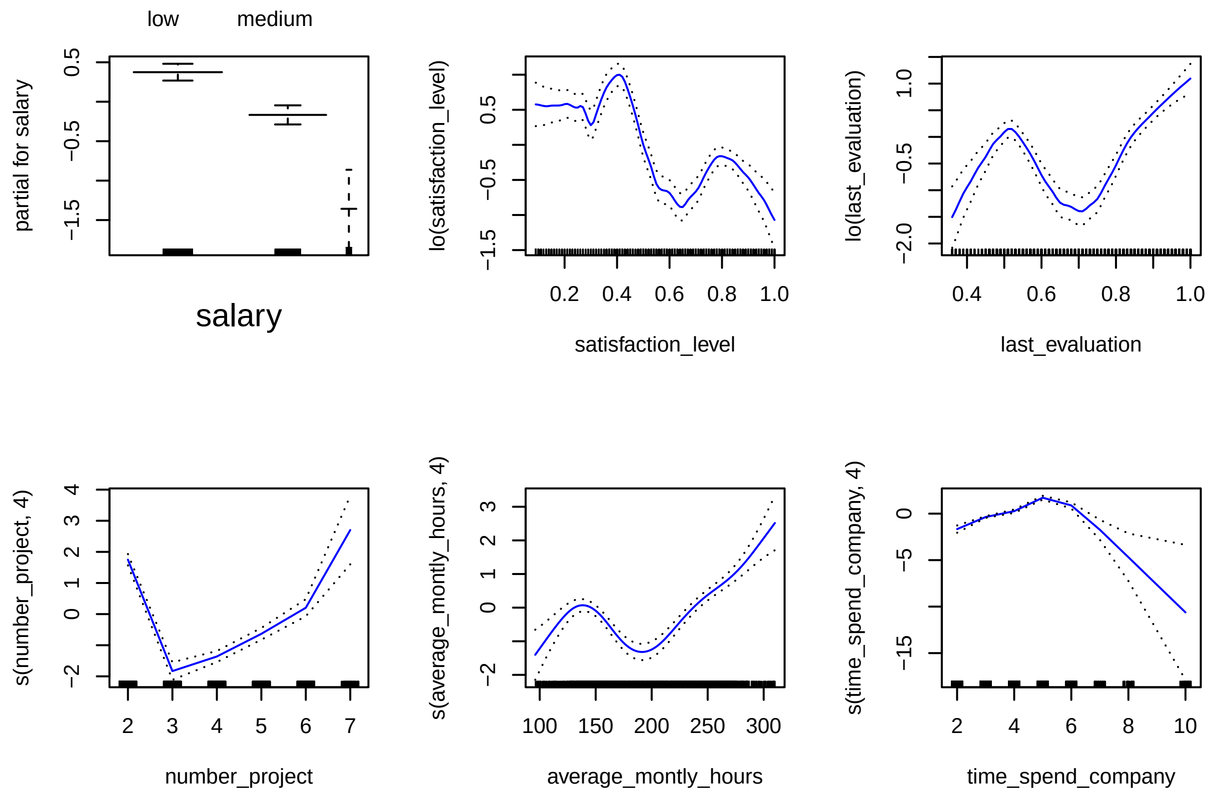
```
## Warning: executing %dopar% sequentially: no parallel backend registered
```

```
## Step:1 left ~ salary + satisfaction_level + last_evaluation + s(number_project,      2) + average_mon
## Step:2 left ~ salary + satisfaction_level + last_evaluation + s(number_project,      4) + average_mon
## Step:3 left ~ salary + lo(satisfaction_level) + last_evaluation + s(number_project,      4) + average
## Step:4 left ~ salary + lo(satisfaction_level) + lo(last_evaluation) +      s(number_project, 4) + ave
## Step:5 left ~ salary + lo(satisfaction_level) + lo(last_evaluation) +      s(number_project, 4) + ave
## Step:6 left ~ salary + lo(satisfaction_level) + lo(last_evaluation) +      s(number_project, 4) + ave
## Step:7 left ~ salary + lo(satisfaction_level) + lo(last_evaluation) +      s(number_project, 4) + s(a
## Step:8 left ~ salary + lo(satisfaction_level) + lo(last_evaluation) +      s(number_project, 4) + s(a
```

```
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 = D2, control = gam.control(epsilon = 1e-04,
##     bf.epsilon = 1e-04, maxit = 50, bf.maxit = 50), trace = FALSE)
## Deviance Residuals:
##       Min        1Q    Median        3Q       Max
## -2.620579 -0.320791 -0.136889 -0.001532  3.422091
##
## (Dispersion Parameter for binomial family taken to be 1)
##
##     Null Deviance: 5517.706 on 4999 degrees of freedom
## Residual Deviance: 2111.998 