Group9 Assignment2 technical document

Alan Chen, Ssu Hsien Lee, Kasturi Pal, Deepanshu Kataria

2023-07-22

Load library

```
library(dplyr)
library(ggplot2)
library(caret)
library(class)
library(rpart)
library(pROC)
library(rpart.plot)
library(e1071)
```

Load the file

```
XYZ =
read.csv('/Users/chenshaokai/Desktop/UMN/Semester1/R/assignment2/grouppart/XY
ZData.csv')
```

Observe data and do the oversampling

```
# remove the userID column that is not helpful for our prediction
XYZ_for_test1 = XYZ[,2:27]

# split our sample data into 70% for cross-validation 30% for the final-
testing
train_rows = createDataPartition(y = XYZ_for_test1$adopter, p = 0.70, list =
FALSE)
XYZData_train = XYZ_for_test1[train_rows,]
XYZData_test = XYZ_for_test1[-train_rows,]

#observing our data, discover it is imbalance and need oversampling for help
table(XYZ_for_test1$adopter)
```

```
##
##     0     1
## 40000  1540
```

```
library(ROSE)
oversample_traindata = ovun.sample(adopter ~ .,
                                   data = XYZData_train,
                                   method = 'over',
                                   N = nrow(XYZData_train)*1.5)$data
```

Do the cross-validation for Decision tree model. Then pre-prune plus post-prune to find one best model.

```
# Seperate our data into five folds for cross-validation test
cv = createFolds(y = oversample_traindata$adopter, k = 5)
auc_all = c()
for (test_rows in cv) {
  XYZcross_train = oversample_traindata[-test_rows,]
  XYZcross_test = oversample_traindata[test_rows,]
  tree_preprun = rpart(adopter ~ ., data = XYZcross_train,
                       method = "class",
                       parms = list(split = "information"),
                       control = rpart.control(cp = 0,
                                               maxdepth = 4,
                                               ))
  pred_tree = predict(tree_preprun, XYZcross_test, type = "prob")
  tree.roc = roc(response = XYZcross_test$adopter,
                 predictor = pred_tree[,2])
  auc_all = c(auc_all,auc(tree.roc))
}
```

Print out the auc for decision tree models of each folds and the mean auc

```
auc_all

## [1] 0.7795042 0.7663095 0.7823970 0.7767950 0.7894357

mean(auc_all)

## [1] 0.7629002
```

Also test on the Naive Bayes model with the same way

```
auc_all = c()
for (test_rows in cv) {
  XYZcross_train = oversample_traindata[-test_rows,]
  nb_XYZcross_train = naiveBayes(adopter ~ ., data = XYZcross_train)
  XYZcross_test = oversample_traindata[test_rows,]
  pred_nb = predict(nb_XYZcross_train, XYZcross_test)
  prob_pred_nb = predict(nb_XYZcross_train, XYZcross_test, type = "raw")
  nb.roc = roc(response = XYZcross_test$adopter,
                 predictor = prob_pred_nb[,2])
  auc_all = c(auc_all,auc(nb.roc))
}
```

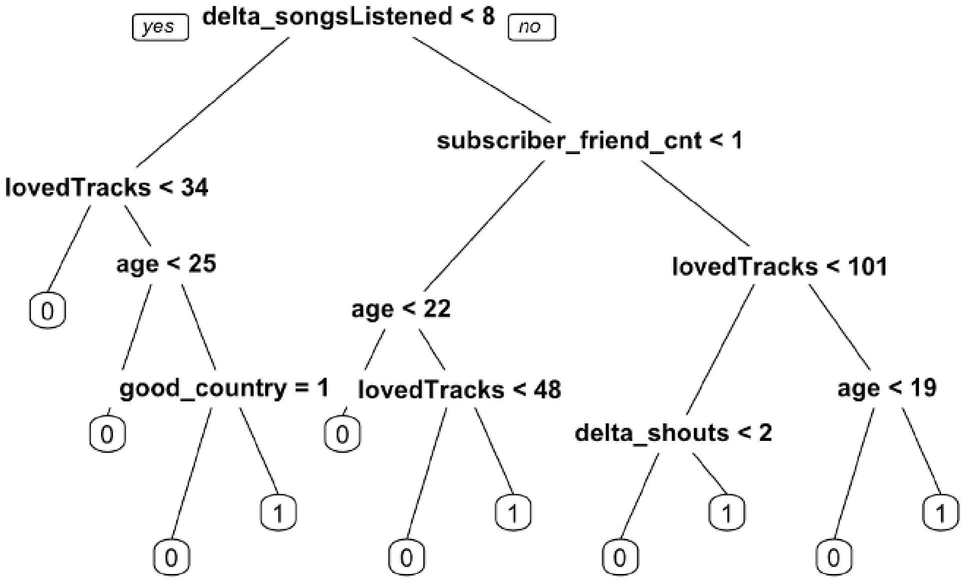
Print out the auc for Naive Bayes models of each folds and the mean auc

```
auc_all

## [1] 0.7351892 0.7471412 0.7373770 0.7495580 0.7455136

mean(auc_all)

## [1] 0.7429558
```

After comparing the AUC between these two models, we decided to choose the Decision Tree model with a higher AUC

Do the final testing with the 30% of our test data

```
tree_final = rpart(adopter ~ ., data = oversample_traindata,
                     method = "class",
                     parms = list(split = "information"),
                     control = rpart.control(cp = 0,
                                             maxdepth = 4,
                     ))
pred_tree_final = predict(tree_final, XYZData_test, type = "prob")
prp(tree_final, varlen = 0)
```



```
tree.roc_final = roc(response = XYZData_test$adopter,
                     predictor = pred_tree_final[,2])

auc_final = tree.roc_final
auc_final
```

```
##
## Call:
## roc.default(response = XYZData_test$adopter, predictor = pred_tree_final[,
2])
##
## Data: pred_tree_final[, 2] 12018 controls (XYZData_test$adopter 0) < 444
cases (XYZData_test$adopter 1).
## Area under the curve: 0.772
```