R Zone 4: Chapter 12 and 15

Brandon Hufstetler, Garrett Alarcon, Jinan Andrews, Anson Cheng, Nick Forrest, and Nestor Herandez

5 August 2019

Chapter 12

READ IN AND PREPARE THE DATA

```
setwd("~/IMGT680")
adult <- read.csv(file = "adult.txt", stringsAsFactors=TRUE)

# Collapse categories as in Chapter 11
# We will work with a small sample of data
adult <- adult[1:500,]
```

DETERMINE HOW MANY INDICATOR VARIABLES ARE NEEDED

Here we are looking at how many different unique levels(categories) are present within each feature in the data. This shows that there are 2 different levels of income, 2 different levels of gender, 5 different levels of race, 9 different levels of workclass, and 7 different levels of marital status in the adult.txt dataset.

```
unique(adult$income)            # One variable for income
```

```
## [1] <=50K. >50K.
## Levels: <=50K. >50K.
```

```
unique(adult$sex)                    # One variable for sex
```

```
## [1] Male   Female
## Levels: Female Male
```

```
unique(adult$race)                   # Four variables for race
```

```
## [1] White              Black              Asian-Pac-Islander
## [4] Amer-Indian-Eskimo Other
## Levels: Amer-Indian-Eskimo Asian-Pac-Islander Black Other White
```

```
unique(adult$workclass)         # Three variables for workclass
```

```
## [1] State-gov        Self-emp-not-inc Private          Federal-gov
## [5] Local-gov        ?                Self-emp-inc
## 9 Levels: ? Federal-gov Local-gov Never-worked Private ... Without-pay
```

```
unique(adult$marital.status)   # Four variables for marital.status
```

```
## [1] Never-married         Married-civ-spouse    Divorced
## [4] Married-spouse-absent Separated             Married-AF-spouse
## [7] Widowed
## 7 Levels: Divorced Married-AF-spouse ... Widowed
```

CREATE INDICATOR VARIABLES

First, we define the new indicator variables that we want to populate. These will all be binary indicators, so if a specific level is present in the data, the value of the indicator is equal to 1. The levels in this data are mutually exclusive, so only one level from each feature can occur in one observation. Therefore, it is not necessary to make an indicator variable for every level contained in a feature. The level that is left out is the assume baseline, as it is the only other option given the other indicator variables from that feature are 0.

```
adult$race_white <- adult$race_black <- adult$race_as.pac.is <-
adult$race_am.in.esk <- adult$wc_gov <-  adult$wc_self <- adult$wc_priv <-
adult$ms_marr <- adult$ms_div <- adult$ms_sep <- adult$ms_wid <-
adult$income_g50K <- adult$sex2 <- c(rep(0, length(adult$income)))
for (i in 1:length(adult$income)) {
  if(adult$income[i]==">50K.")
    adult$income_g50K[i]<-1
  if(adult$sex[i] == "Male")
    adult$sex2[i] <- 1
  if(adult$race[i] == "White") adult$race_white[i] <- 1
  if(adult$race[i] == "Amer-Indian-Eskimo") adult$race_am.in.esk[i] <- 1
  if(adult$race[i] == "Asian-Pac-Islander") adult$race_as.pac.is[i] <- 1
  if(adult$race[i] == "Black") adult$race_black[i] <- 1
  if(adult$workclass[i] == "Gov") adult$wc_gov[i] <- 1
  if(adult$workclass[i] == "Self") adult$wc_self[i] <- 1
  if(adult$workclass[i] == "Private" ) adult$wc_priv[i] <- 1
  if(adult$marital.status[i] == "Married") adult$ms_marr[i] <- 1
  if(adult$marital.status[i] == "Divorced" ) adult$ms_div[i] <- 1
  if(adult$marital.status[i] == "Separated" ) adult$ms_sep[i] <- 1
  if(adult$marital.status[i] == "Widowed" ) adult$ms_wid[i] <- 1
}
```

