**CLASSIFICATION AND CLUSTERING**

**ARVIND PAWAR**

**DHRUVIN KOTHARI**

**NORTHEASTERN UNIVERSITY**

**ALY6015 INTERMEDIATE ANALYTICS**

**INSTRUCTOR NAME- TENGLONG LI**

**DATE- 03/20/2019**



We are going to use data from a marketing campaign implemented by a major banking institution. The outcome, y, is whether a bank salesperson was able to get a client to sign up for a term deposit (and is labeled 0 for no, and 1 for yes). The objective is to utilize classification algorithms to help the bank management and sales team understand how to maximize clients signing up for a term deposit.

1. **Import the “banking\_data.csv” dataset into your R Studio.**

> banking<-read.csv("C:/Analytics/Intermediate Analytics/Assignment 4/banking\_data.csv")

> str(banking)

'data.frame': 41188 obs. of 16 variables:

$ X : int 1 2 3 4 5 6 7 8 9 10 ...

$ age : int 56 57 37 40 56 45 59 41 24 25 ...

$ marital : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2 2 2 3 3 ...

$ education : int 0 1 1 0 1 0 2 NA 2 1 ...

$ occupation: int 0 0 0 1 0 0 1 0 1 0 ...

$ default : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1 1 ...

$ housing : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3 3 ...

$ contact : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...

$ quarter : int 0 0 0 0 0 0 0 0 0 0 ...

$ day : int 1 1 1 1 1 1 1 1 1 1 ...

$ duration : int 261 149 226 151 307 198 139 217 380 50 ...

$ campaign : int 1 1 1 1 1 1 1 1 1 1 ...

$ pdays : int 999 999 999 999 999 999 999 999 999 999 ...

$ previous : int 0 0 0 0 0 0 0 0 0 0 ...

$ poutcome : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...

$ y

2. To start, let’s calculate the probability that any contacted client signs up for a term deposit. Please fit an intercepts-only logistic regression model to the banking data (recall that an intercept only model can be fit in R as follows: y~1). Using your estimate for the intercept calculate the probability that y=1 using the formula . Confirm that this is correct by constructing a table for the outcome variable, y with no in one column and yes in the other column). Use these values to calculate the probability by hand. Do they match? [Note, a good library for constructing tables is library(gmodels) with the function: CrossTable(y)].

> logistic\_model<-glm(formula = y~1,family = binomial(link="logit"),data=banking)

> summary(logistic\_model)

- Call:

glm(formula = y ~ 1, family = binomial(link = "logit"), data = banking)

Deviance Residuals:

Min 1Q Median 3Q Max

-0.4889 -0.4889 -0.4889 -0.4889 2.0897

Coefficients:

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

(Intercept) -2.06391 0.01558 -132.4 <2e-16 \*\*\*

--

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 28999 on 41187 degrees of freedom

Residual deviance: 28999 on 41187 degrees of freedom

AIC: 29001

Number of Fisher Scoring iterations: 4

> q<-logistic\_model$coefficients

> w<-exp(q)/(1+exp(q))

> CrossTable(banking$y)

Cell Contents

|-------------------------|

| N |

| N / Table Total |

|-------------------------|

Total Observations in Table: 41188

| no | yes |

|-----------|-----------|

| 36548 | 4640 |

| 0.887 | 0.113 |

|-----------|-----------|

3. Next, the bank marketing team would like to know whether their campaign was more successful among lower vs. higher educated clients. Construct a logistic regression model to answer this question. [Remember to use factor(education) in your model so that R treats this as a categorical variable]. What is the Odds Ratio for the highest education group (education=2) compared to the lowest education group (education=0)? How would you interpret this in plain words to the marketing team? Is this a significant association? What is the probability that the lowest education group (education=0) signed up for a term deposit (y=1) in response to this campaign? What is the probability that the highest education group (education=2) signed up for a term deposit (y=1) in response to this campaign?

> logistic\_model2 <- glm(y ~factor(education==0), family=binomial(link="logit"), data=banking)

> summary(logistic\_model2)

Call:

glm(formula = y ~ factor(education == 0), family = binomial(link = "logit"),

data = banking)

Deviance Residuals:

Min 1Q Median 3Q Max

-0.5110 -0.5110 -0.5110 -0.4272 2.2088

Coefficients:

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

(Intercept) -1.96981 0.01859 -105.9 <2e-16 \*\*\*

factor(education == 0)TRUE -0.37821 0.03672 -10.3 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 27548 on 39456 degrees of freedom

Residual deviance: 27437 on 39455 degrees of freedom

(1731 observations deleted due to missingness)

AIC: 27441

Number of Fisher Scoring iterations: 5

> logistic\_model3<- glm(y ~factor(education==1), family=binomial(link="logit"), data=banking)

> summary(logistic\_model3)

Call:

glm(formula = y ~ factor(education == 1), family = binomial(link = "logit"),

data = banking)

Deviance Residuals:

Min 1Q Median 3Q Max

-0.4878 -0.4878 -0.4878 -0.4789 2.1082

Coefficients:

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

(Intercept) -2.06896 0.01831 -112.972 <2e-16 \*\*\*

factor(education == 1)TRUE -0.03869 0.03772 -1.026 0.305

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 27548 on 39456 degrees of freedom

Residual deviance: 27547 on 39455 degrees of freedom

(1731 observations deleted due to missingness)

AIC: 27551

Number of Fisher Scoring iterations: 4

**4. Next, the sales team would like to know which day of the week is best to contact clients. Which day of the week yielded the highest probability of client term deposit sign ups?**

> #part 4

> best\_day<-glm(y~factor(day),family=binomial(link = "logit"),data=banking)

> summary(best\_day)

Call:

glm(formula = y ~ factor(day), family = binomial(link = "logit"),

data = banking)

Deviance Residuals:

Min 1Q Median 3Q Max

-0.5083 -0.5007 -0.4981 -0.4578 2.1484

Coefficients:

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

(Intercept) -2.20298 0.03621 -60.847 < 2e-16 \*\*\*

factor(day)2 0.18955 0.05000 3.791 0.000150 \*\*\*

factor(day)3 0.17864 0.05004 3.570 0.000357 \*\*\*

factor(day)4 0.22175 0.04899 4.527 5.99e-06 \*\*\*

factor(day)5 0.09255 0.05134 1.803 0.071444 .

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 28999 on 41187 degrees of freedom

Residual deviance: 28972 on 41183 degrees of freedom

AIC: 28982

Number of Fisher Scoring iterations: 4

> CrossTable(banking$day,banking$y)

Cell Contents

|-------------------------|

| N |

| Chi-square contribution |

| N / Row Total |

| N / Col Total |

| N / Table Total |

|-------------------------|

Total Observations in Table: 41188

| banking$y

banking$day | no | yes | Row Total |

-------------|-----------|-----------|-----------|

1 | 7667 | 847 | 8514 |

| 1.664 | 13.111 | |

| 0.901 | 0.099 | 0.207 |

| 0.210 | 0.183 | |

| 0.186 | 0.021 | |

-------------|-----------|-----------|-----------|

2 | 7137 | 953 | 8090 |

| 0.241 | 1.901 | |

| 0.882 | 0.118 | 0.196 |

| 0.195 | 0.205 | |

| 0.173 | 0.023 | |

-------------|-----------|-----------|-----------|

3 | 7185 | 949 | 8134 |

| 0.148 | 1.165 | |

| 0.883 | 0.117 | 0.197 |

| 0.197 | 0.205 | |

| 0.174 | 0.023 | |

-------------|-----------|-----------|-----------|

4 | 7578 | 1045 | 8623 |

| 0.708 | 5.574 | |

| 0.879 | 0.121 | 0.209 |

| 0.207 | 0.225 | |

| 0.184 | 0.025 | |

-------------|-----------|-----------|-----------|

5 | 6981 | 846 | 7827 |

| 0.184 | 1.449 | |

| 0.892 | 0.108 | 0.190 |

| 0.191 | 0.182 | |

| 0.169 | 0.021 | |

-------------|-----------|-----------|-----------|

Column Total | 36548 | 4640 | 41188 |

| 0.887 | 0.113 | |

-------------|-----------|-----------|-----------|

> prob1<-exp(coef(best\_day)[1])

