Income Class Prediction from Census Data

Imputation

Due to the large size of the data set, preProcess() will not be used to impute the missing values, to avoid running out of memory. Each feature's missing value will be imputed one at a time using an appropriate subset of the data set where necessary.

In [1]:
```
library(dplyr)
library(ggplot2)
library(lattice)
library(stringr)
library(gridExtra)
library(caret)
```

...

Imputing Features Missing in 1995 Data

In [2]:
```
X94 <- read.csv('C:/Datasets/censusincomeX94.csv')
X95 <- read.csv('C:/Datasets/censusincomeX95.csv')
y94 <- read.csv('C:/Datasets/censusincomey94.csv')
```

Training decision tree models using 5-fold cross validation:

In [9]:
```
regionmodel <- train(X94[,-1],y94[,2],method='rpart',trControl = trainControl(method
regionmodel
```

```
CART

98279 samples
  390 predictor
    6 classes: ' Abroad', ' Midwest', ' Northeast', ' Not in universe', ' South', '
West'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 78624, 78622, 78624, 78622, 78624
Resampling results across tuning parameters:

  cp          Accuracy   Kappa
  0.05809814  0.8930086  0.6267978
  0.09019832  0.8776441  0.5868241
  0.21967963  0.8677239  0.4639262

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.05809814.
```

In [11]:
```
statemodel <- train(X94[,-1],y94[,3],method='rpart',trControl = trainControl(method=
statemodel
```

```
Warning message in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainIn
fo, :
"There were missing values in resampled performance measures."

CART

98279 samples
  390 predictor
   50 classes: ' Abroad', ' Alabama', ' Alaska', ' Arizona', ' Arkansas', ' Califor
nia', ' Colorado', ' Connecticut', ' Delaware', ' District of Columbia', ' Florid
a', ' Georgia', ' Idaho', ' Illinois', ' Indiana', ' Iowa', ' Kansas', ' Kentucky',
' Louisiana', ' Maine', ' Maryland', ' Massachusetts', ' Michigan', ' Minnesota', '
Mississippi', ' Missouri', ' Montana', ' Nebraska', ' Nevada', ' New Hampshire', '
New Jersey', ' New Mexico', ' New York', ' North Carolina', ' North Dakota', ' Not
in universe', ' Ohio', ' Oklahoma', ' Oregon', ' Pennsylvania', ' South Carolina',
' South Dakota', ' Tennessee', ' Texas', ' Utah', ' Vermont', ' Virginia', ' West V
irginia', ' Wisconsin', ' Wyoming'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 97572, 78762, 78768, 78766, 78760, 78767, ...
Resampling results across tuning parameters:

  cp           Accuracy   Kappa
  0.008519135  0.8692248  0.5366913
  0.009916805  0.8656168  0.5231865
  0.056905158  0.8530201  0.2060923

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.008519135.
```

In [12]:
```
migmsamodel <- train(X94[,-1],y94[,4],method='rpart',trControl = trainControl(method
migmsamodel
```

```
CART

98279 samples
  390 predictor
    9 classes: ' Abroad to MSA', ' Abroad to nonMSA', ' MSA to MSA', ' MSA to nonMS
A', ' Nonmover', ' NonMSA to MSA', ' NonMSA to nonMSA', ' Not identifiable', ' Not
in universe'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 78622, 78624, 78624, 78624, 78622
Resampling results across tuning parameters:

  cp           Accuracy   Kappa
  0.002856976  0.9475982  0.8261007
  0.082735701  0.9359376  0.7864721
  0.616407207  0.8685066  0.3106448

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.002856976.
```

In [13]:
```
migregmodel <- train(X94[,-1],y94[,5],method='rpart',trControl = trainControl(method
migregmodel
```

```
CART

98279 samples
  390 predictor
    8 classes: ' Abroad', ' Different county same state', ' Different division same
region', ' Different region', ' Different state same division', ' Nonmover', ' Not
in universe', ' Same county'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 78621, 78623, 78625, 78622, 78625
Resampling results across tuning parameters:

  cp           Accuracy   Kappa
  0.000247799  0.9392342  0.7991918
  0.082735701  0.9278689  0.7606918
  0.570170836  0.8851824  0.4505382

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.000247799.
```

In [14]:
```
movregmodel <- train(X94[,-1],y94[,6],method='rpart',trControl = trainControl(method
movregmodel
```

```
CART

98279 samples
  390 predictor
    9 classes: ' Abroad', ' Different county same state', ' Different state in Midw
est', ' Different state in Northeast', ' Different state in South', ' Different sta
te in West', ' Nonmover', ' Not in universe', ' Same county'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 78624, 78623, 78624, 78623, 78622
Resampling results across tuning parameters:

  cp           Accuracy   Kappa
  0.000247799  0.9393258  0.7994223
  0.082735701  0.9278686  0.7606894
  0.570170836  0.8851834  0.4505378

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.000247799.
```

In [15]:
```
sunbeltmodel <- train(X94[,-1],y94[,7],method='rpart',trControl = trainControl(metho
sunbeltmodel
```

```
CART

98279 samples
  390 predictor
    3 classes: ' No', ' Not in universe', ' Yes'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 78624, 78622, 78623, 78624, 78623
Resampling results across tuning parameters:

  cp          Accuracy   Kappa
  0.01125095  0.9441081  0.7993937
  0.09019832  0.9322030  0.7625213
  0.54284261  0.8916668  0.4469628

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.01125095.
```

Using decision tree models to impute missing values from 1995 data:

In [16]:
```
y95 <- as.data.frame(predict(regionmodel,X95))
names(y95) <- c('region')
y95$state <-predict(statemodel,X95)
y95$migration.msa <-predict(migmsamodel,X95)
y95$migration.reg <-predict(migregmodel,X95)
y95$move.reg <-predict(movregmodel,X95)
y95$sunbelt <-predict(sunbeltmodel,X95)
```

Export the data as a backup.

In [22]:
```
write.csv(y95,'C:/Datasets/censusincomey95.csv')
```

Summary of imputed data

In [9]:
```
summary(y95)
```

```
              region                     state                 migration.msa
 Not in universe: 1224    Not in universe:98015    Not in universe:98015
 South          :96791
          migration.reg                   move.reg                      sunbelt
 Not in universe: 1224    Not in universe: 1224    No                 :96791
 Same county    :96791    Same county    :96791    Not in universe: 1224
```

Data set after 1995 imputation:

In [14]:
```
df <- read.csv('C:/Datasets/censusincomedf.csv')
df[df$year=='95',c(22:23,27:29,31)] <- y95
head(df,5)
```

X	age	class.worker	industry	occupation	edu	wage.hr	recent.enroll	marital	major.industry	..
1	73	Not in universe	0	0	High school graduate	0	Not in universe	Widowed	Not in universe or children	..
2	58	Self-employed-not incorporated	4	34	Some college but no degree	0	Not in universe	Divorced	Construction	..
3	18	Not in universe	0	0	10th grade	0	High school	Never married	Not in universe or children	..
4	9	Not in universe	0	0	Children	0	Not in universe	Never married	Not in universe or children	..
5	10	Not in universe	0	0	Children	0	Not in universe	Never married	Not in universe or children	..

Imputing Remaining Missing Values

The dataframe above was exported and is loaded below (avoids having to re-run all the code above).

In [2]:
```
df <- read.csv('C:/Datasets/censusincomedfimpute.csv')
```

Convert appropriate features to factors.

