Untitled

Viswanth

2 January 2018

## Project  
#To Build a Logistic Regression Model to predict default10yr a categorical variable in creditset using the predictors age,income and loan using the creditset.csv  
##Solution:  
  
#In this project We need to predict default10yr using the other predictor variables given in the dataset using Logstic Regression as our #response variable is a categorical variable .We will train our model and then test our model by splitting the crediset into 2 datasets .  
  
# Importing the dataset

# creditset<-read.csv(file.choose()) OR  
#we can use this to upload/import our creditset csv file from computer too  
creditset<- read.csv("G:/Business Analytics\_R\_Acad glid/creditset.csv")  
dim(creditset)

## [1] 2000 6

names(creditset)

## [1] "clientid" "income" "age" "loan" "LTI"   
## [6] "default10yr"

#We can see that creditset has 2000 observations with 6 variables/columns . We need to predict response Variable "default10yr" which is categorical variable ,so we have to make a Logistic Regression Model

library(caTools)

## Warning: package 'caTools' was built under R version 3.4.3

# Splitting creditset into training and testing DATASET

# Splitting creditset into training and testing DATASET

table(creditset$default10yr)

##   
## 0 1   
## 1717 283

set.seed(2)  
split<-sample.split(creditset,SplitRatio = 0.75)  
split

## [1] TRUE TRUE TRUE TRUE FALSE FALSE

training<-subset(creditset,split=="TRUE")  
dim(training)

## [1] 1334 6

testing<-subset(creditset,split== "FALSE")  
dim(testing)

## [1] 666 6

names(creditset)

## [1] "clientid" "income" "age" "loan" "LTI"   
## [6] "default10yr"

####Checking Actual 0's & 1's of default10yr in creditset,trainingset,testing set . O -> Non-defaulter , 1-> defaulter

table(creditset$default10yr) #tells us how many defaulters & non defualters are there in the creditset

##   
## 0 1   
## 1717 283

table(training$default10yr) #tells us how many defaulters & non defualters are there in the training set

##   
## 0 1   
## 1133 201

table(testing$default10yr) #tells us how many defaulters & non defualters are there in the testing set

##   
## 0 1   
## 584 82

#we need to predict default10yr from the data given.As default10yr is a categorial 1 being defaulter 0 being non -defaulter.We need to use logistic regression for predicting our dependent variable default10yr.In above we have divided our data set into 2 parts training and testing. We are going to build our model /train our model with training dataset and will test the same model on our testing dataset. training & testing are the subsets of creditset.For building our Logistic Regression model clientid and LTI are not the predictors of default10yr we will not use these variables while building our model . Our predictor variables are age,income and loan .We need to decide and optimize our model in such a way that the accuracy is more and the threshold cutoff probability is such that the model is not dangerous like we predict defaulters as non-defaulters as such an info will be dangerous for bank meaning False-Negative should be minimum but not much accuracy of the model is lost

#Building Logistic Regression Model

model<-glm(default10yr~income+age+loan,data=training,family= binomial)   
model

##   
## Call: glm(formula = default10yr ~ income + age + loan, family = binomial,   
## data = training)  
##   
## Coefficients:  
## (Intercept) income age loan   
## 9.8166458 -0.0002367 -0.3465840 0.0017076   
##   
## Degrees of Freedom: 1333 Total (i.e. Null); 1330 Residual  
## Null Deviance: 1131   
## Residual Deviance: 316.7 AIC: 324.7

#Hence our logit function predictors are as below  
  
 # Constant term is 9.8166458  
 #Coefficient for Income is -0.0002367   
 # Coefficient for age is -0.3465840  
 # Coefficient for loan is 0.0017076  
# Our Logit function becomes   
  
 # logit(odds) = logit(default10yr) = 9.8166458 + ((-0.0002367) \* Income) + ((-0.3465840) \* age) + ((0.0017076) \*loan)  
   
 # p(default10yr) = e^(9.8166458 + ((-0.0002367) \* Income) + ((-0.3465840) \* age) + ((0.0017076) \*loan)) / ( 1+ e^(9.8166458 + ((-0.0002367) \* Income) + ((-0.3465840) \* age) + ((0.0017076) \*loan))