on 4978.146 degrees of freedom
## AIC: 2155.706
##
## Number of Local Scoring Iterations: 1
##
## Anova for Parametric Effects
##                                Df Sum Sq Mean Sq F value    Pr(>F)
## salary                        2.0   41.2  20.619  22.895 1.265e-10 ***
## lo(satisfaction_level)        1.0    9.2   9.235  10.255  0.001372 **
## lo(last_evaluation)           1.0   79.8  79.783  88.590 < 2.2e-16 ***
## s(number_project, 4)          1.0   16.4  16.430  18.243 1.980e-05 ***
## s(average_montly_hours, 4)    1.0   56.2  56.221  62.427 3.377e-15 ***
## s(time_spend_company, 4)      1.0  185.8 185.767 206.274 < 2.2e-16 ***
## Residuals                  4978.1 4483.2   0.901
## ---
## 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)         2.3     198.77 < 2.2e-16 ***
## lo(last_evaluation)            2.5     183.30 < 2.2e-16 ***
## s(number_project, 4)           3.0     457.18 < 2.2e-16 ***
## s(average_montly_hours, 4)     3.0     202.74 < 2.2e-16 ***
## s(time_spend_company, 4)       3.0     144.99 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

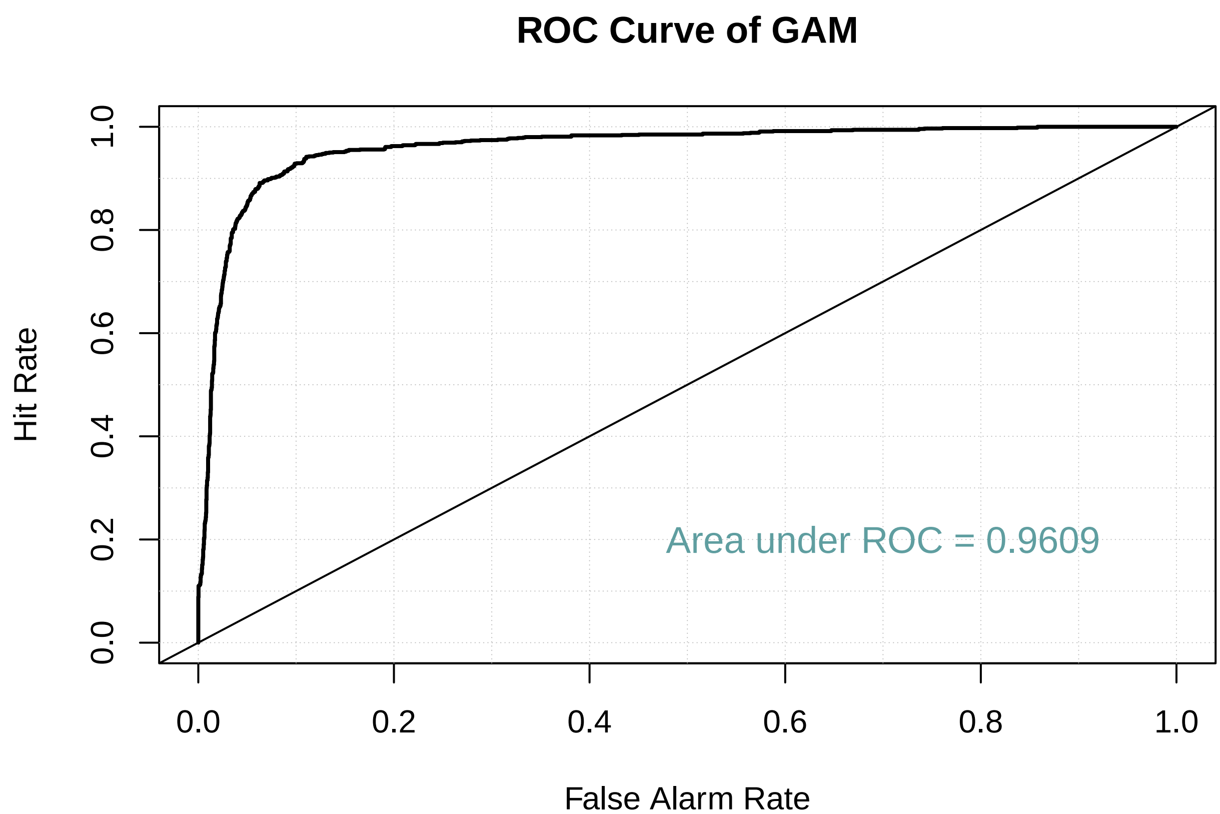


From the above plot, each smoothing parameter was determined adaptively in the back-fitting algorithm. In this scenario since smoothing splines are used, optimization of the tuning parameter is automatically done via minimum GCV. Also Step-wise selection with AIC was used to do the variable selection.

