**Homework: This homework will help you develop understanding and knowledge on importance of data balancing for predictive modeling and applications. Please submit your answers for each question in a word document with your explanations with a supporting R script file you used for this assignment.**

1. **Install and load package ROSE in Rstudio.** The package comes with an inbuilt imbalanced data set named as “hacide”. It comprises of two files: hacide.train and hacide.test. Load these data files to your working environment with the following code.

install.packages("ROSE")  
library(ROSE)

data(hacide)  
str(hacide.train)

As you can see, the training data set contains 3 variables of 1000 observations. “*cls”* is the response binary variable. *x1* and *x2* are dependent variables.

install.packages("ROSE")

library(ROSE)

data(hacide)

str(hacide.train)

plot(hacide.train$cls)

hist(hacide.train$x1)

hist(hacide.train$x2)

table(hacide.train$cls)

sum(hacide.train$cls==1)/length(hacide.train$cls)

Chart, histogram

Description automatically generated**Visualize the distribution of all three variables using the right visualization tool. (1)**

Graphical user interface, chart

Description automatically generated with medium confidence

Chart, histogram

Description automatically generated**What is the imbalance severity in this data set? (1)**

0 = 98% 1 = 02%

1. **Build a decision tree model D1 to predict “cls” with all other variables in the data set. Use the hacide.test data to assess the accuracy, F1 score, precision and recall of this model D1. (2)**

Hint: D1= rpart(cls ~ ., data = hacide.train)

library(rpart)

library(party)

library(partykit)

D1= rpart(cls ~ ., data = hacide.train)

plot(D1)

text(D1)

plot(as.party(D1))

printcp(D1)

plotcp(D1)

D1.pred <- predict(D1, hacide.test, decision.values=TRUE, type="class")

D1.CM=table(D1.pred = D1.pred, true = hacide.test$cls)

D1.CM

sum(diag(D1.CM))/sum(sum(D1.CM))

D1.CM[2, 2]/(D1.CM[2, 2] + D1.CM[1, 2])

D1.CM[2, 2]/(D1.CM[2, 2] + D1.CM[2, 1])

(2\*D1.CM[2, 2])/((2\*D1.CM[2, 2]) + D1.CM[2, 1]+D1.CM[1, 2])

true

D1.pred 0 1

0 245(TN) 4(FP)

1 0(FN) 1(TP)

Accuracy =(TP+TN)/Total = (1+245)/250=0.984

Precision =TP/(TP+FP)= 1/(1 + 4)=0.2

Sensitivity =TP/(TP+FN)= 1/(1+0)=1

F1 =2TP/2TP+FP+FN= (2\*1)/(2\*1)+4+0=.3333

Diagram

Description automatically generated

1. **Use over sampling with ovun.sample() function of the ROSE package to balance the data. What is the total observation number in the new dataset? (2)**

over.hacide <- ovun.sample(cls ~ ., data = hacide.train, method = "over")$data

table(over.hacide$cls)

summary(over.hacide)

0 1

980 986

cls x1 x2

0:980 Min. :-3.7347 Min. :-3.17886

1:986 1st Qu.:-1.4999 1st Qu.:-1.70876

Median :-0.3689 Median :-1.10791

Mean :-0.5081 Mean :-0.86111

3rd Qu.: 0.3581 3rd Qu.:-0.03829

Max. : 1.9886 Max. : 3.03422

1. **Build a decision tree model D2 to predict “cls” again with all other variables in the data set. Use the hacide.test data to assess the accuracy, F1 score, precision and recall of this model D2. (2)**

D2= rpart(cls ~ ., data = over.hacide)

plot(D2)

text(D2)

plot(as.party(D2))

printcp(D2)

plotcp(D2)

D2.pred <- predict(D2, hacide.test, decision.values=TRUE, type="class")

D2.CM=table(D2.pred = D2.pred, true = hacide.test$cls)

D2.CM

sum(diag(D2.CM))/sum(sum(D2.CM))

D2.CM[2, 2]/(D2.CM[2, 2] + D2.CM[1, 2])

D2.CM[2, 2]/(D2.CM[2, 2] + D2.CM[2, 1])

(2\*D2.CM[2, 2])/((2\*D2.CM[2, 2]) + D2.CM[2, 1]+D2.CM[1, 2])

true

D2.pred 0 1

0 243(TN) 2(FP)

1 2(FN) 3(TP)

Accuracy =(TP+TN)/Total = (3+243)/250=0.984

Precision =TP/(TP+FP)= 3/(3 + 2)=0.6

Sensitivity =TP/(TP+FN)= 3/(3 + 2)=0.6

Diagram, schematic

Description automatically generatedF1 =2TP/2TP+FP+FN= (2\*3)/(2\*3)+2+2=0.6

1. **What are the AUC statistics of both models? Plot the ROC curves for both models. (2)**

library(pROC)

par(pty="s")

D1test\_prob = as.numeric(predict(D1, hacide.test, decision.values=TRUE,type = "class"))-1

D1test\_roc = roc(hacide.test$cls ~ D1test\_prob, plot = TRUE, print.auc = TRUE, legacy.axes=TRUE, percent=TRUE, xlab="False Positive Rate", ylab="True Positive Rate", lwd=4)

auc(D1test\_roc)

par(pty="m")

par(pty="s")

D2test\_prob = as.numeric(predict(D2, hacide.test, decision.values=TRUE,type = "class"))-1

D2test\_roc = roc(hacide.test$cls ~ D2test\_prob, plot = TRUE, print.auc = TRUE, legacy.axes=TRUE, percent=TRUE, xlab="False Positive Rate", ylab="True Positive Rate", lwd=4)

auc(D2test\_roc)

Chart

Description automatically generatedpar(pty="m")

Chart, line chart

Description automatically generated

D2 ROC Curve

AUC = 79.59%

D1 ROC Curve

AUC = 60%