#Optimizing Our Model  
# Building others models trying to optimize "model" by trying combinations of predictors  
  
model1<-glm(default10yr~income,data=training,family= binomial)  
summary(model1)

##   
## Call:  
## glm(formula = default10yr ~ income, family = binomial, data = training)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.5958 -0.5813 -0.5674 -0.5523 1.9862   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.897e+00 2.536e-01 -7.480 7.44e-14 \*\*\*  
## income 3.687e-06 5.292e-06 0.697 0.486   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1130.9 on 1333 degrees of freedom  
## Residual deviance: 1130.4 on 1332 degrees of freedom  
## AIC: 1134.4  
##   
## Number of Fisher Scoring iterations: 4

model2<-glm(default10yr~age,data=training,family= binomial)  
summary(model2)

##   
## Call:  
## glm(formula = default10yr ~ age, family = binomial, data = training)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.39150 -0.48762 -0.21609 -0.09672 2.02203   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.06921 0.32098 9.562 <2e-16 \*\*\*  
## age -0.14283 0.01076 -13.278 <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: 1130.91 on 1333 degrees of freedom  
## Residual deviance: 809.99 on 1332 degrees of freedom  
## AIC: 813.99  
##   
## Number of Fisher Scoring iterations: 6

model3<-glm(default10yr~loan,data=training,family= binomial)  
summary(model3)

##   
## Call:  
## glm(formula = default10yr ~ loan, family = binomial, data = training)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6076 -0.5434 -0.3574 -0.2567 2.3937   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.667e+00 1.981e-01 -18.51 <2e-16 \*\*\*  
## loan 3.450e-04 2.748e-05 12.56 <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: 1130.91 on 1333 degrees of freedom  
## Residual deviance: 941.76 on 1332 degrees of freedom  
## AIC: 945.76  
##   
## Number of Fisher Scoring iterations: 5

model4<-glm(default10yr~age+income,data=training,family= binomial)  
summary(model4)

##   
## Call:  
## glm(formula = default10yr ~ age + income, family = binomial,   
## data = training)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.39233 -0.48738 -0.21597 -0.09667 2.02211   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 3.065e+00 4.347e-01 7.050 1.79e-12 \*\*\*  
## age -1.428e-01 1.077e-02 -13.266 < 2e-16 \*\*\*  
## income 8.987e-08 6.139e-06 0.015 0.988   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1130.91 on 1333 degrees of freedom  
## Residual deviance: 809.99 on 1331 degrees of freedom  
## AIC: 815.99  
##   
## Number of Fisher Scoring iterations: 6

model5<-glm(default10yr~age+loan,data=training,family= binomial)  
summary(model5)

##   
## Call:  
## glm(formula = default10yr ~ age + loan, family = binomial, data = training)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.22066 -0.28396 -0.08938 -0.01918 2.60955   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.904e+00 3.987e-01 4.776 1.79e-06 \*\*\*  
## age -2.147e-01 1.654e-02 -12.986 < 2e-16 \*\*\*  
## loan 6.128e-04 4.831e-05 12.686 < 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: 1130.91 on 1333 degrees of freedom  
## Residual deviance: 521.43 on 1331 degrees of freedom  
## AIC: 527.43  
##   
## Number of Fisher Scoring iterations: 7

model6<-glm(default10yr~income+loan,data=training,family= binomial)  
summary(model6)

##   
## Call:  
## glm(formula = default10yr ~ income + loan, family = binomial,   
## data = training)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6818 -0.5102 -0.2789 -0.1034 2.2054   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.429e+00 2.842e-01 -5.028 4.97e-07 \*\*\*  
## income -9.240e-05 1.074e-05 -8.607 < 2e-16 \*\*\*  
## loan 6.568e-04 5.181e-05 12.677 < 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: 1130.91 on 1333 degrees of freedom  
## Residual deviance: 842.77 on 1331 degrees of freedom  
## AIC: 848.77  
##   
## Number of Fisher Scoring iterations: 6

# In all the above models from model1 to model6 the AIC value has not decreased hence we cannot remove any of the predictors from Logistic Regression model named "model" . So we can say all the three predictors age,income & loan are good predictors of default10yr. Our Final model for predicting default10yr is model which is optimized.