```
yhat.gam <- predict(fit.step, newdata=D2, 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="blue", main="ROC Curve of GAM")
text(x=0.7, y=0.2, paste("Area under ROC =", round(AUC.GAM, digits=4),
sep=" "), col="cadetblue", cex=1.2)
```



The area under the curve of the general addictive model is 96.09%

```
**MULTIVARIATE ADAPTIVE REGRESSION SPLINES**
```

(7) Train a multivariate adaptive regression splines model. Present the final model if possible. Obtain variable importance ranking and partial dependence plots (for continuous predictors only) to gain insights about what important factors predict employee detention or turnover.

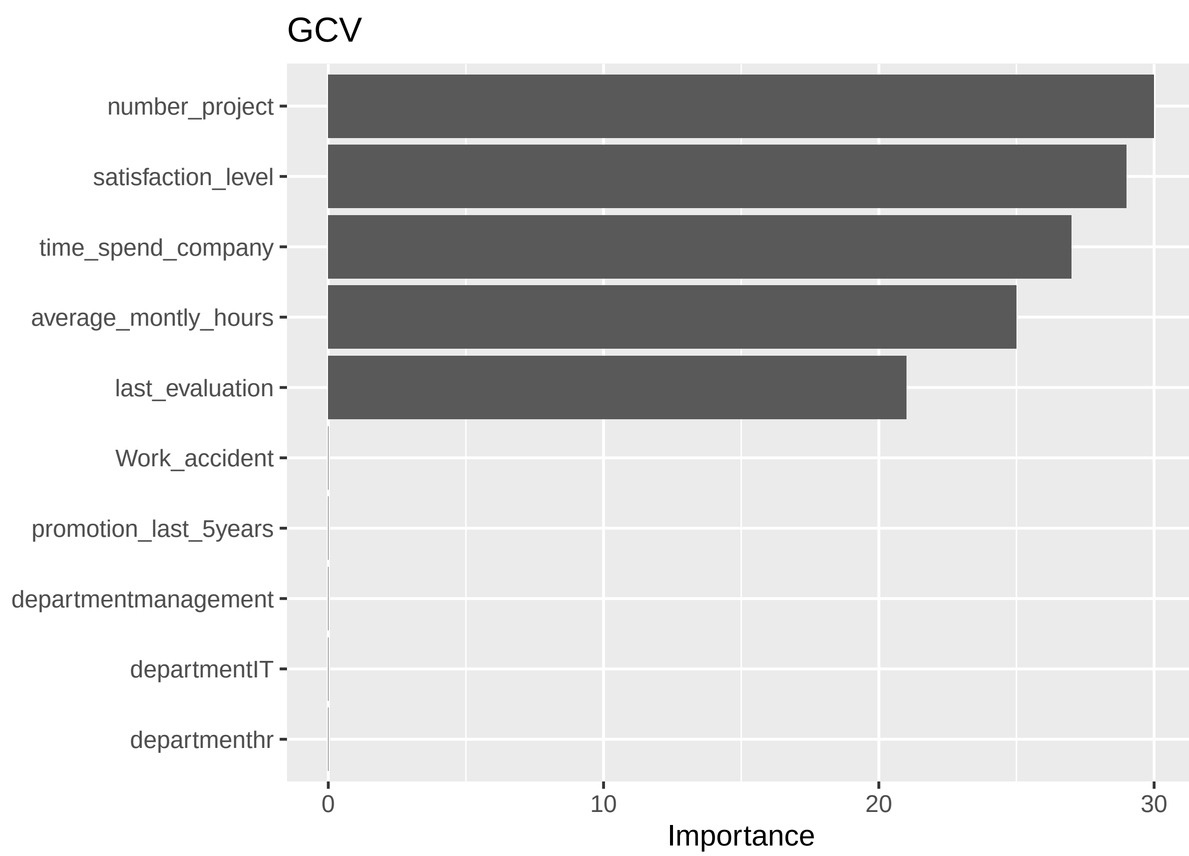
```
library("earth")
library(ggplot2) # plotting
library(caret) # automating the tuning process
library(vip) # variable importance
library(pdp) # variable relationships
fit.mars <- earth(left ~ ., data = D1, degree=3,