In [3]:
```
df$industry <-factor(df$industry)
df$occupation <-factor(df$occupation)
df$self.employed <-factor(df$self.employed)
df$vet.benefit <-factor(df$vet.benefit)
df$year <-factor(df$year)
df <- df[,-c(1,2)]
summary(df)
```

```
      age                                  class.worker       industry    
 Min.   : 0.00    Not in universe                 :97029   0       :97467  
 1st Qu.:16.00    Private                         :72021   33      :17069  
 Median :34.00    Self-employed-not incorporated: 8442   43      : 8283  
 Mean   :34.93    Local government                : 7783   4       : 5984  
 3rd Qu.:50.00    State government                : 4227   42      : 4683  
 Max.   :90.00    Self-employed-incorporated      : 3264   45      : 4482  
                 (Other)                          : 3528   (Other):58326  
   occupation                                   edu           wage.hr      
 0       :97467    High school graduate       :48374   Min.   :   0.00  
 2       : 8756    Children                   :44347   1st Qu.:   0.00  
 26      : 7886    Some college but no degree:27809   Median :   0.00  
 19      : 5412    Bachelors degree(BA AB BS):19859   Mean   :  56.34  
 29      : 5105    7th and 8th grade          : 7976   3rd Qu.:   0.00  
 36      : 4144    10th grade                 : 7539   Max.   :9999.00  
 (Other):67524   (Other)                     :40390                     
                 recent.enroll                                 marital     
  College or university:  5679    Divorced                         :12707  
  High school          :  6853    Married-A F spouse present      :   665  
```

Check which features have missing values.

In [4]:
```
for(i in 1:42){
    if(sum(is.na(df[,i]))!=0){
        print(names(df)[i])
        print(sum(is.na(df[,i])))
    }
}
```

```
[1] "state"
[1] 707
[1] "father.nat"
[1] 6703
[1] "mother.nat"
[1] 6107
[1] "self.nat"
[1] 3389
```

The state model predicted 'Not in universe' for all the missing values for the state factor in 1995, so the NA's in the state column from 1994 will be replaced with 'Not in universe'.

In [5]:
```
df[is.na(df$state),'state'] <- ' Not in universe'
```

First if nationality will be imputed based on nationality of parents/children. If either parents are US nationality, then the child is assumed to be US nationality.

In [6]:
```
df[!is.na(df$father.nat) & (df$father.nat == ' United-States') & is.na(df$self.nat),
```

In [7]:
```
df[!is.na(df$mother.nat) & (df$mother.nat == ' United-States') & is.na(df$self.nat),
```

Otherwise the child will be assumed to have the nationality of the father, or mother if the father's nationality is unknown.

In [8]:
```
df[!is.na(df$father.nat) & is.na(df$self.nat),'self.nat']<-df[!is.na(df$father.nat)
```

In [9]:
```
df[!is.na(df$mother.nat) & is.na(df$self.nat),'self.nat']<-df[!is.na(df$mother.nat)
```

The mother and father are assumed to have the same nationality if one of their nationalities is unknown.

In [10]:
```
df[!is.na(df$mother.nat) & is.na(df$father.nat),'father.nat']<-df[!is.na(df$mother.n
```

In [11]:
```
df[is.na(df$mother.nat) & !is.na(df$father.nat),'mother.nat']<-df[is.na(df$mother.na
```

Otherwise they are assumed to have the same nationality as the child if known.

In [12]:
```
df[!is.na(df$self.nat) & is.na(df$father.nat),'father.nat']<-df[!is.na(df$self.nat)
```

In [13]:
```
df[!is.na(df$self.nat) & is.na(df$mother.nat),'mother.nat']<-df[!is.na(df$self.nat)
```

This leaves the cases where all three nationalities are missing. Theses will be imputed using other relevant factors, which are first dummy coded.

In [28]:
```
dfnat <- df[!is.na(df$self.nat),c('race','hispanic','region','state','citizen','inco
dummynat <- dummyVars(income~.,dfnat,fullRank=TRUE)
Xnat <- predict(dummynat,dfnat)
head(Xnat,5)
```

```
Warning message in model.frame.default(Terms, newdata, na.action = na.action, xlev
= object$lvls):
"variable 'income' is not a factor"
```

	race. Asian or Pacific Islander	race. Black	race. Other	race. White	hispanic. Central or South American	hispanic. Chicano	hispanic. Cuban	hispanic. Do not know	hispanic. Mexican-American	hispanic. Mexican (Mexicano)	...
1	0	0	0	1	0	0	0	0	0	0	...
2	0	0	0	1	0	0	0	0	0	0	...
3	1	0	0	0	0	0	0	0	0	0	...
4	0	0	0	1	0	0	0	0	0	0	...
5	0	0	0	1	0	0	0	0	0	0	...

In [29]:
```
yselfnat <- df[!is.na(df$self.nat),'self.nat']
yfathernat <- df[!is.na(df$self.nat),'father.nat']
ymothernat <- df[!is.na(df$self.nat),'mother.nat']
```

Decision tree models are trained using 5-fold cross validation.

In [31]:
```
selfnatmodel <- train(Xnat,yselfnat,method='rpart',trControl=trainControl(method='cv
selfnatmodel
```

```
CART

193484 samples
    71 predictor
    42 classes: ' Cambodia', ' Canada', ' China', ' Columbia', ' Cuba', ' Dominican
-Republic', ' Ecuador', ' El-Salvador', ' England', ' France', ' Germany', ' Greec
e', ' Guatemala', ' Haiti', ' Holand-Netherlands', ' Honduras', ' Hong Kong', ' Hun
gary', ' India', ' Iran', ' Ireland', ' Italy', ' Jamaica', ' Japan', ' Laos', ' Me
xico', ' Nicaragua', ' Outlying-U S (Guam USVI etc)', ' Panama', ' Peru', ' Philipp
ines', ' Poland', ' Portugal', ' Puerto-Rico', ' Scotland', ' South Korea', ' Taiwa
n', ' Thailand', ' Trinadad&Tobago', ' United-States', ' Vietnam', ' Yugoslavia'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 154784, 154789, 154791, 154785, 154787
Resampling results across tuning parameters:

  cp          Accuracy   Kappa
  0.03502645  0.9423930  0.6984328
  0.03792820  0.9325629  0.6447530
  0.28437163  0.9107422  0.2474609

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.03502645.
```

In [32]:
```
fathernatmodel <- train(Xnat,yfathernat,method='rpart',trControl=trainControl(method
fathernatmodel
```

```
CART

193484 samples
    71 predictor
    42 classes: ' Cambodia', ' Canada', ' China', ' Columbia', ' Cuba', ' Dominican
-Republic', ' Ecuador', ' El-Salvador', ' England', ' France', ' Germany', ' Greec
e', ' Guatemala', ' Haiti', ' Holand-Netherlands', ' Honduras', ' Hong Kong', ' Hun
gary', ' India', ' Iran', ' Ireland', ' Italy', ' Jamaica', ' Japan', ' Laos', ' Me
xico', ' Nicaragua', ' Outlying-U S (Guam USVI etc)', ' Panama', ' Peru', ' Philipp
ines', ' Poland', ' Portugal', ' Puerto-Rico', ' Scotland', ' South Korea', ' Taiwa
n', ' Thailand', ' Trinadad&Tobago', ' United-States', ' Vietnam', ' Yugoslavia'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 154784, 154790, 154787, 154788, 154787
Resampling results across tuning parameters:

  cp          Accuracy   Kappa
  0.02379494  0.8559571  0.3743215
  0.02627358  0.8476979  0.3455345
  0.12087015  0.8312828  0.1556817

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.02379494.
```

In [33]:
```
mothernatmodel <- train(Xnat,ymothernat,method='rpart',trControl=trainControl(method
mothernatmodel
```

```
CART

193484 samples
    71 predictor
    42 classes: ' Cambodia', ' Canada', ' China', ' Columbia', ' Cuba', ' Dominican
-Republic', ' Ecuador', ' El-Salvador', ' England', ' France', ' Germany', ' Greec
e', ' Guatemala', ' Haiti', ' Holand-Netherlands', ' Honduras', ' Hong Kong', ' Hun
gary', ' India', ' Iran', ' Ireland', ' Italy', ' Jamaica', ' Japan', ' Laos', ' Me
xico', ' Nicaragua', ' Outlying-U S (Guam USVI etc)', ' Panama', ' Peru', ' Philipp
ines', ' Poland', ' Portugal', ' Puerto-Rico', ' Scotland', ' South Korea', ' Taiwa
n', ' Thailand', ' Trinadad&Tobago', ' United-States', ' Vietnam', ' Yugoslavia'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 154788, 154784, 154791, 154788, 154785
Resampling results across tuning parameters:

  cp          Accuracy   Kappa
  0.02312502  0.8607120  0.3307438
  0.02381309  0.8548459  0.3345540
  0.13285667  0.8361929  0.1609285

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.02312502.
```

The predictors are dummy coded for the missing values.

In [35]:
```
dfnanat <- df[is.na(df$self.nat),c('race','hispanic','region','state','citizen','inc
Xnanat <- predict(dummynat,dfnanat)
head(Xnanat,5)
```

```
Warning message in model.frame.default(Terms, newdata, na.action = na.action, xlev
= object$lvls):
"variable 'income' is not a factor"
```

	race. Asian or Pacific Islander	race. Black	race. Other	race. White	hispanic. Central or South American	hispanic. Chicano	hispanic. Cuban	hispanic. Do not know	hispanic. Mexican-American	hispanic. Mexican (Mexicano)	...
12	0	1	0	0	0	0	0	0	0	0	...
88	0	0	1	0	0	0	0	0	0	0	...
93	0	0	0	1	0	0	0	0	0	0	...
130	1	0	0	0	0	0	0	0	0	0	...
194	0	0	0	1	0	0	0	0	0	0	...

THe missing values are imputed using the decision tree models.

In [37]:
```
nat <- as.data.frame(predict(selfnatmodel,Xnanat))
names(nat)<-c('self.nat')
nat$father.nat <- predict(fathernatmodel,Xnanat)
nat$mother.nat <- predict(mothernatmodel,Xnanat)
summary(nat)
```

```
      self.nat               father.nat                mother.nat
 Germany :2793   United-States:2802   United-States:2802
 Cuba     :   9   Mexico        :   8   Mexico        :   8
 Mexico   :   8   Cambodia      :   0   Cambodia      :   0
 Cambodia:   0   Canada        :   0   Canada        :   0
 Canada   :   0   China         :   0   China         :   0
 China    :   0   Columbia      :   0   Columbia      :   0
(Other)  :   0   (Other)       :   0   (Other)       :   0
```

In [38]:
```
df[is.na(df$self.nat),'self.nat']<-nat$self.nat
df[is.na(df$father.nat),'father.nat']<-nat$father.nat
df[is.na(df$mother.nat),'mother.nat']<-nat$mother.nat
```

The complete data set with no missing values is exported for later use.

In [42]:
```
write.csv(df,'C:/Datasets/censusincomedfFull.csv')
```

The data is dummy coded and exported for later use.

In [43]:
```
dummy <- dummyVars(income~.,df,fullRank=TRUE)
X <- predict(dummy,df)
dimnames(X)[[2]] <- gsub(' ','',dimnames(X)[[2]])
head(X,5)
```

```
Warning message in model.frame.default(Terms, newdata, na.action = na.action, xlev
= object$lvls):
"variable 'income' is not a factor"
```

	age	class.worker.Localgovernment	class.worker.Neverworked	class.worker.Notinuniverse	class.worker
1	73	0	0	1	
2	58	0	0	0	
3	18	0	0	1	
4	9	0	0	1	
5	10	0	0	1	

In [44]:
```
write.csv(X,'C:/Datasets/censusincomeXFull.csv')
```