#Predicting Probabilities for training dataset

# now we will find the predicted probabilties associated with our model for training dataset  
library(CARS)

## Warning: package 'CARS' was built under R version 3.4.3

res<-predict(model,training,type="response")  
head(res,n=5)

## 1 2 3 4 7   
## 3.909786e-06 2.198782e-02 6.628029e-06 3.250508e-03 2.377538e-01

# Confusion Matrix for our results when compared with predicted values for training dataset  
# we have threshold prob of 0.5 in this case which is taken by default

tabtraining0.5<-table(ActualValue=training$default10yr,PredictedValue=res>=0.5) #checked with probability cutoff = 0.5  
tabtraining0.5

## PredictedValue  
## ActualValue FALSE TRUE  
## 0 1105 28  
## 1 42 159

accuracytraining0.5<-sum(diag(tabtraining0.5)/sum(tabtraining0.5))  
accuracytraining0.5

## [1] 0.9475262

#our model built is accurate to upto 95% meaning it predicted defaulter as defaulter & non-defaulter as non-defaulter in the training dataset from creditset with almost 95% accuracy  
# Predicting Probabilties for testing dataset

# now checking our model for accuracy with testing dataset  
# Finding the predicted probabilties associated with observations using our model for testing dataset  
res1<-predict(model,testing,type="response")  
head(res1)

## 5 6 11 12 17   
## 9.245496e-01 1.160065e-07 1.078035e-03 8.220100e-05 5.608433e-04   
## 18   
## 1.193050e-07

# Confusion Matrix for our results when compared with predicted values for testing dataset  
tabtesting0.5<-table(ActualValue=testing$default10yr,PredictedValue=res1>=0.5) #with probability cutoff= 0.5  
tabtesting0.5

## PredictedValue  
## ActualValue FALSE TRUE  
## 0 562 22  
## 1 14 68

accuracytesting0.5<-sum(diag(tabtesting0.5)/sum(tabtesting0.5))  
accuracytesting0.5 #accuracy at 0.5 cutoff prob

## [1] 0.9459459

#our model built is accurate to upto 95% meaning after verifying ith testing dataset from creditset.As it predicted & we verified defaulter as defaulter & non-defaulter as non-defaulter in the testing dataset from creditset with almost 95% accuracy   
   
# ROC curve

# now we will try to check using ROCR package whether our cutoff Probability is with less False-Negative or not .  
# We can also check using ROC Curve whether our cutoff probablity is most closest to NW corner of the graph or not meaning it has least distance with NW corner of graph .  
  
library(ROCR)

## Warning: package 'ROCR' was built under R version 3.4.3

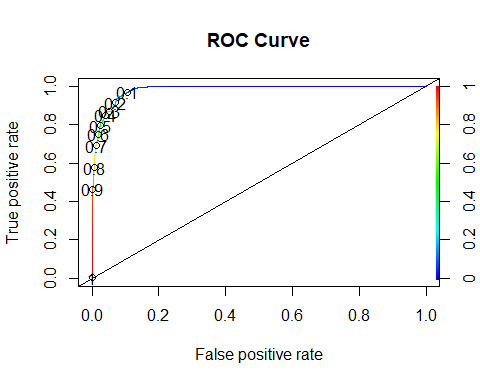
## Loading required package: gplots

## Warning: package 'gplots' was built under R version 3.4.3

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

ROCRPred<-prediction(res,training$default10yr)  
ROCRPref<-performance(ROCRPred,"tpr","fpr")  
plot(ROCRPref,colorize=TRUE,main="ROC Curve",print.cutoffs.at=seq(0.1,by=0.1))  
abline(a=0,b=1)



# We can see that the distance between the 0.4 probabilty and the Northwest corner is less as compared to 0.5  
# so we can decide to take our cutoff probabilty as 0.4  
# this probability will also decrese number of false-Negatives which were dangerous to our model

#### Trying different Probabilty values as cutoff to check diffrent Confusion matrices  
# checking confusion Matrix for diffrent Probabilities for our training dataset  
table(ActualValue=training$default10yr,PredictedValue=res>=0.3) #checked with probability cutoff = 0.3

## PredictedValue  
## ActualValue FALSE TRUE  
## 0 1073 60  
## 1 26 175

table(ActualValue=training$default10yr,PredictedValue=res>=0.4) #checked with probability cutoff = 0.4

## PredictedValue  
## ActualValue FALSE TRUE  
## 0 1088 45  
## 1 30 171

table(ActualValue=training$default10yr,PredictedValue=res>=0.5) #checked with probability cutoff = 0.5

## PredictedValue  
## ActualValue FALSE TRUE  
## 0 1105 28  
## 1 42 159

# so finally As per the model our probability cutoff is 0.4 as it gives me a good accuracy along with less false negatives   
# Thus giving the Confusion matrix as   
  
tabtraining0.4<-table(ActualValue=training$default10yr,PredictedValue=res>=0.4) #checked with probability cutoff = 0.4  
tabtraining0.4

## PredictedValue  
## ActualValue FALSE TRUE  
## 0 1088 45  
## 1 30 171

# checking actual 0's & 1's in training dataset  
table(training$default10yr)

##   
## 0 1   
## 1133 201

accuracytraining0.4<-sum(diag(tabtraining0.4))/sum(tabtraining0.4)  
accuracytraining0.4

## [1] 0.9437781

#Sensitivity & Specificity & Accuracy for training dataset

#calculating Sensitivity & Specificity for dataset training for p=0.4  
sensitivitytraining0.4<-tabtraining0.4[4]/(tabtraining0.4[2]+tabtraining0.4[4])  
sensitivitytraining0.4

## [1] 0.8507463

specificitytraining0.4<-tabtraining0.4[1]/(tabtraining0.4[1] + tabtraining0.4[3])  
specificitytraining0.4

## [1] 0.9602824

# Classification Accuracy for group 0 for training dataset is equal to "specificitytraining0.4" calculated as shown in output  
  
# Classification Accuracy for group 1 is equal to calculated as "sensitivitytraining0.4" shown in output  
  
# Classification Accuracy of our model for training dataset is given by equal to  
classification\_accuracy\_training0.4<- (sensitivitytraining0.4+specificitytraining0.4)/2  
classification\_accuracy\_training0.4

## [1] 0.9055144

#Overall accuracy found by using the table meaning true predictions out of actuals is given by  
# (TruePositive + TrueNegative ) / (TP+TN+FP+FN)  
accuracy\_tabtraining0.4<-sum(diag(tabtraining0.4))/sum(tabtraining0.4)  
accuracy\_tabtraining0.4

## [1] 0.9437781

#Verifying our Model on Testing dataset

# checking actual 0's & 1's in testing dataset  
table(testing$default10yr)

##   
## 0 1   
## 584 82

# We have below Confusion matrix as for testing   
tabtesting0.4<-table(ActualValue=testing$default10yr,PredictedValue=res1>=0.4) #checked with probability cutoff = 0.4  
tabtesting0.4

## PredictedValue  
## ActualValue FALSE TRUE  
## 0 559 25  
## 1 10 72

## Verifying our Model with Probability 0.4 for the testing dataset

# Verifying our Model with Probability 0.4 for the testing dataset  
  
#calculating Sensitivity & Specificity for dataset testing for p=0.4  
sensitivitytesting0.4<-tabtesting0.4[4]/(tabtesting0.4[2]+tabtesting0.4[4])  
sensitivitytesting0.4

## [1] 0.8780488

specificitytesting0.4<-tabtesting0.4[1]/(tabtesting0.4[1] + tabtesting0.4[3])  
specificitytesting0.4

## [1] 0.9571918

# Classification Accuracy for group 0 for testing dataset is equal to "specificitytesting0.4" calculated as shown in output  
  
# Classification Accuracy for group 1 is equal to calculated as "sensitivitytesting0.4" shown in output  
  
# Classification Accuracy of our model for testing dataset is given by equal to  
classification\_accuracy\_testing0.4<- (sensitivitytesting0.4+specificitytesting0.4)/2  
classification\_accuracy\_testing0.4

## [1] 0.9176203

#Overall accuracy found by using the table,meaning true total predictions out of total actuals is given by  
# (TruePositive + TrueNegative ) / (TP+TN+FP+FN)  
accuracy\_tabtesting0.4<-sum(diag(tabtesting0.4))/sum(tabtesting0.4)  
accuracy\_tabtesting0.4

## [1] 0.9474474

#Our Logistic Regression model "model" is a good model which is able to predict the data which was even not supplied to it i.e. testing dataset. As model was trained with training dataset. We get good Overall Accuracy and also model is less risky .

# Calculating Area under ROC curve  
# install.packages("verification")  
library(verification)

## Warning: package 'verification' was built under R version 3.4.3

## Loading required package: fields

## Warning: package 'fields' was built under R version 3.4.3

## Loading required package: spam

## Warning: package 'spam' was built under R version 3.4.3

## Loading required package: dotCall64

## Warning: package 'dotCall64' was built under R version 3.4.3

## Loading required package: grid

## Spam version 2.1-2 (2017-12-21) is loaded.  
## Type 'help( Spam)' or 'demo( spam)' for a short introduction   
## and overview of this package.  
## Help for individual functions is also obtained by adding the  
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##   
## Attaching package: 'spam'

## The following objects are masked from 'package:base':  
##   
## backsolve, forwardsolve

## Loading required package: maps

## Warning: package 'maps' was built under R version 3.4.3

## Loading required package: boot

## Loading required package: CircStats

## Warning: package 'CircStats' was built under R version 3.4.3

## Loading required package: MASS

## Loading required package: dtw

## Warning: package 'dtw' was built under R version 3.4.3

## Loading required package: proxy

## Warning: package 'proxy' was built under R version 3.4.3

##   
## Attaching package: 'proxy'

## The following object is masked from 'package:spam':  
##   
## as.matrix

## The following objects are masked from 'package:stats':  
##   
## as.dist, dist

## The following object is masked from 'package:base':  
##   
## as.matrix

## Loaded dtw v1.18-1. See ?dtw for help, citation("dtw") for use in publication.

roc.area(training$default10yr,res)

## $A  
## [1] 0.9826288  
##   
## $n.total  
## [1] 1334  
##   
## $n.events  
## [1] 201  
##   
## $n.noevents  
## [1] 1133  
##   
## $p.value  
## [1] 5.273768e-106

# Area unde ROC curve is equal to as shown below

# Calculating Nagelkerke R square  
# install.packages("fmsb")  
library(fmsb)  
NagelkerkeR2(model)

## $N  
## [1] 1334  
##   
## $R2  
## [1] 0.7992079

# our Nagelkerke R square is calculated as shownabove .

logLik(model)

## 'log Lik.' -158.3399 (df=4)

# The probability of the observed results given the parameter estimates is known as the Likelihood. Since the likelihood is a small number less than 1, it is customary to use -2 times the log likelihood (-2LL) as an estimate of how well the model fits the data. A good model is one that results in a high likelihood of the observed results. This translates into a small value for -2LL (if a model fits perfectly, the likelihood=1 and -2LL=0)  
  
#This is far from zero, however because there is no upper boundary for -2LL it is difficult to make a statement about the meaning of the score.   
 # It is more often used to see whether adding additional variables to the model leads to a significant reduction in the -2LL.

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

## Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.