Experiment_with_data

Saurabh Gupta

June 12, 2017

Problem statement

In this problem, we have to classify the "Income.Group" based on the predictors. We use rpart classifiers to classify the dataset.

Load the library

```
suppressMessages(library(caret))
suppressMessages(library(doParallel))
suppressMessages(library(Hmisc))
library(data.table)
```

Data loading and exploration

We load the data from given urls. Our goal is to predict the Income.Group for testData and we prepare our model on Vehicle dataset.

```
cl <- makeCluster(detectCores())
registerDoParallel(cl)

#fileUrlTrain <- "https://datahack-prod.s3.ap-south-1.amazonaws.com/workshop_train_file/train_gbW7HTd.c
#download.file(fileUrlTrain,destfile = "./fileTrain.csv")
Vehicle <- read.csv("fileTrain.csv",header=T,na.strings = "")

#fileUrlTest <- "https://datahack-prod.s3.ap-south-1.amazonaws.com/workshop_test_file/test_2AFBew7.csv"
#download.file(fileUrlTest,destfile = "./fileTest.csv")
testData <- read.csv("fileTest.csv",header = T,na.strings = "")
```

Data Cleaning

We convert the Income.Group value (if it is "<=50K" it gives X0 otherwise it gives X1)

```
names(Vehicle) <- gsub("[.]","_",names(Vehicle))
names(testData) <- gsub("[.]","_",names(testData))
Vehicle$Income_Group <- factor(ifelse(Vehicle$Income_Group=="<=50K",0,1))
Vehicle$Income_Group <- make.names(Vehicle$Income_Group)
Vehicle$Income_Group <- factor(Vehicle$Income_Group)
```

Data Visualization

Classes in the response variable is unbalanced. Hence sampling is need to get the good accuracy.

```
str(Vehicle)
```

```
## 'data.frame':    32561 obs. of  12 variables:
##  $ ID            : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Age           : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ Workclass     : Factor w/ 8 levels "Federal-gov",..: 7 6 4 4 4 4 4 6 4 4 ...
##  $ Education     : Factor w/ 16 levels "10th","11th",..: 10 10 12 2 10 13 7 12 13 10 ...
##  $ Marital_Status: Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 5 3 1 3 3 3 4 3 5 3 ...
##  $ Occupation    : Factor w/ 14 levels "Adm-clerical",..: 1 4 6 6 10 4 8 4 10 4 ...
##  $ Relationship  : Factor w/ 6 levels "Husband","Not-in-family",..: 2 1 2 1 6 6 2 1 2 1 ...
##  $ Race          : Factor w/ 5 levels "Amer-Indian-Eskimo",..: 5 5 5 3 3 5 3 5 5 5 ...
##  $ Sex           : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 1 1 2 1 2 ...
##  $ Hours_Per_Week: int  40 13 40 40 40 40 16 45 50 40 ...
##  $ Native_Country: Factor w/ 41 levels "Cambodia","Canada",..: 39 39 39 39 5 39 23 39 39 39 ...
##  $ Income_Group  : Factor w/ 2 levels "X0","X1": 1 1 1 1 1 1 1 2 2 2 ...
```

```
table(Vehicle$Income_Group)
```

```
##
##    X0    X1
## 24720  7841
```

```
prop.table(table(Vehicle$Income_Group))
```

```
##
##        X0        X1
## 0.7591904 0.2408096
```

Missing value detection and treatment

We replaced missing value to the value which occurs frequently in that predictor

```
colSums(is.na(Vehicle))
```

```
##             ID            Age      Workclass      Education Marital_Status
##              0              0           1836              0              0
##     Occupation   Relationship           Race            Sex Hours_Per_Week
##           1843              0              0              0              0
## Native_Country   Income_Group
##            583              0
```

```
Vehicle$Workclass <- impute(Vehicle$Workclass,mode)
Vehicle$Occupation <- impute(Vehicle$Occupation,mode)
Vehicle$Native_Country <- impute(Vehicle$Native_Country,mode)
```

Create train and test data

```
index <- createDataPartition(Vehicle$Income_Group,p=0.7,list=FALSE)
training <- Vehicle[index,]
testing <- Vehicle[-index,]
```