glm=list(family=binomial(link = "logit")))
print(fit.mars)
```

```
## GLM (family binomial, link logit):
##  nulldev   df       dev   df   devratio     AIC iters converged
##  10946.7 9998   2276.81 9968      0.792    2339    18         1
##
## Earth selected 31 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 15 11
## Earth GCV 0.0366708    RSS 361.1186    GRSq 0.797146    RSq 0.800178
```

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

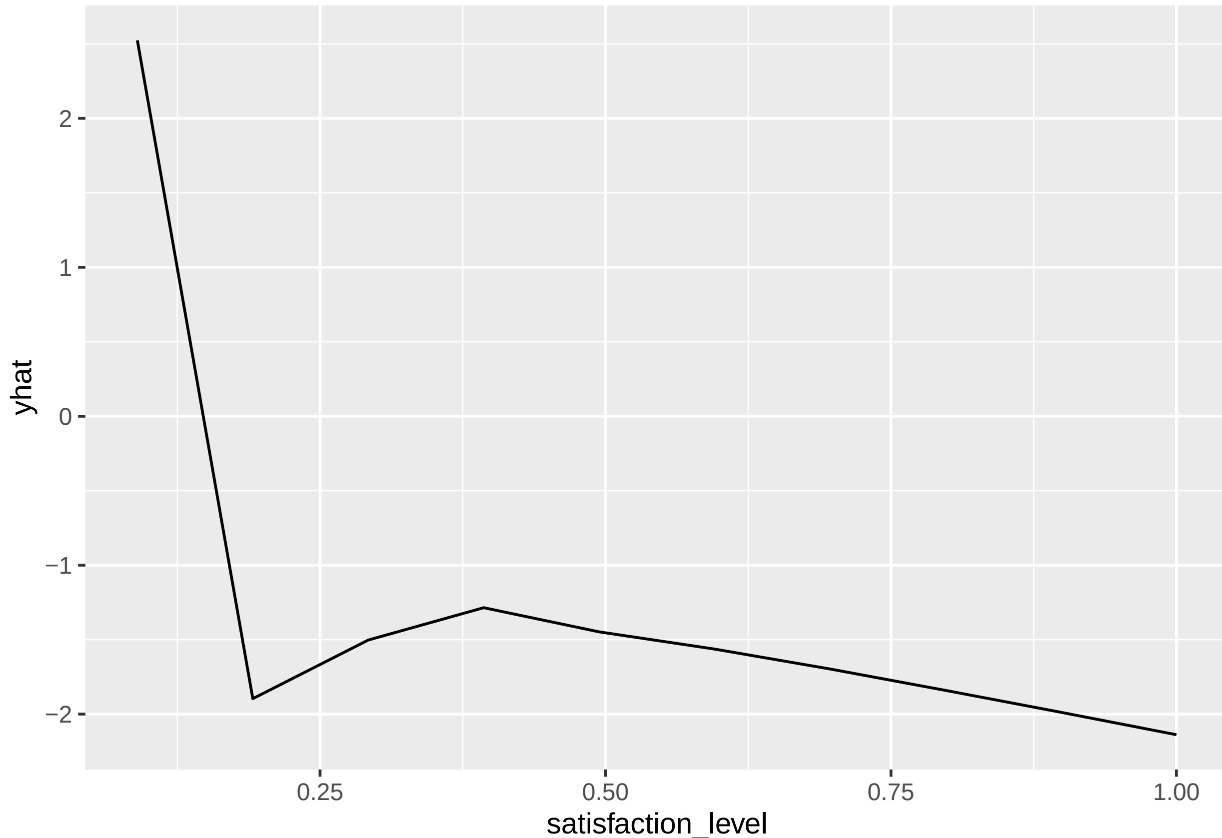
```
##                                                                               left
## (Intercept)                                                            -0.01663660
## h(number_project-3)                                                     0.03440938
## h(3-number_project)                                                     1.12544848
## h(number_project-3)*h(time_spend_company-5)                            -0.02054638
## h(number_project-3)*h(5-time_spend_company)                             0.02935624
## h(satisfaction_level-0.38)*h(3-number_project)                         -2.03862335
## h(0.38-satisfaction_level)*h(3-number_project)                         -2.02511373
## h(satisfaction_level-0.23)*h(number_project-3)                          0.14257280
## h(0.23-satisfaction_level)*h(number_project-3)                          0.35692027
## h(satisfaction_level-0.23)*h(last_evaluation-0.99)*h(number_project-3) 11.80171589
```

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



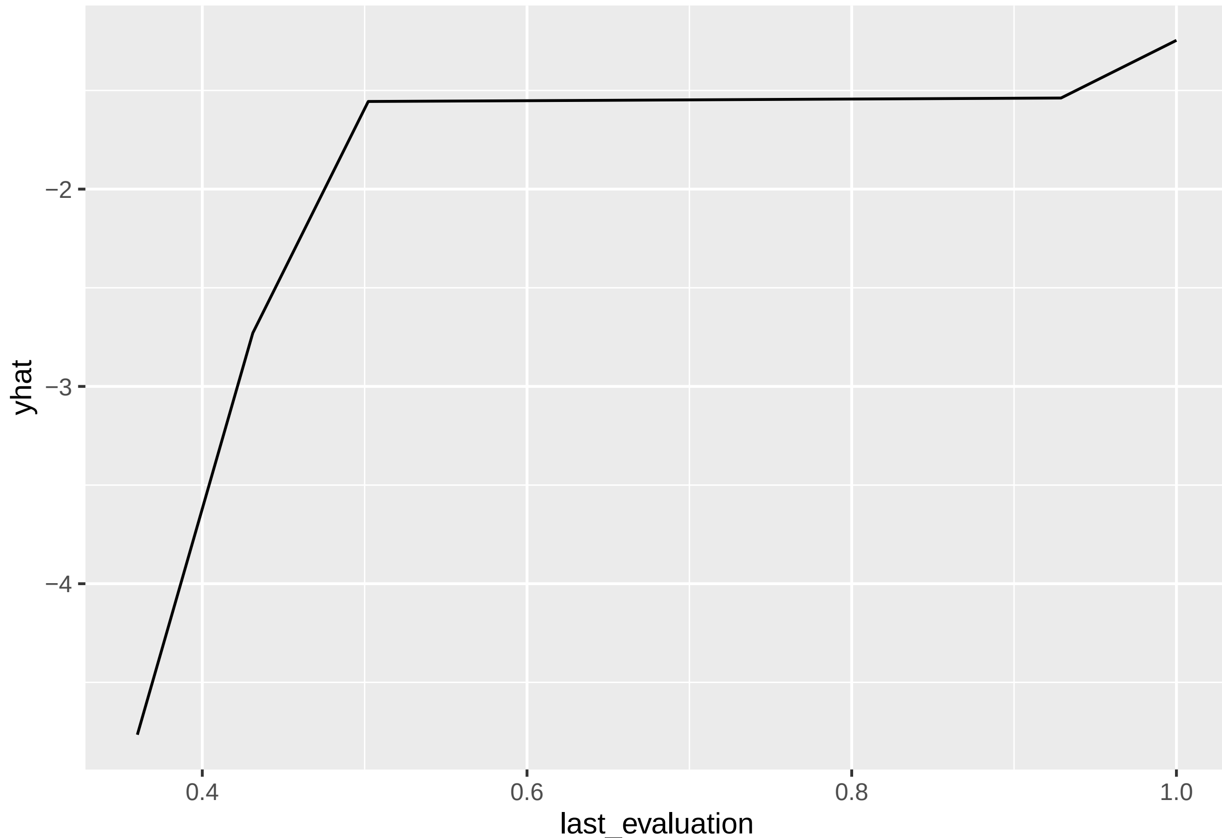
From the above variable importance ranking plot, The top two important continuous variables are satisfaction level and last evaluation. We now investigate the type of relationships between these two continuous variables and the response variable using the partial dependency plot.

