Employee Attrition using ANN

PVS

Friday, March 03, 2017

nn <- read.csv("D:/NN/1452762979\_586\_\_HR\_Employee\_Attrition\_Data.csv",sep=",",header=T)  
  
library(partykit)

## Loading required package: grid

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

set.seed(1234)  
  
nn$random <- runif(nrow(nn), 0, 1);  
nn.dev <- nn[which(nn$random <= 0.7),]  
nn.holdout <- nn[which(nn$random > 0.7),]  
  
##occ.matrix <- model.matrix(~ Occupation - 1, data = nn.dev)  
##nn.dev <- data.frame(nn.dev, occ.matrix)  
  
##Gender.matrix <- model.matrix(~ Gender - 1, data = nn.dev)  
##nn.dev <- data.frame(nn.dev, Gender.matrix)  
  
  
##occ.matrix <- model.matrix(~ Occupation - 1, data = nn.holdout)  
##nn.holdout <- data.frame(nn.holdout, occ.matrix)  
  
##Gender.matrix <- model.matrix(~ Gender - 1, data = nn.holdout)  
##nn.holdout <- data.frame(nn.holdout, Gender.matrix)  
  
c(nrow(nn.dev), nrow(nn.holdout))

## [1] 2082 858

str(nn.dev)

## 'data.frame': 2082 obs. of 36 variables:  
## $ Age : int 41 49 37 33 32 59 30 38 36 35 ...  
## $ Attrition : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 1 1 1 1 ...  
## $ BusinessTravel : Factor w/ 3 levels "Non-Travel","Travel\_Frequently",..: 3 2 3 2 2 3 3 2 3 3 ...  
## $ DailyRate : int 1102 279 1373 1392 1005 1324 1358 216 1299 809 ...  
## $ Department : Factor w/ 3 levels "Human Resources",..: 3 2 2 2 2 2 2 2 2 2 ...  
## $ DistanceFromHome : int 1 8 2 3 2 3 24 23 27 16 ...  
## $ Education : int 2 1 2 4 2 3 1 3 3 3 ...  
## $ EducationField : Factor w/ 6 levels "Human Resources",..: 2 2 5 2 2 4 2 2 4 4 ...  
## $ EmployeeCount : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ EmployeeNumber : int 1 2 3 4 6 7 8 9 10 11 ...  
## $ EnvironmentSatisfaction : int 2 3 4 4 4 3 4 4 3 1 ...  
## $ Gender : Factor w/ 2 levels "Female","Male": 1 2 2 1 2 1 2 2 2 2 ...  
## $ HourlyRate : int 94 61 92 56 79 81 67 44 94 84 ...  
## $ JobInvolvement : int 3 2 2 3 3 4 3 2 3 4 ...  
## $ JobLevel : int 2 2 1 1 1 1 1 3 2 1 ...  
## $ JobRole : Factor w/ 9 levels "Healthcare Representative",..: 8 7 3 7 3 3 3 5 1 3 ...  
## $ JobSatisfaction : int 4 2 3 3 4 1 3 3 3 2 ...  
## $ MaritalStatus : Factor w/ 3 levels "Divorced","Married",..: 3 2 3 2 3 2 1 3 2 2 ...  
## $ MonthlyIncome : int 5993 5130 2090 2909 3068 2670 2693 9526 5237 2426 ...  
## $ MonthlyRate : int 19479 24907 2396 23159 11864 9964 13335 8787 16577 16479 ...  
## $ NumCompaniesWorked : int 8 1 6 1 0 4 1 0 6 0 ...  
## $ Over18 : Factor w/ 1 level "Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ OverTime : Factor w/ 2 levels "No","Yes": 2 1 2 2 1 2 1 1 1 1 ...  
## $ PercentSalaryHike : int 11 23 15 11 13 20 22 21 13 13 ...  
## $ PerformanceRating : int 3 4 3 3 3 4 4 4 3 3 ...  
## $ RelationshipSatisfaction: int 1 4 2 3 3 1 2 2 2 3 ...  
## $ StandardHours : int 80 80 80 80 80 80 80 80 80 80 ...  
## $ StockOptionLevel : int 0 1 0 0 0 3 1 0 2 1 ...  
## $ TotalWorkingYears : int 8 10 7 8 8 12 1 10 17 6 ...  
## $ TrainingTimesLastYear : int 0 3 3 3 2 3 2 2 3 5 ...  
## $ WorkLifeBalance : int 1 3 3 3 2 2 3 3 2 3 ...  
## $ YearsAtCompany : int 6 10 0 8 7 1 1 9 7 5 ...  
## $ YearsInCurrentRole : int 4 7 0 7 7 0 0 7 7 4 ...  
## $ YearsSinceLastPromotion : int 0 1 0 3 3 0 0 1 7 0 ...  
## $ YearsWithCurrManager : int 5 7 0 0 6 0 0 8 7 3 ...  
## $ random : num 0.114 0.622 0.609 0.623 0.64 ...