MINIMAX TRANSFORM THE CONTINUOUS VARIABLES

Standardization is necessesary to use continuous variables as inputs for neural networks. Here we use mix-max normalization to transform the continuous data into values between 0 and 1. Normalization is necessary because otherwise, the model would favor the larger values when determining weights for each node in the model during back propogation. Non-normalized data would lead to inaccurate predictions on new data.

```
adult$age_mm <- (adult$age - min(adult$age))/(max(adult$age)-min(adult$age))
adult$edu.num_mm <- (adult$education.num - min(adult$education.num))/
  (max(adult$education.num)-min(adult$education.num))
adult$capital.gain_mm <- (adult$capital.gain - min(adult$capital.gain))/
  (max(adult$capital.gain)- min(adult$capital.gain))
adult$capital.loss_mm <- (adult$capital.loss - min(adult$capital.loss))/
  (max(adult$capital.loss)- min(adult$capital.loss))
adult$hours.p.w_mm <- (adult$hours.per.week - min(adult$hours.per.week))/
  (max(adult$hours.per.week)-min(adult$hours.per.week))
newdat <- as.data.frame(adult[,-c(1:15)]) # Get rid of the variables we no longer need
```

RUN THE NEURAL NET

This neural net aims to predict whether or not the income of a new adult is greater than $50k. The weights are the parameters being fit to optimize the model. Unless a random seed is set, the weights are going to be different everyt time we run this code. This is because the fitting/training process may start from a different point eveyr time. This particular network has an input layer, a hidden layer, and an output layer, the hidden layer has 8 nodes. R defaults to using a log-linear activation function within each of these nodes. There is only 1 ouput node.

```
#install.packages("nnet")
library(nnet) # Requires package nnet
net.dat <- nnet(income_g50K ~ ., data = newdat, size = 8)
```

```
## # weights:  153
## initial  value 117.307127
## final  value 113.000000
## converged
```

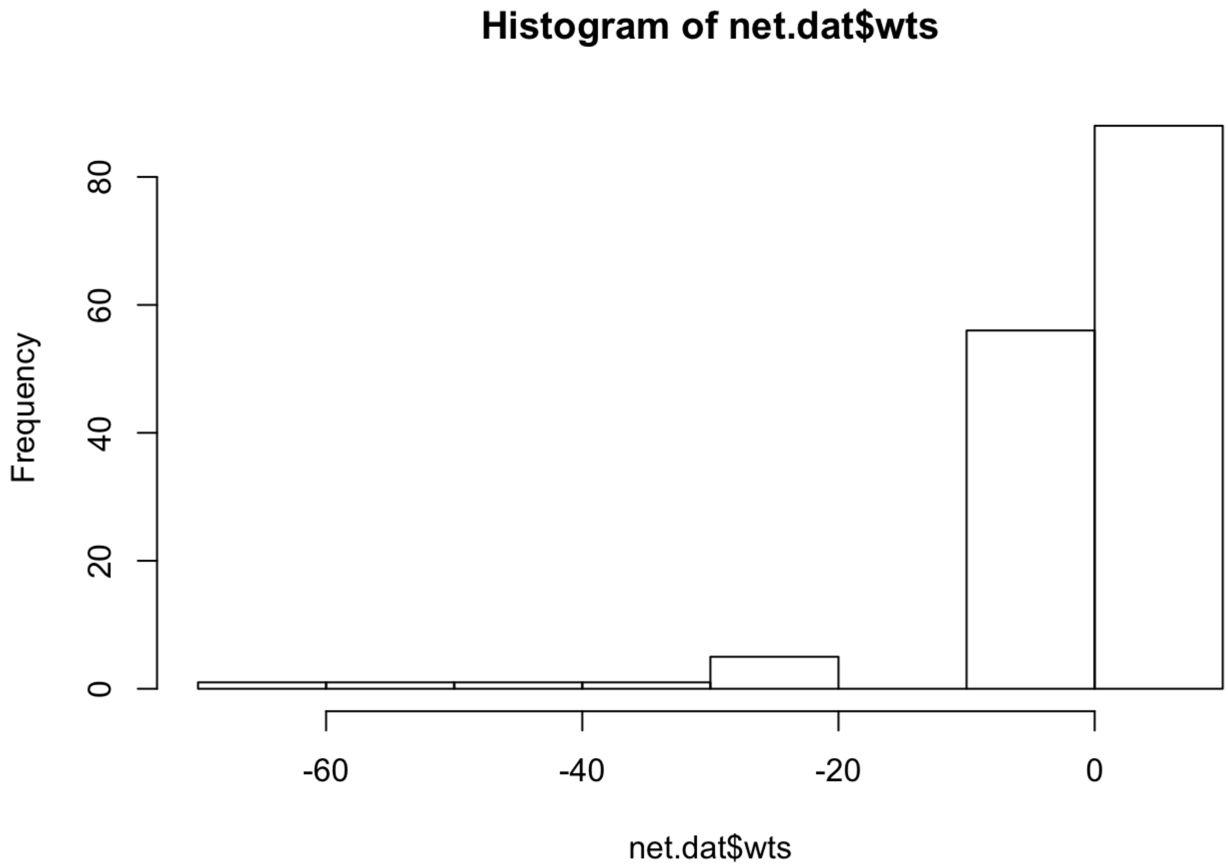
```
table(round(net.dat$fitted.values, 1))   # If fitted values are all the same, rerun nn
et
```

```
##
##   0
## 500
```

```
net.dat$wts # Weights
```

```
##   [1]   5.576328564   2.432287051  -0.124962647   0.511874255   1.234032682
##   [6]  -0.168162014   4.247847436   0.164664884  -0.622892645  -0.108709748
##  [11]   0.559480148   0.791222677   4.697688124   1.513744210   2.419668754
##  [16]  -0.684821672  -0.355686154   1.585437818   0.402640517  -0.358100918
##  [21]   0.344262124   0.645266384   0.137870456   0.198914711  -0.104716595
##  [26]  -0.213687929   0.475873303   0.541152486   0.483000094  -0.042817990
##  [31]   0.887861517   0.117642483   0.725848669  -0.589513689  -0.180137474
##  [36]  -0.531901763   1.615643896   1.069079461  -0.059559753   0.164737619
##  [41]   0.106813010   0.169813903   1.359902485   0.608638690  -0.292671354
##  [46]   0.469167084   0.169944717   0.198801497   1.409595964  -0.003432528
##  [51]   1.119252427   0.363886374  -0.125365884   0.588487362   8.058801557
##  [56]   4.450090649  -0.221877805  -0.386420869   2.193642041   0.684478955
##  [61]   6.747928403  -0.348834457  -0.235372827  -0.046571807   0.045882315
##  [66]   1.464836894   6.189806330   2.084854903   4.277025979  -0.798523927
##  [71]   0.518391735   2.438682354   3.669242152   2.013027938   0.218895290
##  [76]   0.730191525   0.106856959   0.373203413   2.266835489  -0.176012714
##  [81]   0.235867464  -0.559091244   0.057163130   0.970419367   3.316662075
##  [86]   0.636618055   1.582825455   0.298885236   0.040790384   0.944019758
##  [91]  -4.174610810  -2.750051847  -0.037191640  -0.383982427  -0.740375518
##  [96]   0.650695599  -3.160049222   0.215007339  -0.415348699   0.518273783
## [101]  -0.588011076  -0.866284805  -3.839364148  -1.200887786  -1.686377134
## [106]  -0.474154048  -0.388442761  -1.131482098   7.312350915   3.051271757
## [111]   0.959140067   0.831213736   1.516875417  -0.455641597   4.906350113
## [116]   0.101409183  -0.381685252   0.554941701   0.113705907   0.515164693
## [121]   5.366145481   2.224354659   3.366793990  -0.045884878  -0.515437770
## [126]   1.798696240  -1.564806086  -0.364777869   0.085437867   0.027863749
## [131]   0.220727466  -0.368230188  -0.307882643   0.265279824  -0.576061505
## [136]  -0.334143216   0.630061560  -0.727826658  -0.284996902  -0.156747973
## [141]  -0.183122512   0.058983149  -0.634587201  -0.819526936 -60.427356970
## [146] -24.573152462 -27.049381963 -53.153140254 -26.047451180 -36.275086044
## [151] -26.388405165 -21.169952572 -41.494848106
```

```
hist(net.dat$wts)
```



Chapter 15

THE CONFUSION MATRIX

After using the C5.0 package, the confusion matrix is included in the output of summary() # See Chapter 11 for data preparation and code to implement the C5.0 package #### The confusion matrix is a good way visualize the performance of the model in a table. A confusion matrix compares the predicted results to the actual results. This gives us useful information to calculate metrics such as True Positive Rate, False Positive Rate, Accuracy, Precision, and F-Score. These metrics are ways to determine the success of the algorithm depending on the contexts of the problem.
Chapter 11 data prep

```
setwd("~/IMGT680")
adult <- read.csv(file = "adult.txt", stringsAsFactors=TRUE)

#Collapse some of the categories by giving them the same factor label
levels(adult$marital.status);levels(adult$workclass)
```

```
## [1] "Divorced"              "Married-AF-spouse"     "Married-civ-spouse"
## [4] "Married-spouse-absent" "Never-married"         "Separated"
## [7] "Widowed"
```

```
## [1] "?"                "Federal-gov"      "Local-gov"
## [4] "Never-worked"     "Private"          "Self-emp-inc"
## [7] "Self-emp-not-inc" "State-gov"        "Without-pay"
```

```
  levels(adult$marital.status)[2:4]<-"Married"
  levels(adult$workclass)[c(2,3,8)]<- "Gov"
  levels(adult$workclass)[c(5,6)] <- "Self"
  levels(adult$marital.status); levels(adult$workclass)
```

```
## [1] "Divorced"      "Married"       "Never-married" "Separated"
## [5] "Widowed"
```

```
## [1] "?"             "Gov"           "Never-worked"  "Private"
## [5] "Self"          "Without-pay"
```

```
  #add columns with normalized data
  adult$age.z <- (adult$age - mean(adult$age))/sd(adult$age)
  adult$education.num.z <- (adult$education.num - mean(adult$education.num))/sd(adult$ed
ucation.num)
  adult$capital.gain.z<- (adult$capital.gain - mean(adult$capital.gain))/sd(adult$capita
l.gain)
  adult$capital.loss.z <- (adult$capital.loss - mean(adult$capital.loss))/sd(adult$capit
al.loss)
  adult$hours.per.week.z <- (adult$hours.per.week - mean(adult$hours.per.week))/ sd(adul
t$hours.per.week)
```

ADD COSTS TO THE MODEL

The root node pslit is considered to indicate the most important single variable for classifying income. We see that without weights, Capital Gains maximizes the C5.0. informtion gain criterion. The next decision node is on marital status, married or non-marriedn and then normalized capital loss no matter the marital status. After capital loss, the most important variables do not split identically anymore. In predicting income, it is not detrimental to misclassify some observations wrong, so accuracy would be a good model evaluaiton metric. according to our confusion matrix, the accuracy for this model is 86.2.

```
  #install.packages("C50")
  library(C50)
  #After data preparation from Chapter 11
  x <- adult[,c(2,6, 9, 10, 16, 17, 18, 19, 20)]
  y <- adult$income

  # Without weights:
  c50fit <- C5.0(x, y, control = C5.0Control(CF=.1))
  summary(c50fit)
```

```
##
## Call:
## C5.0.default(x = x, y = y, control = C5.0Control(CF = 0.1))
##
##
## C5.0 [Release 2.07 GPL Edition]      Tue Jul 16 13:00:47 2019
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 25000 cases (10 attributes) from undefined.data