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



From the above plot, it can be observed that there's a non-linear relationship between the response variable and satisfaction level. In particular, as the satisfaction level increases, the number of people who left decreases drastically,increased slightly and then decreased again.

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



It can be observed that there's a non-linear relationship between last evaluation and the response. typically, as last evaluation increases, there's an increase in the people leaving and stays constant for some time and then slightly increase

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

```
## $cvAUC
## [1] 0.9806366
##
## $se
## [1] 0.002619765
##
## $ci
## [1] 0.9755019 0.9857712
##
## $confidence
## [1] 0.95
```

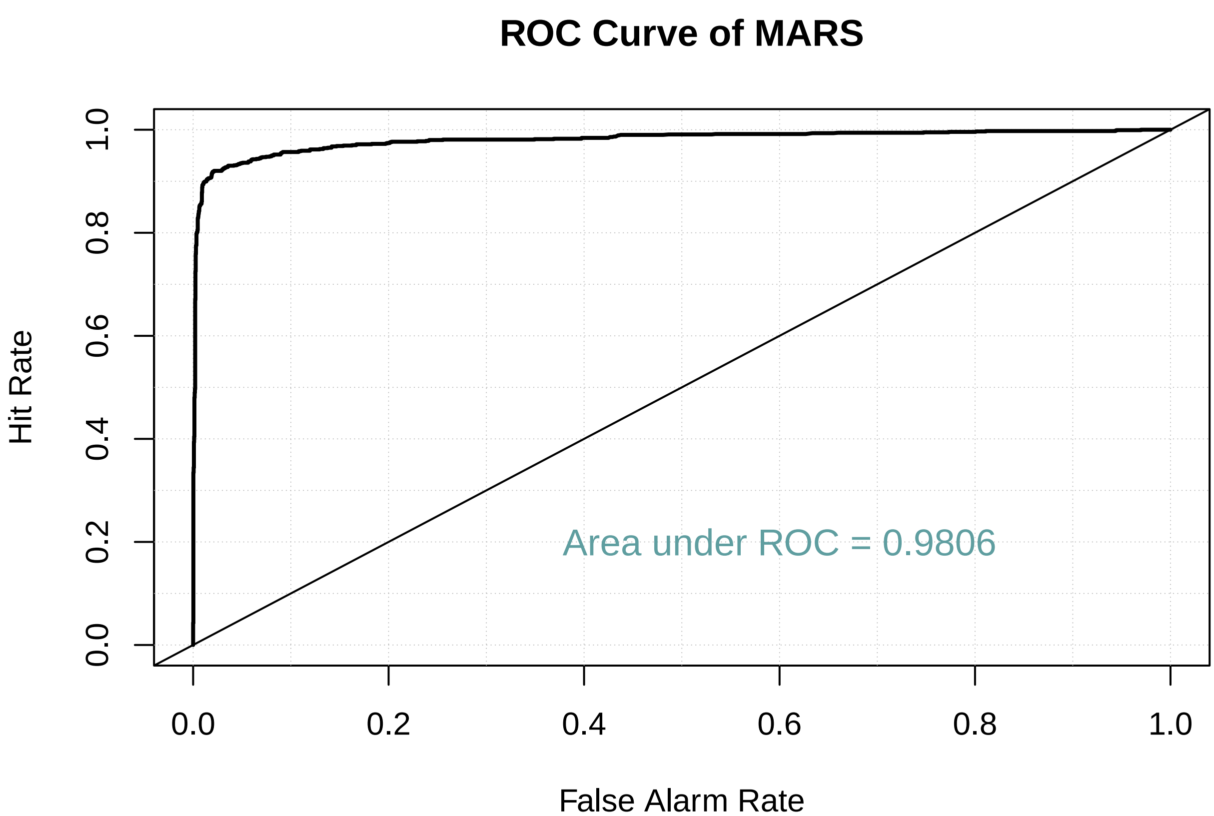
```
auc.ci <- round(AUC.MARS$ci, digits=4)
library(verification)
mod.mars <- verify(obs=yobs, pred=yhat.mars)
```

```
## 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 of MARS")
```

```
## 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.MARS$cvAUC, digits=4),
sep=" "), col="cadetblue", cex=1.2)
```



The area under the curve of MARS is 98.06%

PROJECT PURSUIT REGRESSION MODEL

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

```r
fit.ppr <- ppr(left ~ ., sm.method = "supsmu",data = D1, nterms = 2, max.terms = 10, bass=3)
summary(fit.ppr)
```

```
## Call:
## ppr(formula = left ~ ., data = D1, 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
## 653.3792 566.8241 450.4446 433.3265 426.9316 426.3276 388.0087 379.1495
## 10 terms
## 380.1441
##
## Projection direction vectors ('alpha'):
##                        term 1        term 2
## satisfaction_level    -0.1486939220 -0.3939244244
## last_evaluation        0.2320430348 -0.1698126882
## number_project         0.0496717950 -0.0440037254
## average_montly_hours   0.0009788920 -0.0006615958
## time_spend_company    -0.0889903496  0.2101185501
## Work_accident         -0.0645464856 -0.0069741546
## promotion_last_5years  0.1609559131 -0.5866733185
## departmentaccounting  -0.2942203051 -0.2036728378
## departmenthr          -0.2909243644 -0.1980620547
## departmentIT          -0.3116178821 -0.2074080370
## departmentmanagement  -0.2878002312 -0.2109121254
## departmentmarketing   -0.2868814285 -0.2078872280
## departmentproduct_mng -0.2995765408 -0.2115477966
## departmentRandD       -0.3120006058 -0.2050447806
## departmentsales       -0.2987558719 -0.2057280474
## departmentsupport     -0.2941332351 -0.2093518822
## departmenttechnical   -0.2933890849 -0.2027749082
## salary.L              -0.0315484962 -0.0118106498
## salary.Q              -0.0146623403 -0.0055550946
##
## Coefficients of ridge terms ('beta'):
##    term 1    term 2
## 0.1244648 0.3269095
```

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

```
## Call:
## ppr(formula = left ~ ., data = D1, 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
## 467.6335 457.4933 429.8918 425.5590 421.4353   0.0000   0.0000
##
## Projection direction vectors ('alpha'):
##                        term 1        term 2        term 3        term 4
## satisfaction_level    -0.1992951814 -0.4290457905  0.1789370683  0.0446352864
## last_evaluation       -0.1454360589  0.0911719102  0.1892995875  0.1650555548
## number_project        -0.0254030149  0.0854218402 -0.0093024855  0.0384001009
## average_montly_hours  -0.0003531472  0.0014699009  0.0002467639  0.0005300376
## time_spend_company     0.1239601645 -0.1072694961  0.0228870199  0.0089483457
## Work_accident         -0.3204161583  0.0307045108  0.0120646958 -0.0038481509
## promotion_last_5years -0.3982687762  0.1604580138 -0.0394211631 -0.0113967955
## departmentaccounting  -0.2371822468  0.2927291204 -0.3060885603 -0.3112351469
## departmenthr          -0.2201651510  0.2831900396 -0.2994486403 -0.3070850946
## departmentIT          -0.2288357602  0.2586289508 -0.3083257379 -0.3091907148
```

```
## departmentmanagement  -0.2380456307  0.2934283767 -0.3211731778 -0.3131422309
## departmentmarketing   -0.2427039390  0.2523087623 -0.2950314156 -0.3157450394
## departmentproduct_mng -0.2471144839  0.2651855802 -0.3033431983 -0.3116865701
## departmentRandD       -0.2408211049  0.2498604934 -0.3058130421 -0.3100727358
## departmentsales       -0.2382896557  0.2817389483 -0.3071889880 -0.3131797051
## departmentsupport     -0.2438228803  0.2815583149 -0.2984049852 -0.3118263777
## departmenttechnical   -0.2374072654  0.2924640370 -0.3036476366 -0.3096297144
## salary.L              -0.2711376025  0.0413516004 -0.0057573530 -0.0061063291
## salary.Q              -0.1549164624  0.0304222773  0.0057127475 -0.0050593537
##
## Coefficients of ridge terms ('beta'):
##    term 1    term 2    term 3    term 4
## 0.1499923 0.1649863 0.1286543 0.2391406
```

```
# PREDICTION
yhat.ppr <- predict(fit1.ppr, newdata=D2)
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))
AUC.PPR
```

```
## $cvAUC
## [1] 0.9679935
##
## $se
## [1] 0.003371243
##
## $ci
## [1] 0.961386 0.974601
##
## $confidence
## [1] 0.95
```

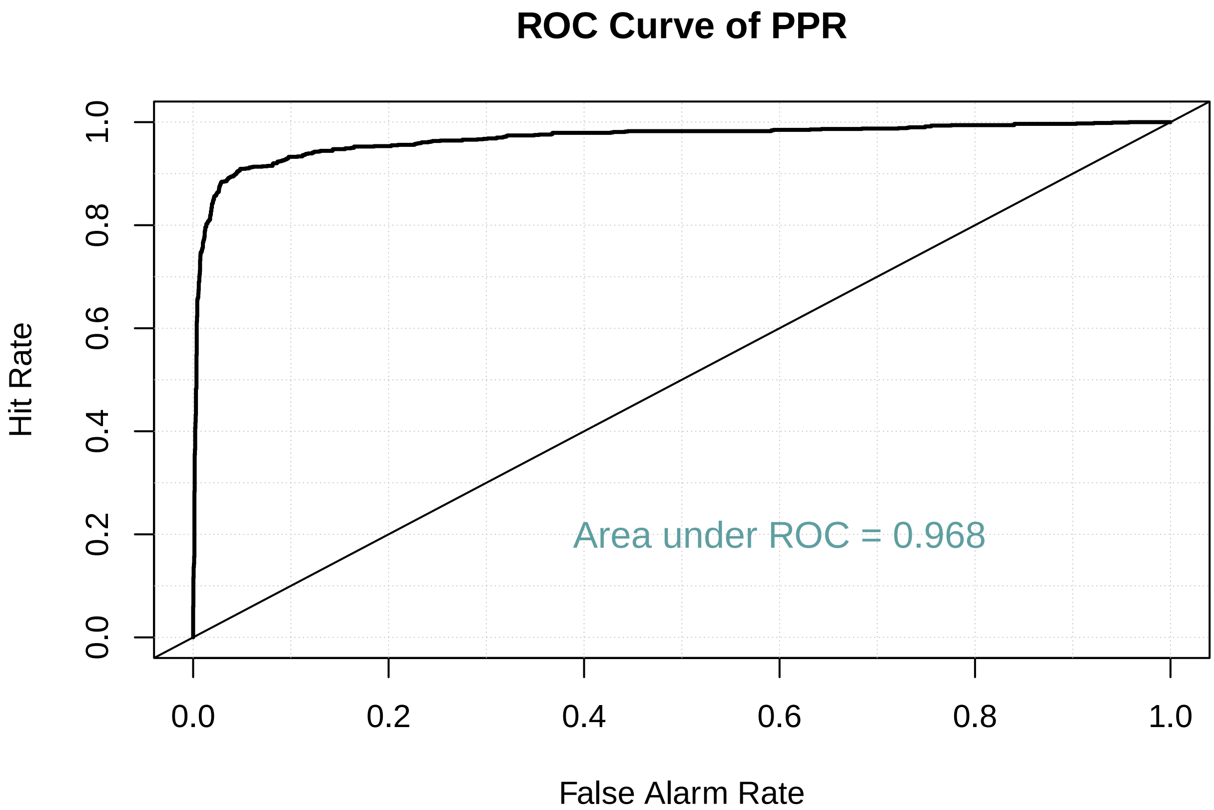
```
auc.ci <- round(AUC.PPR$ci, digits=4)
library(verification)
mod.ppr <- verify(obs=yobs, pred=yhat.ppr)
```

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

```
roc.plot(mod.ppr, plot.thres = NULL, main="ROC Curve of 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.
```

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



The area under the curve of the project pursuit regression model is 96.8%

COMPARING RESULTS

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

```
##          Method    AUC
## 1         LASSO 0.8220
## 2 Random Forest 0.9943
## 3           GAM 0.9609
## 4          MARS 0.9806
## 5           PPR 0.9680
```

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

Method	AUC
LASSO	0.8220
Random Forest	0.9943
GAM	0.9609
MARS	0.9806
PPR	0.9680

From the above results, among the five supervised learning approaches, Random forest gave the best results since it has the largest area under the curve. Thus, the random forest model did best in correctly predicting *the probability of employees turnover in the company. Also, among all the methods, we see that satisfaction level and number of projects are the top two important variables that predict an employees turnover in the company*