## Installing the Neural Net package;

## If already installed do not run the below step

## install.packages("neuralnet")

library(neuralnet)

nn1 <- neuralnet(formula = Target ~ Age + Balance + SCR + No\_OF\_CR\_TXNS + Holding\_Period , data = nn.dev, hidden = 3, err.fct = "sse", linear.output = FALSE, lifesign = "full", lifesign.step = 10, threshold = 0.1, stepmax = 2000 )

plot (nn1)

quantile(nn1$net.result[[1]], c(0,1,5,10,25,50,75,90,95,99,100)/100) ## The distribution of the estimated results

misClassTable = data.frame(Target = nn.devnet.result[[1]] ) misClassTablePrediction>0.143,1,0) with(misClassTable, table(Target, Classification))

## We can use the confusionMatrix function of the caret package

## install.packages("caret")

library(caret) confusionMatrix(misClassTableClassification)

## build the neural net model by scaling the variables

x <- subset(nn.dev, select = c("Age","Balance", "SCR", "No\_OF\_CR\_TXNS", "Holding\_Period" ##,"OccupationPROF", "OccupationSAL", "OccupationSELF.EMP", "OccupationSENP","GenderF", "GenderM", "GenderO" ) ) nn.devscaled <- scale(x) nn.devscaled <- cbind(nn.dev[2], nn.devscaled) View(nn.devscaled)

nn2 <- neuralnet(formula = Target ~ Age + Balance + SCR + No\_OF\_CR\_TXNS + Holding\_Period , ## + OccupationPROF + OccupationSAL + OccupationSELF.EMP + OccupationSENP + GenderF + GenderM + GenderO, data = nn.devscaled, hidden = 3, err.fct = "sse", linear.output = FALSE, lifesign = "full", lifesign.step = 10, threshold = 0.1, stepmax = 2000 ) nn2 plot(nn2)

quantile(nn2$net.result[[1]], c(0,1,5,10,25,50,75,90,95,99,100)/100)

misClassTable = data.frame(Target = nn.devscalednet.result[[1]] ) misClassTablePredict.score>0.21,1,0) with(misClassTable, table(Target, Predict.class))

confusionMatrix(misClassTableTarget)

## Error Computation

sum((misClassTablePredict.score)^2)/2

## deciling code

decile <- function(x){ deciles <- vector(length=10) for (i in seq(0.1,1,.1)){ deciles[i\*10] <- quantile(x, i, na.rm=T) } return ( ifelse(x<deciles[1], 1, ifelse(x<deciles[2], 2, ifelse(x<deciles[3], 3, ifelse(x<deciles[4], 4, ifelse(x<deciles[5], 5, ifelse(x<deciles[6], 6, ifelse(x<deciles[7], 7, ifelse(x<deciles[8], 8, ifelse(x<deciles[9], 9, 10 )))))))))) }

## deciling

misClassTablePredict.score)

## Ranking code

## install.packages("data.table")

library(data.table) tmp\_DT = data.table(misClassTable) rank <- tmp\_DT[, list( cnt = length(Target), cnt\_resp = sum(Target), cnt\_non\_resp = sum(Target == 0)) , by=deciles][order(-deciles)] rankcnt\_resp / rankcum\_resp <- cumsum(rankcum\_non\_resp <- cumsum(rankcum\_rel\_resp <- round(rankcnt\_resp),2); rankcum\_non\_resp / sum(rankks <- abs(rankcum\_rel\_non\_resp);

library(scales) rankrrate) rankcum\_rel\_resp) rankcum\_rel\_non\_resp)

View(rank)

## Scoring another dataset using the Neural Net Model Object

## To score we will use the compute function

?compute x <- subset(nn.holdout, select = c("Age","Balance", "SCR", "No\_OF\_CR\_TXNS", "Holding\_Period") ) x.scaled <- scale(x) compute.output = compute(nn2, x.scaled) nn.holdoutnet.result View(nn.holdout) quantile(nn.holdout$Predict.score, c(0,1,5,10,25,50,75,90,95,99,100)/100)

## Building Neural Net Moodel using caret package

dev\_trainplot = predict( preProcess(nn.dev[,-1], method="range"), nn.dev[,-1] ) nrow(nn.dev) featurePlot(dev\_trainplot[, lapply(dev\_trainplot, class) %in% c("numeric", "integer")], as.factor(nn.dev$Target), "box")

## install.packages("e1071")

## install.packages("nnet")

cv\_opts = trainControl(method="cv", number=10) library(e1071) head(nn.dev) results\_nnet = train(as.factor(Target)~., data=nn.dev[,-1], method="avNNet", trControl=cv\_opts, preProcess="range", tuneLength=2, trace=T, maxit=1000) results\_nnet

preds\_nnet = predict(results\_nnet, nn.dev[,-1]) confusionMatrix(preds\_nnet, as.factor(nn.dev$Target))

```