Model Building

```
ctrl <- trainControl(method="repeatedcv",
                     repeats = 10,
```

```
                     classProbs = T,
                     sampling = "up",
                     allowParallel = TRUE,
                     summaryFunction = twoClassSummary
)

rpart_mod <- train(Income_Group ~ .,
                   data=training,
                   method="rpart",
                   trControl=ctrl,
                   tuneLength=30,
                   metric="ROC",
                   na.action = na.omit
)
```

```
## Loading required package: rpart
```

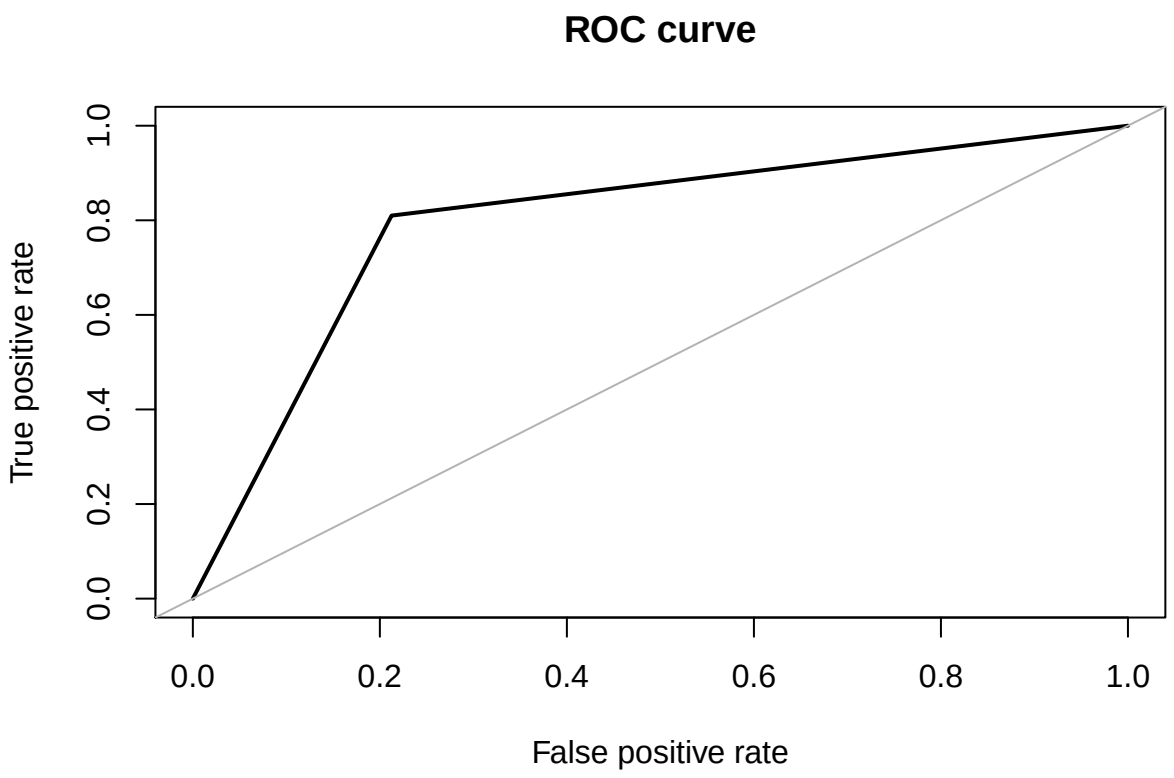
Prediction for training data

```
predicted <- predict(rpart_mod,testing)
caret::confusionMatrix(predicted,testing$Income_Group)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   X0   X1
##         X0 5840  447
##         X1 1576 1905
##
##                Accuracy : 0.7929
##                  95% CI : (0.7847, 0.8009)
##     No Information Rate : 0.7592
##     P-Value [Acc > NIR] : 1.341e-15
##
##                   Kappa : 0.5133
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.7875
##             Specificity : 0.8099
##          Pos Pred Value : 0.9289
##          Neg Pred Value : 0.5473
##              Prevalence : 0.7592
##          Detection Rate : 0.5979
##    Detection Prevalence : 0.6436
##       Balanced Accuracy : 0.7987
##
##        'Positive' Class : X0
##
```

ROC

```
suppressMessages(library(ROSE))
roc.curve(testing$Income_Group,predicted)
```




```
## Area under the curve (AUC): 0.799
```

Prediction for testData

```
# predicted <- ifelse(predicted == "XO","<=50K",">50K")
# df <- data.frame(ID = testData$ID,Income.Group=predicted)
# write.csv(df,file = "final_solutions.csv")
```