##
## Decision tree:
##
## capital.gain.z > 0.7694287: >50K. (1065/18)
## capital.gain.z <= 0.7694287:
## :...marital.status in {Divorced,Never-married,Separated,Widowed}:
##     :...capital.loss.z > 5.282196:
##     :   :...capital.loss.z <= 5.646056: <=50K. (38/12)
##     :   :   capital.loss.z > 5.646056: >50K. (36/7)
##     :   capital.loss.z <= 5.282196:
##     :   :...capital.gain.z > 0.4757047:
##     :       :...capital.gain.z <= 0.494004: >50K. (19)
##     :       :   capital.gain.z > 0.494004: <=50K. (67/6)
##     :       capital.gain.z <= 0.4757047:
##     :       :...education.num.z <= 0.7503064: <=50K. (10336/211)
##     :           education.num.z > 0.7503064:
##     :           :...education.num.z <= 1.532462: <=50K. (2319/280)
##     :               education.num.z > 1.532462:
##     :               :...age.z <= -0.4826879: <=50K. (57/7)
##     :                   age.z > -0.4826879:
##     :                   :...age.z <= 1.124586: >50K. (92/34)
##     :                       age.z > 1.124586: <=50K. (22/3)
##     marital.status = Married:
##     :...capital.loss.z > 4.175664:
##         :...capital.loss.z <= 4.711484: >50K. (439/10)
##         :   capital.loss.z > 4.711484:
##         :   :...capital.loss.z <= 5.175032: <=50K. (48)
##         :       capital.loss.z > 5.175032:
##         :       :...education.num.z <= 0.7503064: <=50K. (46/18)
##         :           education.num.z > 0.7503064: >50K. (47/1)
##         capital.loss.z <= 4.175664:
##         :...capital.gain.z > 0.5241912: >50K. (85/4)
##             capital.gain.z <= 0.5241912:
##             :...education.num.z <= 0.7503064:
##                 :...capital.gain.z > 0.4444489: <=50K. (47)
##                 :   capital.gain.z <= 0.4444489:
##                 :   :...capital.gain.z > 0.4023739: >50K. (45/8)
##                 :       capital.gain.z <= 0.4023739:
##                 :       :...capital.gain.z > 0.2690694: <=50K. (131)
##                 :           capital.gain.z <= 0.2690694:
##                 :           :...capital.gain.z <= 0.2543766: <=50K. (7342/1991)
##                 :               capital.gain.z > 0.2543766: >50K. (50/2)
```

```
##                   education.num.z > 0.7503064:
##                   :...capital.loss.z > 1.342044: <=50K. (25/5)
##                       capital.loss.z <= 1.342044:
##                       :...capital.gain.z > 0.2690694:
##                           :...capital.gain.z <= 0.4023739: <=50K. (25)
##                           :   capital.gain.z > 0.4023739:
##                           :   :...capital.gain.z <= 0.4444489: >50K. (12/1)
##                           :       capital.gain.z > 0.4444489: <=50K. (17)
##                           capital.gain.z <= 0.2690694:
##                           :...hours.per.week.z <= -0.6835725:
##                               :...sex = Female: >50K. (69/28)
##                               :   sex = Male:
##                               :   :...education.num.z <= 1.532462: <=50K. (140/31)
##                               :       education.num.z > 1.532462: >50K. (31/12)
##                               hours.per.week.z > -0.6835725:
##                               :...age.z > -0.4096299: >50K. (1876/571)
##                                   age.z <= -0.4096299:
##                                   :...age.z <= -0.9940934: <=50K. (50/14)
##                                       age.z > -0.9940934: >50K. (424/188)
##
##
## Evaluation on training data (25000 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##      30 3462(13.8%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##   18132   884    (a): class <=50K.
##    2578  3406    (b): class >50K.
##
##
##  Attribute usage:
##
##  100.00% capital.gain.z
##   95.74% marital.status
##   95.74% capital.loss.z
##   92.81% education.num.z
##   10.36% hours.per.week.z
##   10.08% age.z
##    0.96% sex
##
##
## Time: 0.1 secs
```

When adding weights, we find that all of the variable orders from using no weights stay the same, but race and workclass are now included in the attribute usage. Adding weights actually increase the overall error rate. Thus, our accuracy decreased from the previous model.

```r
# With weights:
costm <- matrix(c(1, 2, 1, 1), byrow = FALSE, 2, 2)
c50cost <- C5.0(x, y, costs = costm, control = C5.0Control(CF=.1))
summary(c50cost)
```

```
##
## Call:
## C5.0.default(x = x, y = y, control = C5.0Control(CF = 0.1), costs = costm)
##
##
## C5.0 [Release 2.07 GPL Edition]      Tue Jul 16 13:00:48 2019
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 25000 cases (10 attributes) from undefined.data
## Read misclassification costs from undefined.costs
##
## Decision tree:
##
## capital.gain.z > 0.7694287:
## :...hours.per.week.z > -0.4396555: >50K. (973/10)
## :   hours.per.week.z <= -0.4396555:
## :   :...age.z <= -0.8479775: <=50K. (6/1)
## :       age.z > -0.8479775: >50K. (86/3)
## capital.gain.z <= 0.7694287:
## :...capital.loss.z > 4.310242:
##     :...marital.status in {Divorced,Never-married,Separated,Widowed}:
##     :   :...capital.loss.z <= 5.282196: <=50K. (102/1)
##     :   :   capital.loss.z > 5.282196:
##     :   :   :...capital.loss.z <= 5.646056: <=50K. (38/12)
##     :   :       capital.loss.z > 5.646056:
##     :   :       :...capital.loss.z <= 7.270963: >50K. (28)
##     :   :           capital.loss.z > 7.270963: <=50K. (8/1)
##     :   marital.status = Married:
##     :   :...capital.loss.z <= 4.711484:
##     :       :...race = Amer-Indian-Eskimo: <=50K. (2/1)
##     :       :   race in {Asian-Pac-Islander,Black,Other,
##     :       :            White}: >50K. (437/9)
##     :       capital.loss.z > 4.711484:
##     :       :...capital.loss.z <= 5.282196:
##     :           :...age.z <= 1.855166: <=50K. (58)
##     :           :   age.z > 1.855166:
##     :           :   :...capital.loss.z <= 5.140141: <=50K. (2)
##     :           :       capital.loss.z > 5.140141: >50K. (5)
##     :           capital.loss.z > 5.282196:
##     :           :...capital.loss.z > 5.803064: <=50K. (7)
##     :               capital.loss.z <= 5.803064:
##     :               :...capital.loss.z > 5.708361: >50K. (46)
##     :                   capital.loss.z <= 5.708361:
##     :                   :...capital.loss.z <= 5.381884: >50K. (6)
##     :                       capital.loss.z > 5.381884: <=50K. (17/7)
##     capital.loss.z <= 4.310242:
##     :...marital.status in {Divorced,Never-married,Separated,Widowed}:
##         :...capital.gain.z <= 0.4757047: <=50K. (12724/558)
##         :   capital.gain.z > 0.4757047:
##         :   :...capital.gain.z <= 0.494004: >50K. (19)
##         :       capital.gain.z > 0.494004: <=50K. (67/6)
```

```
##         marital.status = Married:
##         :...capital.gain.z > 0.5241912:
##             :...capital.gain.z <= 0.7246822: >50K. (81)
##             :   capital.gain.z > 0.7246822: <=50K. (4)
##             capital.gain.z <= 0.5241912:
##             :...education.num.z <= 0.7503064:
##                 :...capital.gain.z > 0.4444489: <=50K. (47)
##                 :   capital.gain.z <= 0.4444489:
##                 :   :...capital.gain.z > 0.4023739:
##                 :       :...hours.per.week.z <= 0.9425408: >50K. (41/5)
##                 :       :   hours.per.week.z > 0.9425408: <=50K. (4/1)
##                 :       capital.gain.z <= 0.4023739:
##                 :       :...capital.gain.z > 0.2690694: <=50K. (131)
##                 :           capital.gain.z <= 0.2690694:
##                 :           :...capital.gain.z <= 0.2543766: <=50K. (7342/1991)
##                 :               capital.gain.z > 0.2543766:
##                 :               :...race in {Amer-Indian-Eskimo,
##                 :                   :        Asian-Pac-Islander,Other,
##                 :                   :        White}: >50K. (49/1)
##                 :                   race = Black: <=50K. (1)
##                 education.num.z > 0.7503064:
##                 :...capital.loss.z > 1.342044: <=50K. (25/5)
##                     capital.loss.z <= 1.342044:
##                     :...capital.gain.z > 0.2690694:
##                         :...capital.gain.z <= 0.4023739: <=50K. (25)
##                         :   capital.gain.z > 0.4023739:
##                         :   :...capital.gain.z <= 0.4444489: >50K. (12/1)
##                         :       capital.gain.z > 0.4444489: <=50K. (17)
##                         capital.gain.z <= 0.2690694:
##                         :...hours.per.week.z <= -0.6835725: <=50K. (240/91)
##                             hours.per.week.z > -0.6835725:
##                             :...age.z <= -0.4096299: <=50K. (474/250)
##                                 age.z > -0.4096299:
##                                 :...workclass in {?,Never-worked,
##                                     :             Without-pay}: >50K. (0)
##                                     workclass = Self:
##                                     :...education.num.z <= 1.532462: <=50K. (279.4/15
3.9)
##                                     :   education.num.z > 1.532462:
##                                     :   :...age.z <= 1.928224: >50K. (80.4/14.2)
##                                     :       age.z > 1.928224: <=50K. (6.4/1)
##                                     workclass in {Gov,Private}:
##                                     :...hours.per.week.z > 0.04817848: >50K. (724.7/1
57)
##                                         hours.per.week.z <= 0.04817848:
##                                         :...education.num.z <= 1.141384: <=50K. (515.
3/319)
##                                             education.num.z > 1.141384:
##                                             :...sex = Female: <=50K. (34.6/19)
##                                                 sex = Male:
##                                                 :...age.z <= -0.2635141: <=50K. (14/
5)
##                                                     age.z > -0.2635141: >50K. (221.2/
48)
```

```
##
##
## Evaluation on training data (25000 cases):
##
##          Decision Tree
##    -----------------------
##    Size      Errors   Cost
##
##      42 3669(14.7%)   0.16   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##   18768   248    (a): class <=50K.
##    3421  2563    (b): class >50K.
##
##
##  Attribute usage:
##
##  100.00% capital.gain.z
##   95.74% marital.status
##   95.74% capital.loss.z
##   41.14% education.num.z
##   14.80% hours.per.week.z
##   10.03% age.z
##    7.36% workclass
##    1.96% race
##    1.09% sex
##
##
## Time: 0.1 secs
```