LogisticRegressionLab.R

andre

2022-10-13

############## Week 11 Lab Logistic Regression #########################  
# We will build a model to predict if a customer will purchase the product (MYDEPV: purchase the product==1) given their age and income and the "Price" that the company may offer (10, 20 or 30 dollars).   
#The company would like to use this information to determine not only a good base price but also whether or not certain groups might be   
# willing to purchase the product if sent a special offer.  
  
  
###################################################  
# Step 1: Set the Working Directory  
###################################################  
#set working directory  
  
###################################################  
# Step 2: Read in and Examine the Data  
###################################################  
myData <- read.csv("survey.csv",header=TRUE, stringsAsFactors = TRUE)  
  
#1. Explore data: Use table() and summary() functions to understand the variables.   
#MyDEPV is the dependent variable, and it means whether or not a customer buy the product.   
  
# table(myData)  
summary(myData)

## MYDEPV Price Income Age   
## Min. :0.000 Min. :10 Min. :17.00 Min. :18.00   
## 1st Qu.:0.000 1st Qu.:10 1st Qu.:29.00 1st Qu.:32.00   
## Median :0.000 Median :20 Median :33.00 Median :32.00   
## Mean :0.432 Mean :20 Mean :42.49 Mean :35.98   
## 3rd Qu.:1.000 3rd Qu.:30 3rd Qu.:55.00 3rd Qu.:43.00   
## Max. :1.000 Max. :30 Max. :99.00 Max. :66.00

# create correlation matrix using Mydata after dropping the first column since it is the dependent variable.  
cor.mat <- cor(myData[,-1])  
cor.mat

## Price Income Age  
## Price 1 0.00000000 0.00000000  
## Income 0 1.00000000 0.09612083  
## Age 0 0.09612083 1.00000000

# transform Price variable as a factor variable.   
## Use relevel Function to re-level the Price factor with value 30 as the base reference.  
myData$Price <-as.factor(myData$Price)  
myData$pricefactor = relevel(as.factor(myData$Price), "30")  
  
##2. split training/test data  
set.seed(2)  
train.index <- sample(c(1:dim(myData)[1]), dim(myData)[1]\*0.6)   
trainData <- myData[train.index, ]  
validationData <- myData[-train.index, ]  
  
###################################################  
#Step 3: Build and Review the Logistic Regression Model  
###################################################  
  
#3 Build a logistic regression model using the train dataset: MYDEPV is the dependent variable.   
  
mylogit <- glm(MYDEPV ~ Income + Age + Price,  
 data = trainData, family = binomial(link="logit"),  
 na.action=na.pass)  
#4 Print out the summary of the mylogit model  
summary(mylogit)

##   
## Call:  
## glm(formula = MYDEPV ~ Income + Age + Price, family = binomial(link = "logit"),   
## data = trainData, na.action = na.pass)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.9104 -0.5958 -0.2825 0.4839 2.9244   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.67088 0.65512 -8.656 < 2e-16 \*\*\*  
## Income 0.11991 0.01107 10.832 < 2e-16 \*\*\*  
## Age 0.03160 0.01433 2.206 0.0274 \*   
## Price20 -0.57631 0.32686 -1.763 0.0779 .   
## Price30 -2.01914 0.38402 -5.258 1.46e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 612.90 on 449 degrees of freedom  
## Residual deviance: 340.16 on 445 degrees of freedom  
## AIC: 350.16  
##   
## Number of Fisher Scoring iterations: 5

# One unit change of Income, the log odds of Purchase increases by 0.12   
# Compared with a Price of 10, Purchase decision at a Price 20 decreases the log odds of Purchase by 0.58  
  
#log odds ratio of purchase = (-6) + 0.12\*Income +0.032\*Age + (-0.58)\*Price20 + (-2)\*Price30   
#logA = ax + by + c  
# A = exp(ax + by +c)  
  
  
# We will now change the reference level at price point 30 then run the logistic model again using the training data.   
# this will change the factor variable base level to 30 to see the effect of this baseline to the model. Compared to the results from a different base level (10),   
#this result shows that customers are more likely to purchase the product.   
mylogit2 = glm(MYDEPV ~ Income + Age + pricefactor ,  
 data = trainData, family = binomial(link="logit"), na.action=na.pass)  
summary(mylogit2)

##   
## Call:  
## glm(formula = MYDEPV ~ Income + Age + pricefactor, family = binomial(link = "logit"),   
## data = trainData, na.action = na.pass)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.9104 -0.5958 -0.2825 0.4839 2.9244   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -7.69002 0.80689 -9.530 < 2e-16 \*\*\*  
## Income 0.11991 0.01107 10.832 < 2e-16 \*\*\*  
## Age 0.03160 0.01433 2.206 0.0274 \*   
## pricefactor10 2.01914 0.38402 5.258 1.46e-07 \*\*\*  
## pricefactor20 1.44282 0.37020 3.897 9.72e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 612.90 on 449 degrees of freedom  
## Residual deviance: 340.16 on 445 degrees of freedom  
## AIC: 350.16  
##   
## Number of Fisher Scoring iterations: 5

###################################################  
#Step 4: variable selection:   
#5: use step function to decide the best variables included in the model mylogit2.  
stepModel <- step(mylogit2)

## Start: AIC=350.16  
## MYDEPV ~ Income + Age + pricefactor  
##   
## Df Deviance AIC  
## <none> 340.16 350.16  
## - Age 1 345.00 353.00  
## - pricefactor 2 374.04 380.04  
## - Income 1 587.72 595.72

#############################################  
#This data includes only four predictor variables, including the new variable pricefactor we just created.   
#Use all the independent variables to build a model mylogit3.  
mylogit3 <- glm(MYDEPV ~ ., data = trainData, family = binomial(link = "logit"), na.action = na.pass)  
  
summary(mylogit3)

##   
## Call:  
## glm(formula = MYDEPV ~ ., family = binomial(link = "logit"),   
## data = trainData, na.action = na.pass)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.9104 -0.5958 -0.2825 0.4839 2.9244   
##   
## Coefficients: (2 not defined because of singularities)  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.67088 0.65512 -8.656 < 2e-16 \*\*\*  
## Price20 -0.57631 0.32686 -1.763 0.0779 .   
## Price30 -2.01914 0.38402 -5.258 1.46e-07 \*\*\*  
## Income 0.11991 0.01107 10.832 < 2e-16 \*\*\*  
## Age 0.03160 0.01433 2.206 0.0274 \*   
## pricefactor10 NA NA NA NA   
## pricefactor20 NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 612.90 on 449 degrees of freedom  
## Residual deviance: 340.16 on 445 degrees of freedom  
## AIC: 350.16  
##   
## Number of Fisher Scoring iterations: 5

#6. run step() function to find the best variables that should be included in the model mylogit3.   
step(mylogit3)

## Start: AIC=350.16  
## MYDEPV ~ Price + Income + Age + pricefactor  
##   
##   
## Step: AIC=350.16  
## MYDEPV ~ Price + Income + Age  
##   
## Df Deviance AIC  
## <none> 340.16 350.16  
## - Age 1 345.00 353.00  
## - Price 2 374.04 380.04  
## - Income 1 587.72 595.72

##   
## Call: glm(formula = MYDEPV ~ Price + Income + Age, family = binomial(link = "logit"),   
## data = trainData, na.action = na.pass)  
##   
## Coefficients:  
## (Intercept) Price20 Price30 Income Age   
## -5.6709 -0.5763 -2.0191 0.1199 0.0316   
##   
## