Logistic regression

Suresh kumar prajapati

2024-03-15

##log[p(X) / (1-p(X))] = β0 + β1X1 + β2X2 + … + βpXp

##where:

#Xj: The jth predictor variable #βj: The coefficient estimate for the jth predictor variable #The formula on the right side of the equation predicts the log odds of the response variable taking #on a value of 1.

##three types of logistic regression models:

1:-Binary logistic regression:

#Example 1: NBA Draft

#Suppose a sports data scientist wants to use the predictor variables (1) points, (2) rebounds, and (3) assists to predict the probability that a given college basketball player gets drafted into the NBA.

#Since there are only two possible outcomes (drafted or not drafted) for the response variable, the data scientist would use a binomial logistic regression model.

#Example 2: Spam Detection`

##Multinomial logistic regression:

#Example 1: Political Preference Example 2: Sports Preference

##Ordinal logistic regression: #School Ratings Example 2: Movie Ratings

##Step 1: Load the Data

#This dataset contains the following information about 10,000 individuals:

default: Indicates whether or not an individual defaulted.

student: Indicates whether or not an individual is a student.

balance: Average balance carried by an individual. income: Income of the individual.

```
library(ISLR)

## Warning: package 'ISLR' was built under R version 4.3.3
```

```r
data<-(Default)
# view summary of dataset
summary(data)
```

```
##  default    student       balance           income     
##  No :9667   No :7056   Min.   :   0.0   Min.   :  772  
##  Yes: 333   Yes:2944   1st Qu.: 481.7   1st Qu.:21340  
##                        Median : 823.6   Median :34553  
##                        Mean   : 835.4   Mean   :33517  
##                        3rd Qu.:1166.3   3rd Qu.:43808  
##                        Max.   :2654.3   Max.   :73554
```

```r
# find total observation in dataset
nrow(data)
```

```
## [1] 10000
```

#Step 2: Create Training and Test Samples

```r
# make this example reproducible
set.seed(1)
#Use 70% of dataset as training set and remaing 30% asas testing set
sample<-sample(c(TRUE,FALSE),1000,replace = TRUE,prob=c(0.7,0.3))
head(sample)
```

```
## [1]  TRUE  TRUE  TRUE FALSE  TRUE FALSE
```

```r
train<-data[sample,]
head(train)
```

```
##   default student   balance   income
## 1      No      No  729.5265 44361.63
## 2      No     Yes  817.1804 12106.13
## 3      No      No 1073.5492 31767.14
## 5      No      No  785.6559 38463.50
## 8      No     Yes  808.6675 17600.45
## 9      No      No 1161.0579 37468.53
```

```r
test<-data[!sample,]
head(test)
```

```
##    default student   balance    income
## 4       No      No  529.2506 35704.494
## 6       No     Yes  919.5885  7491.559
## 7       No      No  825.5133 24905.227
## 15      No      No 1112.9684 23810.174
## 17      No      No    0.0000 50265.312
## 18      No     Yes  527.5402 17636.540
```

#Step 3: Fit the Logistic Regression Model

#The coefficients in the output indicate the average change in log odds of defaulting. For example, a one unit increase in balance is associated with an average increase of 0.005988 in the log odds of defaulting. #P-value of student status: 0.0843

#P-value of balance: <0.0000

#P-value of income: 0.4304

```
#fit logistic regression model
model <- glm(default~student+balance+income,family="binomial",data=train)
model

##
## Call:  glm(formula = default ~ student + balance + income, family = 
"binomial",
##     data = train)
##
## Coefficients:
## (Intercept)   studentYes      balance       income
##  -1.113e+01   -5.168e-01    5.789e-03    5.305e-06
##
## Degrees of Freedom: 6959 Total (i.e. Null);  6956 Residual
## Null Deviance:       2028
## Residual Deviance: 1075  AIC: 1083

#disable scientific notation for model summary
options(scipen=999)

#view model summary
summary(model)

##
## Call:
## glm(formula = default ~ student + balance + income, family = "binomial",
##     data = train)
##
## Coefficients:
##                 Estimate    Std. Error z value            Pr(>|z|)
## (Intercept) -11.128574419   0.613496157 -18.140 <0.0000000000000002 ***
## studentYes   -0.516806001   0.289432892  -1.786              0.0742 .
## balance       0.005789308   0.000281945  20.533 <0.0000000000000002 ***
## income        0.000005305   0.000010167   0.522              0.6018
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2027.6  on 6959  degrees of freedom
## Residual deviance: 1075.3  on 6956  degrees of freedom
## AIC: 1083.3
```

```
##
## Number of Fisher Scoring iterations: 8
```

#McFadden's R2, which ranges from 0 to just under 1. Values close to #0 indicate that the model has no predictive power. In practice, #values over 0.40 indicate that a model fits the data very well.

#A value of 0.4728807 is quite high for McFadden's R2, which indicates that our model fits the data very well and has high predictive power.

the importance of each predictor variable in the model by using the varImp function from the caret package:

```
library(pscl)

## Warning: package 'pscl' was built under R version 4.3.3

## Classes and Methods for R originally developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University (2002-2015),
## by and under the direction of Simon Jackman.
## hurdle and zeroinfl functions by Achim Zeileis.

R2=pscl::pR2(model)["McFadden"]

## fitting null model for pseudo-r2

R2

##  McFadden
## 0.4696546
```

#The importance of each predictor variable in the model by using the varImp function from the caret package:

#Higher values indicate more importance. These results match up nicely with the p-values from the model. Balance is by far the most important predictor variable, followed by student status and then income.

```
library(caret)

## Warning: package 'caret' was built under R version 4.3.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.3.3

## Loading required package: lattice
```

```
caret::varImp(model)

##                Overall
## studentYes  1.7855814
## balance    20.5334678
## income      0.5217886
```

#VIF values above 5 indicate severe multicollinearity. Since none of the predictor variables in our models have a VIF over 5, we can assume that multicollinearity is not an issue in our model.

```
# calculate VIF value for each predictor variable in our model
car::vif(model)

##  student  balance   income
## 2.877491 1.075029 2.808981
```

##Step 4: Use the Model to Make Predictions ##ogistic regression model, we can then use it to make predictions about whether or not an individual will default based on their student status, balance, and income:

```
#define two individuals
new <- data.frame(balance = 1400, income =2000, student = c("Yes", "No"))
new

##   balance income student
## 1    1400   2000     Yes
## 2    1400   2000      No

#predict probability of defaulting
predict(model, new, type="response")

##          1          2
## 0.02847781 0.04684502
```

#The probability of an individual with a balance of $1,400, an income of $2,000, and a student status of “Yes” has a probability of defaulting of .0273. Conversely, an individual with the same balance and income but with a student status of “No” has a probability of defaulting of 0.0439.

```
#calculate probability of default for each individual in test dataset
#optim
predicted <- predict(model, test, type="response")
head(predicted)

##             4             6             7            15            17
## 0.00037992315 0.00186643817 0.00199065834 0.01036459734 0.00001917452
##            18
## 0.00020389059
```

#Step 5: Model Diagnostics `

```r
#library(InformationValue)

#convert defaults from "Yes" and "No" to 1's and 0's
x=test$default <- ifelse(test$default=="Yes", 1,0)
head(x)

## [1] 0 0 0 0 0 0
```

#Calculate a 95% confidence interval for each odds ratio

```r
#calculate odds ratio for each predictor variable
exp(coef(model))

##   (Intercept)    studentYes       balance        income
## 0.00001468661 0.59642248159 1.00580609850 1.00000530517
```

##The odds ratio for each coefficient represents the average increase in the odds of an individual defaulting, assuming all other predictor variables are held constant.

```r
#calculate odds ratio and 95% confidence interval for each predictor variable

exp(cbind(Odds_Ratio = coef(model), confint(model)))

## Waiting for profiling to be done...

##                Odds_Ratio          2.5 %        97.5 %
## (Intercept) 0.00001468661 0.000004246637 0.00004714205
## studentYes  0.59642248159 0.338695384161 1.05449572875
## balance     1.00580609850 1.005269626135 1.00638248359
## income      1.00000530517 0.999985355335 1.00002524030
```

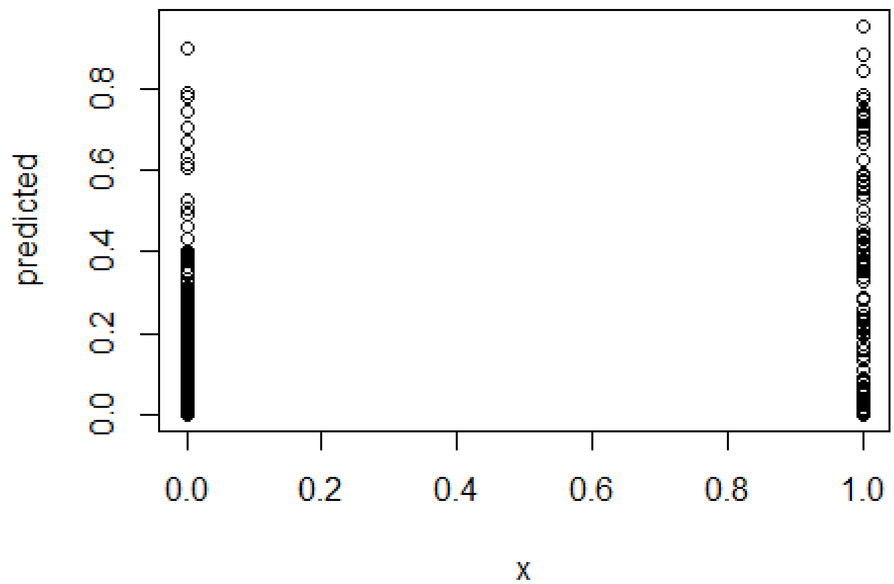
#The predictor variable balance has an odds ratio of 1.0057.

This means for each additional dollar in the balanced carried by an individual, the odds that the individual defaults on their loan increase by a factor of 1.0057, assuming student status and income are held constant.

```r
#plot the ROC curve
library(ROCR)

## Warning: package 'ROCR' was built under R version 4.3.3

plot(x, predicted)
```



predicted
0.0
0.2
0.4
0.6
0.8
0.0
0.2
0.4
0.6
0.8
1.0
x