> prob1

(Intercept)

0.1104735

> prob2<-exp(coef(best\_day)[1]) + exp(coef(best\_day)[2])

> prob2

(Intercept)

1.319175

> prob3<-exp(coef(best\_day)[1]) + exp(coef(best\_day)[3])

> prob3

(Intercept)

1.306061

> prob4<-exp(coef(best\_day)[1]) + exp(coef(best\_day)[4])

> prob4

(Intercept)

1.35873

> prob5<-exp(coef(best\_day)[1]) + exp(coef(best\_day)[5])

> prob5

(Intercept)

1.207444

> #probability for each day

> prob1/(1+prob1)

(Intercept)

0.09948321

> prob2/(1+prob2)

(Intercept)

0.5688123

> prob3/(1+prob3)

(Intercept)

0.5663602

> prob4/(1+prob4)

(Intercept)

0.576043

> prob5/(1+prob5)

(Intercept)

0.5469873

From the above we can observe that Day 4 that is Thursday has the highest probability where as Day 1 that is Monday has the lowest probability.

5. Lastly, the IT team would like to build a program that prompts sales personnel to up their game when speaking to a client with a high probability of signing up. But first, they need you to build a predictive model. First, split the data into a training dataset and a test dataset, with 80% of observations randomly going to the training data and 20% randomly going to the test data. Then, using any or all of the data at your disposal please fit a logistic regression model with y as the outcome and the training data for the dataset. Next, use the predict function to get predicted values of the outcome from the test dataset (simulating future data). What percent of cases did you get correct (i.e., what was the prediction accuracy of your model)? Use a cut-off of 0.5 for translating your predicted probabilities into values of “yes” and “no”.

* 1. Can we improve our prediction accuracy by utilizing the optimalCutoff() function? How about by using a classification tree model instead of a logistic regression?

**Solution:**

> set.seed(123)

> row.number <- sample(x=1:nrow(banking), size=0.8\*nrow(banking))

> train = banking[row.number,]

> test = banking[-row.number,]

> head(train)

X age marital education occupation default housing contact

11845 11845 33 single 2 1 no unknown telephone

32468 32468 28 married 1 0 no yes cellular

16845 16845 48 married 2 1 no yes telephone

36368 36368 24 single 1 2 no no cellular

38733 38733 45 single 2 1 no unknown cellular

1877 1877 33 divorced 2 1 no no telephone

quarter day duration campaign pdays previous poutcome y

11845 1 5 38 5 999 0 nonexistent no

32468 0 5 237 4 999 0 nonexistent no

16845 1 4 154 1 999 0 nonexistent no

36368 1 2 253 1 999 0 nonexistent no

38733 2 2 133 3 999 1 failure no

1877 0 5 279 1 999 0 nonexistent no

>

> model<-glm(y~.,data=train, family="binomial")

> step(model,direction = "backward")

Start: AIC=13837.12

y ~ X + age + marital + education + occupation + default + housing +

contact + quarter + day + duration + campaign + pdays + previous +

poutcome

Df Deviance AIC

- housing 2 13796 13834

- day 1 13796 13836

- quarter 1 13796 13836

- previous 1 13797 13837

<none> 13795 13837

- education 1 13803 13843

- marital 3 13808 13844

- age 1 13806 13846

- contact 1 13817 13857

- pdays 1 13825 13865

- default 2 13833 13871

- campaign 1 13832 13872

- poutcome 2 13837 13875

- occupation 1 13845 13885

- X 1 15190 15230

- duration 1 18066 18106

Step: AIC=13834.03

y ~ X + age + marital + education + occupation + default + contact +

quarter + day + duration + campaign + pdays + previous +

poutcome

Df Deviance AIC

- day 1 13797 13833

- quarter 1 13797 13833

- previous 1 13798 13834

<none> 13796 13834

- education 1 13804 13840

- marital 3 13809 13841

- age 1 13807 13843

- contact 1 13818 13854

- pdays 1 13826 13862

- default 2 13834 13868

- campaign 1 13833 13869

- poutcome 2 13838 13872

- occupation 1 13846 13882

- X 1 15190 15226

- duration 1 18069 18105

Step: AIC=13832.61

y ~ X + age + marital + education + occupation + default + contact +

quarter + duration + campaign + pdays + previous + poutcome

Df Deviance AIC

- quarter 1 13798 13832

- previous 1 13798 13832

<none> 13797 13833

- education 1 13804 13838

- marital 3 13809 13839

- age 1 13808 13842

- contact 1 13819 13853

- pdays 1 13826 13860

- default 2 13835 13867

- campaign 1 13833 13867

- poutcome 2 13838 13870

- occupation 1 13847 13881

- X 1 15192 15226

- duration 1 18077 18111

Step: AIC=13831.87

y ~ X + age + marital + education + occupation + default + contact +

duration + campaign + pdays + previous + poutcome

Df Deviance AIC

- previous 1 13800 13832

<none> 13798 13832

- education 1 13806 13838

- marital 3 13810 13838

- age 1 13810 13842

- contact 1 13820 13852

- pdays 1 13828 13860

- default 2 13837 13867

- campaign 1 13835 13867

- poutcome 2 13840 13870

- occupation 1 13850 13882

- X 1 15194 15226

- duration 1 18077 18109

Step: AIC=13831.56

y ~ X + age + marital + education + occupation + default + contact +

duration + campaign + pdays + poutcome

Df Deviance AIC

<none> 13800 13832

- education 1 13808 13838

- marital 3 13812 13838

- age 1 13811 13841

- contact 1 13822 13852

- campaign 1 13836 13866

- default 2 13838 13866

- pdays 1 13840 13870

- occupation 1 13852 13882

- poutcome 2 13872 13900

- X 1 15208 15238

- duration 1 18080 18110

Call: glm(formula = y ~ X + age + marital + education + occupation +

default + contact + duration + campaign + pdays + poutcome,

family = "binomial", data = train)

Coefficients:

(Intercept) X age

-5.9365086 0.0001033 0.0076837

maritalmarried maritalsingle maritalunknown

0.0086800 0.1941717 -0.4412506

education occupation defaultunknown

0.0831892 0.2965930 -0.4678314

defaultyes contacttelephone duration

-8.0656900 0.3071242 0.0045741

campaign pdays poutcomenonexistent

-0.0776053 -0.0015116 0.5390568

poutcomesuccess

0.7400797

Degrees of Freedom: 31385 Total (i.e. Null); 31370 Residual

(1564 observations deleted due to missingness)

Null Deviance: 22000

Residual Deviance: 13800 AIC: 13830

>

> model2<-glm(y ~ X + age + marital + education + occupation +

+ +default + contact + duration + campaign + pdays + poutcome,family = "binomial", data = train)

> model2

Call: glm(formula = y ~ X + age + marital + education + occupation +

+default + contact + duration + campaign + pdays + poutcome,

family = "binomial", data = train)

Coefficients:

(Intercept) X age

-5.9365086 0.0001033 0.0076837

maritalmarried maritalsingle maritalunknown

0.0086800 0.1941717 -0.4412506

education occupation defaultunknown

0.0831892 0.2965930 -0.4678314

defaultyes contacttelephone duration

-8.0656900 0.3071242 0.0045741

campaign pdays poutcomenonexistent

-0.0776053 -0.0015116 0.5390568

poutcomesuccess

0.7400797

Degrees of Freedom: 31385 Total (i.e. Null); 31370 Residual

(1564 observations deleted due to missingness)

Null Deviance: 22000

Residual Deviance: 13800 AIC: 13830

>

> #predicted values of outcome from test dataset

> predicted\_value<-predict(model2,newdata = test,type = "response")

> head(predicted\_value,1)

20

0.004229497

>

> test$y\_pred\_num <- ifelse(predicted\_value> 0.5, 1, 0)

> str(test)

'data.frame': 8238 obs. of 17 variables:

$ X : int 20 25 51 54 55 59 66 67 75 85 ...

$ age : int 39 37 54 53 55 55 37 44 37 38 ...

$ marital : Factor w/ 4 levels "divorced","married",..: 3 2 2 3 2 2 2 3 2 3 ...

$ education : int 0 1 1 2 0 2 2 0 2 2 ...

$ occupation: int 1 1 1 1 0 2 1 0 1 1 ...

$ default : Factor w/ 3 levels "no","unknown",..: 2 1 1 1 2 2 1 1 2 1 ...

$ housing : Factor w/ 3 levels "no","unknown",..: 1 3 1 1 1 3 1 3 3 1 ...

$ contact : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...

$ quarter : int 0 0 0 0 0 0 0 0 0 0 ...

$ day : int 1 1 1 1 1 1 1 1 1 1 ...

$ duration : int 195 172 160 179 269 145 232 91 214 20 ...

$ campaign : int 1 1 1 1 2 1 1 1 1 1 ...

$ pdays : int 999 999 999 999 999 999 999 999 999 999 ...

$ previous : int 0 0 0 0 0 0 0 0 0 0 ...

$ poutcome : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...

$ y : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...

$ y\_pred\_num: num 0 0 0 0 0 0 0 0 0 0 ...

> test$y\_predicted<-factor(test$y\_pred\_num, levels=c(0, 1))

> str(test)

'data.frame': 8238 obs. of 18 variables:

$ X : int 20 25 51 54 55 59 66 67 75 85 ...

$ age : int 39 37 54 53 55 55 37 44 37 38 ...

$ marital : Factor w/ 4 levels "divorced","married",..: 3 2 2 3 2 2 2 3 2 3 ...

$ education : int 0 1 1 2 0 2 2 0 2 2 ...

$ occupation : int 1 1 1 1 0 2 1 0 1 1 ...

$ default : Factor w/ 3 levels "no","unknown",..: 2 1 1 1 2 2 1 1 2 1 ...

$ housing : Factor w/ 3 levels "no","unknown",..: 1 3 1 1 1 3 1 3 3 1 ...

$ contact : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...

$ quarter : int 0 0 0 0 0 0 0 0 0 0 ...

$ day : int 1 1 1 1 1 1 1 1 1 1 ...

$ duration : int 195 172 160 179 269 145 232 91 214 20 ...

$ campaign : int 1 1 1 1 2 1 1 1 1 1 ...

$ pdays : int 999 999 999 999 999 999 999 999 999 999 ...

$ previous : int 0 0 0 0 0 0 0 0 0 0 ...

$ poutcome : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...

$ y : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...

$ y\_pred\_num : num 0 0 0 0 0 0 0 0 0 0 ...

$ y\_predicted: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...

> test$y1<-factor(test$y\_predicted,labels=c("no","yes"),levels=c(0,1))

> sensitivity(as.integer(test$y), as.integer(test$y1), threshold = 0.5)

[1] 0.9566344

> specificity(as.integer(test$y), as.integer(test$y1), threshold = 0.5)

[1] 0

>

> #Creating a confusion matrix

> Confusion\_Matrix<-table(test$y,test$y\_predicted)

> Confusion\_Matrix

0 1

no 6841 174

yes 559 298

**6841 people did not sign up for the bonds and our model also predicted that, however 559 people didn’t sign up, but our model predicts that they signed up, 174 people signed up for the bonds and our model predicted that they didn’t sign up.**

>

> #calclating Classification rate

> sum(diag(Confusion\_Matrix))/sum(Confusion\_Matrix)

[1] 0.9068852

> #calculating Missclassification rate

> 1-sum(diag(Confusion\_Matrix))/sum(Confusion\_Matrix)

[1] 0.09311484

> table(test$y)

no yes

7333 905

> 7333/8238

[1] 0.8901432

> y\_observed <- test$y

> mean(test$y1==test$y)

> 0.9068852

**6.** Can we improve our prediction accuracy by utilizing the optimalCutoff() function? How about by using a classification tree model instead of a logistic regression?

Solution:

> optcf <- optimalCutoff(test$y,y\_pred\_num)[1]

> ypred21 <- Ifelse(y\_pred\_num > optcf,1,0)

> ypred22 <- factor(ypred21,levels=c(0,1))

> yobs <- test$y

> mean(ypred == yobs)

> 0.9134420

**From the above result of optimalcutoff() function we can say that our prediction efficiency has been improved.**

> install.packages("rpart")

> library(rpart)

> qmodel<-rpart(y~.,method = "class", train1)

> qmodel

n= 32950

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

\* denotes terminal node

1) root 32950 3684 0 (0.88819423 0.11180577)

2) X< 36224.5 28998 1936 0 (0.93323677 0.06676323)

4) duration< 606.5 26592 821 0 (0.96912605 0.03087395) \*

5) duration>=606.5 2406 1115 0 (0.53657523 0.46342477)

10) duration< 834.5 1273 464 0 (0.63550668 0.36449332) \*

11) duration>=834.5 1133 482 1 (0.42541924 0.57458076) \*

3) X>=36224.5 3952 1748 0 (0.55769231 0.44230769)

6) duration< 165.5 1413 229 0 (0.83793347 0.16206653) \*

7) duration>=165.5 2539 1020 1 (0.40173297 0.59826703)

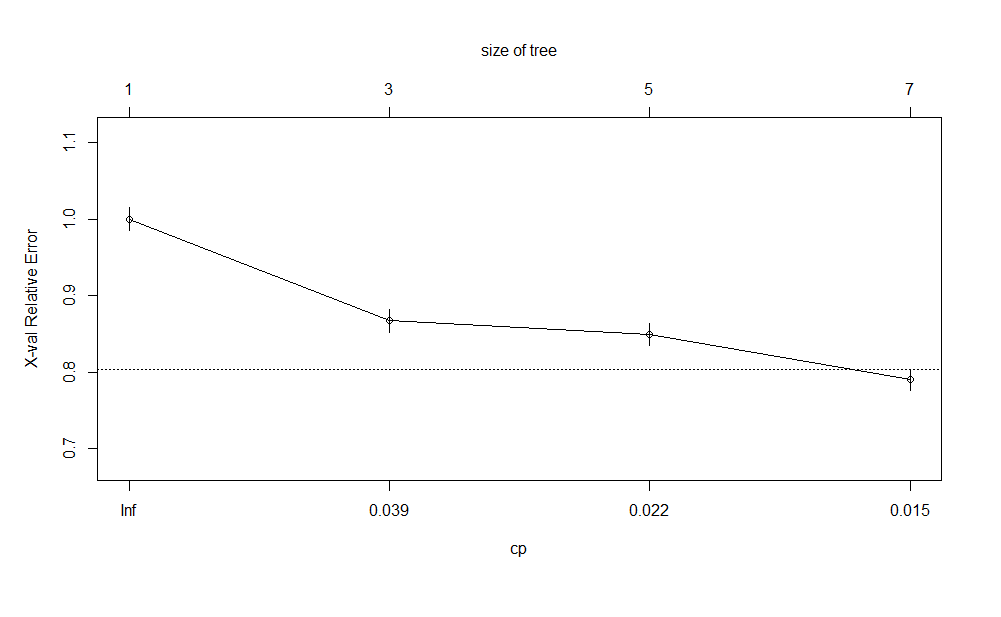
14) pdays>=513 1774 880 1 (0.49605411 0.50394589)

28) duration< 250.5 631 236 0 (0.62599049 0.37400951) \*

29) duration>=250.5 1143 485 1 (0.42432196 0.57567804) \*

15) pdays< 513 765 140 1 (0.18300654 0.81699346) \*

> printcp(qmodel)



Classification tree:

rpart(formula = y ~ ., data = train1, method = "class")

Variables actually used in tree construction:

[1] duration pdays X

Root node error: 3684/32950 = 0.11181

n= 32950

CP nsplit rel error xerror xstd

1 0.067725 0 1.00000 1.00000 0.015527

2 0.022937 2 0.86455 0.86699 0.014578

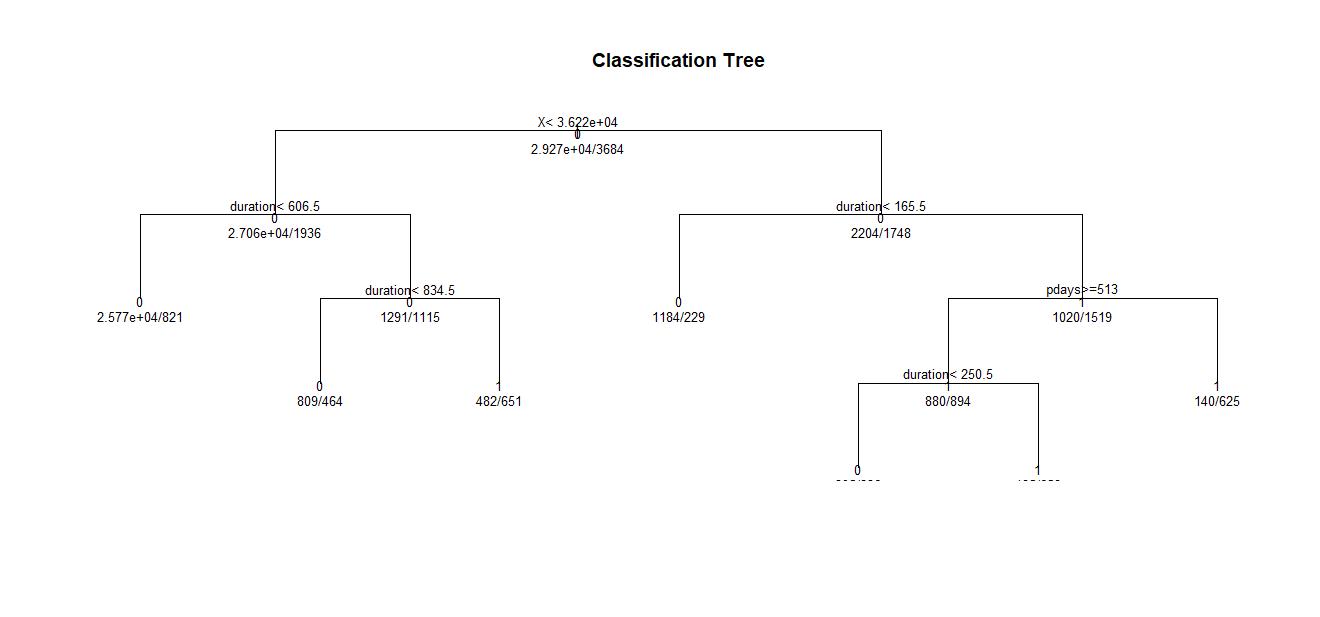
3 0.021580 4 0.81868 0.84881 0.014441

4 0.010000 6 0.77552 0.78990 0.013981

> plotcp(qmodel)

> plot(qmodel, uniform = TRUE, main="Classification Tree")

> text(qmodel,use.n = TRUE, all = TRUE, cex=0.8)

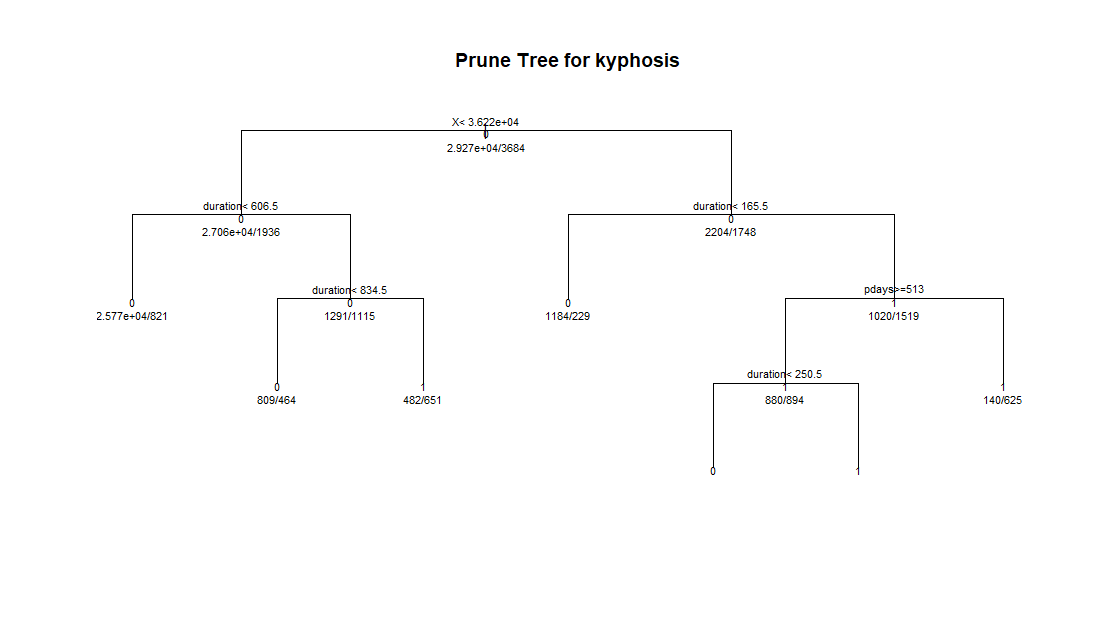


#prune the tree

> prune\_t<-prune(qmodel,cp=qmodel$cptable[which.min(qmodel$cptable[,"xerror"]),"CP"])

> plot(prune\_t,uniform = TRUE, main="Prune Tree for kyphosis")

> text(prune\_t,use.n = TRUE, all=TRUE,cex=0.9)



**From the above outcomes, we think that the classification tree would be better than logistic regression as the efficiency increases by using classification and it also spreads the data into a tree like structure which is easy to understand and can know the probabilities at each node.**

**References:**

* What is Logistic Regression? - Statistics Solutions. (2019). Statistics Solutions. Retrieved 20 March 2019, from <https://www.statisticssolutions.com/what-is-logistic-regression/>
* Quick-R: Tree-Based Models. (2019). Statmethods.net. Retrieved 20 March 2019, from <https://www.statmethods.net/advstats/cart.html>
* Decision Trees in R. (2019). DataCamp Community. Retrieved 20 March 2019, from <https://www.datacamp.com/community/tutorials/decision-trees-R>
* prune.rpart function | R Documentation. (2019). Rdocumentation.org. Retrieved 20 March 2019, from <https://www.rdocumentation.org/packages/rpart/versions/4.1-13/topics/prune.rpart>