```
R version 3.2.2 (2015-08-14) -- "Fire Safety"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> setwd("C:/Users/TonyLaptop/Desktop/rowan/DM1/lecture9/lecture9")
>  data<-read.csv("censusAdult50k.csv",header = TRUE, sep = ",", stringsAsFactors = TRUE)
>  library(plyr)
>
>
>
>  minn = min(data$Age)
>  maxx = max(data$Age)
>  data$Age2 <- sapply(data$Age,function(b) {
+      return((b-minn)/(maxx-minn))
+  }
+  )
>
>
>
> minn = min(data$CapitalGain)
> maxx = max(data$CapitalGain)
> data$Gain2 <- sapply(data$CapitalGain,function(b) {
+     return((b-minn)/(maxx-minn))
+ }
+ )
```

```
>
> minn = min(data$CapitalLoss)
> maxx = max(data$CapitalLoss)
> data$Loss2 <- sapply(data$CapitalLoss,function(b) {
+     return((b-minn)/(maxx-minn))
+ }
+ )
>
> minn = min(data$EducationNum)
> maxx = max(data$EducationNum)
> data$Educ2 <- sapply(data$EducationNum,function(b) {
+     return((b-minn)/(maxx-minn))
+ }
+ )
>
> minn = min(data$HoursPerWeek)
> maxx = max(data$HoursPerWeek)
> data$Hours2 <- sapply(data$HoursPerWeek,function(b) {
+     return((b-minn)/(maxx-minn))
+ }
+ )
> m <-model.matrix(~ Income + Age2 + Educ2 + MaritalStatus + Relationship + Race + Sex + Hours2 +
Job+Gain2 + Loss2, data=data)
>
> names(m[1,])
 [1] "(Intercept)"                          "Income >50K"                          "Age2"
 [4] "Educ2"                                "MaritalStatus Married-AF-spouse"      "MaritalStatus
Married-civ-spouse"
 [7] "MaritalStatus Married-spouse-absent" "MaritalStatus Never-married"          "MaritalStatus
Separated"
[10] "MaritalStatus Widowed"                "Relationship Not-in-family"           "Relationship Other-
relative"
[13] "Relationship Own-child"               "Relationship Unmarried"               "Relationship Wife"
[16] "Race Asian-Pac-Islander"              "Race Black"                           "Race Other"
[19] "Race White"                           "Sex Male"                             "Hours2"
[22] "Job Armed-Forces"                     "Job Craft-repair"                     "Job Exec-
managerial"
[25] "Job Farming-fishing"                  "Job Handlers-cleaners"                "Job Machine-op-
inspct"
```

```
[28] "Job Other"                        "Job Other-service"                "Job Priv-house-
serv"
[31] "Job Prof-specialty"               "Job Protective-serv"              "Job Sales"
[34] "Job Tech-support"                 "Job Transport-moving"             "Gain2"
[37] "Loss2"
>colnames(m)=c("dk","Income","Age2","Educ2","MarStat1","MarStat2","MarStat3","Unmarried","Separated",
"Widowed","Rel1","Rel2","Rel3","Unmarried2","Wife","Race1","Black","RaceOther","White","Male","Hours"
,"Milit","Craft","Executive","Farmer","Cleaner","MachineOp","JobOth","OthServ","HouseServ","ProfSpec"
,"ProtServ","Sales","TechSup","Moving","Gain","Loss")
> v<-as.data.frame(m[,2:37])
>
> set.seed(2)
> train=sample(1:nrow(v),nrow(v)/2)
> test=-train
>
> trainingData=v[train,]
> testingData=v[test,]
> formula =
Income~Age2+Educ2+MarStat1+MarStat2+MarStat3+Unmarried+Separated+Widowed+Rel1+Rel2+Rel3+Unmarried2+
Wife+Race1+Black+RaceOther+White+Male+Hours+Milit+Craft+Executive+Farmer+
Cleaner+MachineOp+JobOth+OthServ+HouseServ+ProfSpec+ProtServ+Sales+TechSup+Moving+Gain+Loss
> library("neuralnet")
Loading required package: grid
Loading required package: MASS
> nnet<-neuralnet(formula,trainingData, hidden=5, threshold=0.5)
> results<-compute(nnet,testingData[,2:36])
> testingData$result<- sapply(results$net.result, function(b) {
+     if (b<=.5){
+        return(0)
+      }else{
+        return(1)
+ }})
> count(testingData,c('Income','result'))
  Income result  freq
1      0      0 11494
2      0      1   843
3      1      0  1569
4      1      1  2375
> err=843+1569
> dim(testingData)
```

```
[1] 16281    37
> err/16281
[1] 0.1481481481
>
> nnet<-neuralnet(formula,trainingData, hidden=20, threshold=0.5)
> results<-compute(nnet,testingData[,2:36])
> testingData$result<- sapply(results$net.result, function(b) {
+    if (b<=.5){
+       return(0)
+     }else{
+       return(1)
+  }})
> count(testingData,c('Income','result'))
  Income result  freq
1      0      0 11348
2      0      1   989
3      1      0  1458
4      1      1  2486
> err=989+1458
> err/16281
[1] 0.1502978932
> nnet<-neuralnet(formula,trainingData, hidden=15, threshold=0.5)
> results<-compute(nnet,testingData[,2:36])
> testingData$result<- sapply(results$net.result, function(b) {
+    if (b<=.5){
+       return(0)
+     }else{
+       return(1)
+  }})
>
> count(testingData,c('Income','result'))
  Income result  freq
1      0      0 11427
2      0      1   910
3      1      0  1541
4      1      1  2403
> err=910+1541
> err/16281
[1] 0.1505435784
> nnet<-neuralnet(formula,trainingData, hidden=8, threshold=0.4)
```

```
> results<-compute(nnet,testingData[,2:36])
> testingData$result<- sapply(results$net.result, function(b) {
+     if (b<=.5){
+        return(0)
+      }else{
+        return(1)
+ }})
> count(testingData,c('Income','result'))
  Income result  freq
1      0      0 11501
2      0      1   836
3      1      0  1510
4      1      1  2434
> err=836+1510
> err/16281
[1] 0.1440943431
> nnet<-neuralnet(formula,trainingData, hidden=5, threshold=0.4)
> results<-compute(nnet,testingData[,2:36])
> testingData$result<- sapply(results$net.result, function(b) {
+     if (b<=.5){
+        return(0)
+      }else{
+        return(1)
+ }})
> count(testingData,c('Income','result'))
  Income result  freq
1      0      0 11380
2      0      1   957
3      1      0  1494
4      1      1  2450
> err=957+1494
> err/16281
[1] 0.1505435784
```