################  
 Anso Michel   
Income Project

################

# 1. The purpose of this project is to analyze the dataset to determine which customers characteristic are associated with high income.

# 2. And to predict high-income customers base on their level of education. High income is classified as >50k

set.seed(123)

library(caret)

library(rattle)

library(plyr)

library(psych)

library(rpart)

library(ggplot2)

############

## STEP 1 ##

############

# Upload income-proj data set (csv file)

incomeProj <- read.csv("~/Income.Proj", stringsAsFactors=TRUE)

# Get acquainted with the data set

View(incomeProj)

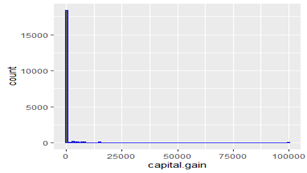
# Delete the variables occupation and X.

incomeProj$X <- incomeProj$occupation <- NULL

# View a histogram of capital.gains

ggplot(incomeProj, aes(capital.gain)) + geom\_histogram(color = "blue",

binwidth=1000)



summary(incomeProj)

# The 99999s are identified as missing numbers. Replace with NA

# Impute the data values for capital.gain = 99999

incomeProj$capital.gain[incomeProj$capital.gain==99999] <- NA

summary(incomeProj$capital.gain)

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

0.0 0.0 0.0 581.6 0.0 41310.0 98

# Now using caret's knn algorithm to impute missing values as capital.gain.imputed

incomeProj.imp.m1 <- preProcess(incomeProj, method = c("knnImpute"))

incomeProj.imp <- predict(incomeProj.imp.m1, incomeProj)

summary(incomeProj.imp)

############

## STEP 2 ##

############

# Create a contingency table to reclassify education variable in two categories with

# columns proportions

# Rename the current education variable

names(incomeProj.imp)[names(incomeProj.imp)=="education"] <- "education.old"

# Reclassify the education categories

incomeProj.imp$educ <-

revalue(incomeProj.imp$education.old, c("11th" = "low",

"10th" = "low", "12th" = "low", "1st-4th" = "low",

"7th-8th" = "low", "5th-6th" = "low", "9th"="low",

"Assoc-acdm"="high","Assoc-voc"="high","HS-grad"="low","Preschool"="low",

"Prof-school"="high","Some-college"="high",

"Bachelors"="high","Doctorate"="high","Masters"="high"))

# Deleting the variable education.old

incomeProj.imp$education.old <- NULL

View(incomeProj.imp)

CT.educ.income.combined <-

table(incomeProj.imp$educ, incomeProj.imp$income)

CT.educ.income.combined.prop <-

round(prop.table(CT.educ.income.combined, margin= 2)\*100, 2)

<=50K >50K

low 51.82 25.20

high 48.18 74.80

# About 75% of individuals in the high income category

# belong to the high education classification.

###########

## STEP 3 ##

###########

# Create a variable as factor that flags anyone who has either capital

# gains or losses into one

incomeProj.imp$capital.GL <- ifelse((incomeProj.imp$capital.gain > 0 |

incomeProj.imp$capital.loss > 0)== TRUE, yes = 1, no = 0)

summary(incomeProj.imp$capital.GL)

incomeProj.imp$capital.GL <- as.factor(incomeProj.imp$capital.GL)

# Create a contingency table of capital.GL against income using

# the new combined categories. Adding column proportions.

(CT.capital.GL.income <-

table(incomeProj.imp$income, incomeProj.imp$capital.GL))

CT.capital.GL.incomeProj.imp.prop <-

round(prop.table(CT.capital.GL.income, margin= 2)\*100, 2)

CT.capital.GL.incomeProj.imp.prop

0 1

<=50K 81.25 42.13

>50K 18.75 57.87

# 58% of customers who have capital gains/losses have high income

##############################

###########

## STEP 4 ##

###########

# Using caret package to partition the data set.

set.seed(123)

inTrain <- createDataPartition(y = incomeProj.imp$income, p = .75,list = FALSE)

# Create training and test sets

incomeProj.imp.training <- incomeProj.imp[ inTrain,]

incomeProj.imp.test <- incomeProj.imp[ -inTrain,]

# The income variable is already

# validated by the createDataPartition.

# Want confirmed

summary(incomeProj.imp.training$income)

<=50K >50K

11430 3571 (11430/3571 = 3.200784)

summary(incomeProj.imp.test$income)

<=50K >50K

3809 1190 (3809/1190 = 3.20084) # stratification is confirmed

###########

## STEP 5 ##

###########

# Validate capital.GL variable

# First, Append a new data variable to each data set

# use a boxplot by group to check the similarity of the distribution

# use the Kruskal\_Wallius test

incomeProj.imp.training$part <- rep("train", nrow(incomeProj.imp.training))

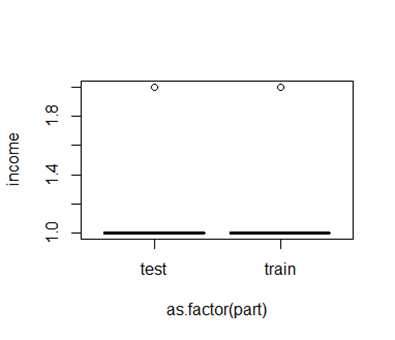
incomeProj.imp.test$part <- rep("test", nrow(incomeProj.imp.test))

# Merging the two data sets using rbind

incomeProj.imp.all.cg <- rbind(incomeProj.imp.training, incomeProj.imp.test)

# different ways to validate the continuous variable

boxplot(income ~ as.factor(part), data = incomeProj.imp.all.cg)



# the distributions are similar

# Performing Kruskal-Wallis test on the merge data set

kruskal.test(capital.GL ~ as.factor(part), data = incomeProj.imp.all.cg)

# p-value is larger than 0.05,

# therefore, the partition is validated with respect to capital.gain variable

Kruskal-Wallis rank sum test

data: capital.GL by as.factor(part)

Kruskal-Wallis chi-squared = 0.86634, df = 1, p-value = 0.352

# p-value is larger than 0.05,

# therefore, the partition is validated with respect to capital.GL variable

###########

## STEP 6 ##

###########

# Validate education

summary(incomeProj.imp.training$educ)

low high

6769 8232

summary(incomeProj.imp.test$educ)

low high

2328 2671

# creating a matrix based on the counts in the simple observed frequencies table

partition.tb1 <- matrix( c(6769, 2328, 8232, 2671), nrow = 2)

# partition table with row and column names

colnames(partition.tb1)<- c("Low","High")

rownames(partition.tb1)<- c("Training","Test")

partition.tb1

Low High

Training 6769 8232

Test 2328 2671

# checking the observed proportions of the training and test sets

round(prop.table(partition.tb1, 1), 4)

Low High

Training 0.4512 0.5488

Test 0.4657 0.5343

# Checking the observed proportions using x² test

chisq.test(partition.tb1, correct = FALSE)

Pearson's Chi-squared test

data: partition.tb1

X-squared = 3.1602, df = 1, p-value = 0.07545

# The partition is validated with p-value larger than 0.05

###########

## STEP 6 ##

###########

# Establish baseline model performance

summary(incomeProj.imp.test$income)

<=50K >50K

3809 1190

3809/(3809+1190) 0.7619

# The All Negative model is our baseline with 76%

###########

## STEP 6 ##

###########

# use caret to standardize incomeProj.imp.training & test data sets and rename them

# incomeProj.imp.training.z and incomeProj.imp.test.z

is.factor(incomeProj.imp.test$income)

[1] TRUE

is.factor(incomeProj.imp.test$educ)

[1] TRUE

preprocess.tr.z <-

preProcess(incomeProj.imp [2:4],

method=c("center", "scale"))

preprocess.tr.z

Created from 20000 samples and 3 variables

Pre-processing:

- centered (2)

- ignored (1)

- scaled (2)

incomeProj.imp.training.z <-

predict(preprocess.tr.z,

incomeProj.imp.training[2:4])

summary(incomeProj.imp.training.z)

###########

# Standardize incomeProj.imp .test data set and rename it incomeProj.imp .test.z

set.seed(123)

incomeProj.imp.test$educ <- as.factor(incomeProj.imp.test$educ)

incomeProj.imp.test$income <- as.factor(incomeProj.imp.test$income)

is.factor(incomeProj.imp.test$income)

is.factor(incomeProj.imp.test$educ)

preprocess.test.z <-

preProcess(incomeProj.imp.test[2:4],

method=c("center", "scale") )

incomeProj.imp.test.z <-

predict(preprocess.test.z,

incomeProj.imp.test[2:4])

summary(incomeProj.imp.test.z)

########### End standardization ###########

###########

## STEP 6 ##

###########

# CART Model

TC <- trainControl(method = "CV", number = 10)

set.seed(123)

fit <- train(income ~ .,

data = incomeProj.imp.training.z,

method = "rpart",

trControl = TC)

# Check for overfitting.

fit$resample

Accuracy Kappa Resample

1 0.8106667 0.3124619 Fold02

2 0.8074617 0.2929309 Fold01

3 0.8126667 0.3223037 Fold03

4 0.8173333 0.3332879 Fold06

5 0.8140000 0.3167659 Fold05

6 0.8080000 0.3115121 Fold04

7 0.8113333 0.3069705 Fold07

8 0.8200000 0.3512921 Fold10

# with about 1% within the folds, there is no evidence of overfitting

# Apply model to the test dataset

testsetpreds <- predict(fit, incomeProj.imp.test.z)

# Get a contingency table of predictions vs actual Approval values from test set.

table(incomeProj.imp.test.z$income, testsetpreds)

testsetpreds

<=50K 50K

<=50K 3757 52

>50K 878 312

The model can predict with an accuracy of 81 % the income level of customers based on their education.

To follow: