# Finance & Risk Analytics

# Assignment

# on

# Default Prediction

# Submitted By :

# Abhishek Ranjan

Adarsh Shrivastava

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

################ Financial & Risk Analytics ####################

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

setwd("C:/Users/Abhishek Ranjan/Desktop/FRA Assignment")

# Reading Training Dataset and basic exploration

data <- read.csv(file = "training.csv", header = TRUE)

summary(data)

str(data)

# Formatting the data

# Converting Casenum & SeriousDlqin2yrs to Factor

data$Casenum <- as.factor(data$Casenum)

data$SeriousDlqin2yrs <- as.factor(data$SeriousDlqin2yrs)

write.csv(file = "impute\_training.csv", data)

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

########################## Data Cleaning & Preparation ##########################

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

# Imputing NA's in NumberofDependents with average

require(plyr)

require(caret)

require(Hmisc)

imputed\_value <- ceiling(mean(data$NumberOfDependents, na.rm = T))

imputed\_value <- as.integer(imputed\_value)

data$NumberOfDependents[is.na(data$NumberOfDependents)] <- imputed\_value

# Removing the column CaseNum before Smote since it will not be included in the model

data <- data[, !(colnames(data) %in% c("Casenum"))]

attach(data)

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

########################### Handling unbalanced data ############################

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

# Smote the data since events(SeriousDlqin2yrs = "1") == 6%

prop.table(table(data$SeriousDlqin2yrs))

require(DMwR)

set.seed(420)

Smoted\_data <- SMOTE(SeriousDlqin2yrs ~. , data, perc.over=200, perc.under=200)

prop.table(table(Smoted\_data$SeriousDlqin2yrs)) # Balanced the data with events == 33%

# Rounding off Over-Sampled data for NumberOfOpenCreditLinesAndLoans && NumberOfDependents

Smoted\_data$NumberOfOpenCreditLinesAndLoans <- round(Smoted\_data$NumberOfOpenCreditLinesAndLoans,digits = 0)

Smoted\_data$NumberOfDependents <- round(Smoted\_data$NumberOfDependents,digits = 0)

Smoted\_data$NumberOfOpenCreditLinesAndLoans <- as.integer(Smoted\_data$NumberOfOpenCreditLinesAndLoans)

Smoted\_data$NumberOfDependents <- as.integer(Smoted\_data$NumberOfDependents)

write.csv(file = "Smoted\_data.csv", Smoted\_data)

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

############ Building the Models for Classification on Training Data ############

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

# Logistic regression Model

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

attach(Smoted\_data)

logit\_model <- glm(SeriousDlqin2yrs ~ RevolvingUtilizationOfUnsecuredLines+NumberOfOpenCreditLinesAndLoans,

data=Smoted\_data, family="binomial")

summary(logit\_model)

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -2.28229 0.14095 -16.192 < 2e-16 \*\*\*

RevolvingUtilizationOfUnsecuredLines 3.13888 0.14623 21.465 < 2e-16 \*\*\*

NumberOfOpenCreditLinesAndLoans 0.04714 0.01008 4.676 2.92e-06 \*\*\*

# Predicting on the Smoted Train dataset

x <- predict(logit\_model, Smoted\_data, type = "response")

Smoted\_data$Prediction <- x

Smoted\_data <- within(Smoted\_data, Prediction[Prediction >=0.5] <- 1)

Smoted\_data <- within(Smoted\_data, Prediction[Prediction <0.5] <- 0)

Smoted\_data$Prediction

confusionMatrix(Smoted\_data$Prediction,Smoted\_data$SeriousDlqin2yrs)

Reference

Prediction 0 1

0 946 283

1 274 632

Accuracy : 0.7391

95% CI : (0.7199, 0.7576)

No Information Rate : 0.5714

P-Value [Acc > NIR] : <2e-16

Kappa : 0.4667

Mcnemar's Test P-Value : 0.7346

Sensitivity : 0.7754

Specificity : 0.6907

Pos Pred Value : 0.7697

Neg Pred Value : 0.6976

Prevalence : 0.5714

Detection Rate : 0.4431

Detection Prevalence : 0.5756

Balanced Accuracy : 0.7331

# ROC Curve to find optimal threshold for the prediction from logit model

library(ROCR)

x <- predict(logit\_model, Smoted\_data, type = "response")

Smoted\_data$Prediction <- x

# calculating the values for ROC curve

pred <- prediction(Smoted\_data$Prediction, Smoted\_data$SeriousDlqin2yrs)

perf <- performance(pred,"tpr","fpr")

# changing params for the ROC plot - width, etc

par(mar=c(5,5,2,2),xaxs = "i",yaxs = "i",cex.axis=1.3,cex.lab=1.4)

# plotting the ROC curve

plot(perf,col="black",lty=3, lwd=3)

# calculating AUC

auc <- performance(pred,"auc")

# now converting S4 class to vector

auc <- unlist(slot(auc, "y.values"))

# adding min and max ROC AUC to the center of the plot

minauc<-min(round(auc, digits = 2))

maxauc<-max(round(auc, digits = 2))

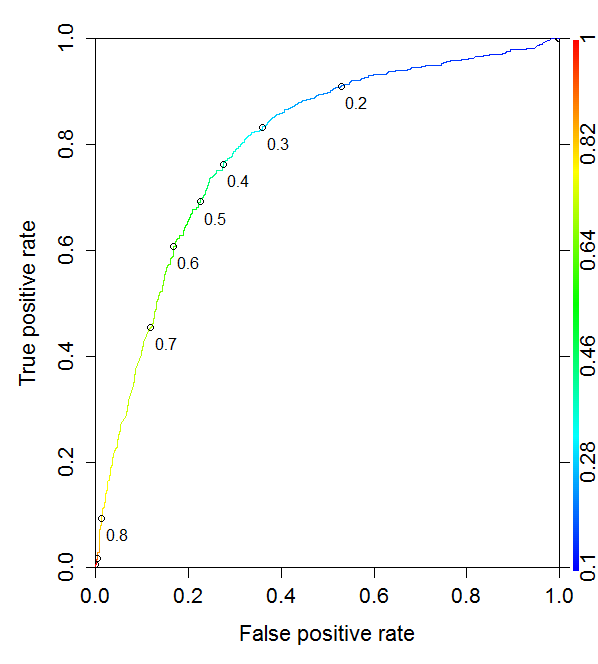
minauct <- paste(c("min(AUC) = "),minauc,sep="")

maxauct <- paste(c("max(AUC) = "),maxauc,sep="")

legend(0.3,0.6,c(minauct,maxauct,"\n"),border="white",cex=1.7,box.col = "white")

str(perf)

plot(perf, colorize=TRUE, print.cutoffs.at=seq(0,1,by=0.1), text.adj=c(-0.2,1.7))



# Checking cutoff alpha value

cutoffs <- data.frame(cut=perf@alpha.values[[1]], fpr=perf@x.values[[1]], tpr=perf@y.values[[1]])

head(cutoffs)

cutoffs <- cutoffs[order(cutoffs$tpr, decreasing=TRUE),]

head(subset(cutoffs, fpr < 0.25))

cut fpr tpr

893 0.3542779 0.3090164 0.7967213

894 0.3536136 0.3098361 0.7967213

891 0.3555140 0.3081967 0.7956284

892 0.3553204 0.3090164 0.7956284

890 0.3564458 0.3081967 0.7945355

888 0.3575794 0.3073770 0.7934426

# Classification Tree

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

library(rpart)

names(Smoted\_data)

attach(Smoted\_data)

#Smoted\_data$SeriousDlqin2yrs <- as.factor(Smoted\_data$SeriousDlqin2yrs)

formula <- SeriousDlqin2yrs ~ RevolvingUtilizationOfUnsecuredLines + DebtRatio +

NumberOfOpenCreditLinesAndLoans + NumberOfDependents

fit <- rpart(formula, method="class")

fit

node), split, n, loss, yval, (yprob)

\* denotes terminal node

1) root 2135 915 0 (0.5714286 0.4285714)

2) RevolvingUtilizationOfUnsecuredLines< 0.4842486 1128 232 0 (0.7943262 0.2056738) \*

3) RevolvingUtilizationOfUnsecuredLines>=0.4842486 1007 324 1 (0.3217478 0.6782522)

6) RevolvingUtilizationOfUnsecuredLines< 0.702259 237 118 1 (0.4978903 0.5021097)

12) DebtRatio< 0.1502466 25 3 0 (0.8800000 0.1200000) \*

13) DebtRatio>=0.1502466 212 96 1 (0.4528302 0.5471698)

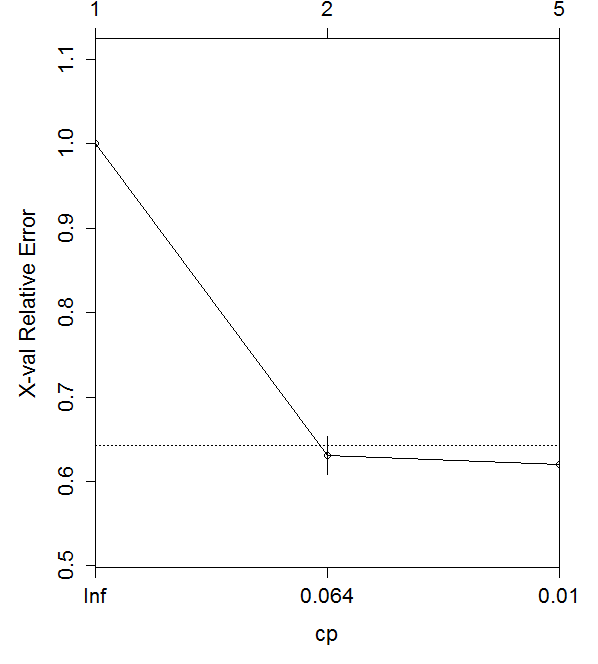
26) RevolvingUtilizationOfUnsecuredLines>=0.6223636 64 27 0 (0.5781250 0.4218750) \*

27) RevolvingUtilizationOfUnsecuredLines< 0.6223636 148 59 1 (0.3986486 0.6013514) \*

7) RevolvingUtilizationOfUnsecuredLines>=0.702259 770 206 1 (0.2675325 0.7324675) \*

printcp(fit)

plotcp(fit)



summary(fit)

plot(fit, uniform=TRUE, main="Classification Tree for Default Modeling")

text(fit, use.n=TRUE, all=TRUE, cex=.8)

# prune the tree

pfit<- prune(fit, cp= fit$cptable[which.min(fit$cptable[,"xerror"]),"CP"])

# plot the pruned tree

plot(pfit, uniform=TRUE,

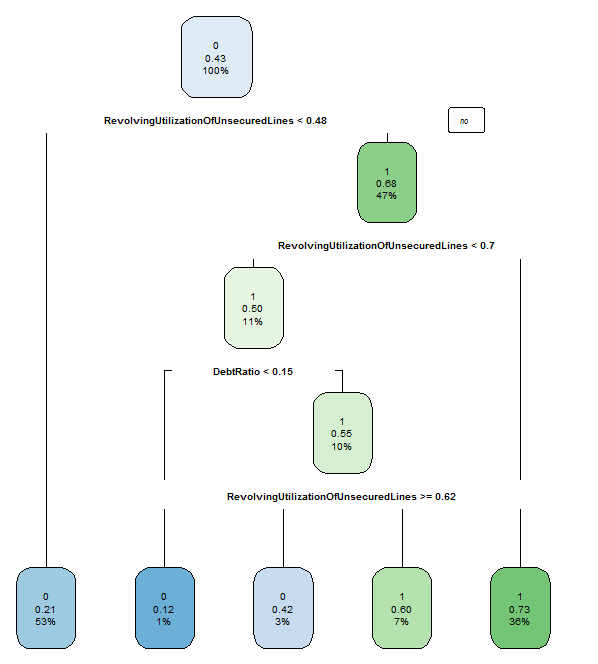
main="Pruned Classification Tree for Default Modeling")

text(pfit, use.n=TRUE, all=TRUE, cex=.8)

#install.packages("rpart.plot")

library(rpart.plot)

rpart.plot(pfit)



# Confusion Matrix of prediction on Smoted Train Dataset

x <- predict(pfit, Smoted\_data, type = "class")

Smoted\_data$Prediction <- x

head(Smoted\_data)

names(Smoted\_data)

attach(Smoted\_data)

confusionMatrix(Smoted\_data$Prediction,Smoted\_data$SeriousDlqin2yrs)

Reference

Prediction 0 1

0 955 262

1 265 653

Accuracy : 0.7532

95% CI : (0.7343, 0.7713)

No Information Rate : 0.5714

P-Value [Acc > NIR] : <2e-16

Kappa : 0.4962

Mcnemar's Test P-Value : 0.9306

Sensitivity : 0.7828

Specificity : 0.7137

Pos Pred Value : 0.7847

Neg Pred Value : 0.7113

Prevalence : 0.5714

Detection Rate : 0.4473

Detection Prevalence : 0.5700

Balanced Accuracy : 0.7482

'Positive' Class : 0

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

################### Evaluating model performance on Test data ###################

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

# Reading the Test dataset

test <- read.csv(file = "test.csv", header = TRUE)

### Predicting logit model on Test Data Set ###

x <- predict(logit\_model, test, type = "response")

test$Prediction <- x

test <- within(test, Prediction[Prediction >=0.35] <- 1)

test <- within(test, Prediction[Prediction <0.35] <- 0)

test$Prediction

confusionMatrix(test$Prediction,test$SeriousDlqin2yrs)

Reference

Prediction 0 1

0 687 21

1 250 42

Accuracy : 0.729

95% CI : (0.7003, 0.7563)

No Information Rate : 0.937

P-Value [Acc > NIR] : 1

Kappa : 0.1484

Mcnemar's Test P-Value : <2e-16

Sensitivity : 0.7332

Specificity : 0.6667

Pos Pred Value : 0.9703

Neg Pred Value : 0.1438

Prevalence : 0.9370

Detection Rate : 0.6870

Detection Prevalence : 0.7080

Balanced Accuracy : 0.6999

'Positive' Class : 0

### Predicting Classification Tree on Test Data Set ###

test <- read.csv(file = "test.csv", header = TRUE)

str(test)

attach(test)

test$SeriousDlqin2yrs <- as.factor(test$SeriousDlqin2yrs)

x <- predict(pfit, newdata = test, type = "class")

test$Prediction <- x

test$Prediction

confusionMatrix(test$Prediction,test$SeriousDlqin2yrs)

Reference

Prediction 0 1

0 712 24

1 225 39

Accuracy : 0.751

95% CI : (0.723, 0.7775)

No Information Rate : 0.937

P-Value [Acc > NIR] : 1

Kappa : 0.1523

Mcnemar's Test P-Value : <2e-16

Sensitivity : 0.7599

Specificity : 0.6190

Pos Pred Value : 0.9674

Neg Pred Value : 0.1477

Prevalence : 0.9370

Detection Rate : 0.7120

Detection Prevalence : 0.7360

Balanced Accuracy : 0.6895

'Positive' Class : 0

**Both the Logit Model and Classification Tree gives out almost the same accuracy of prediction.**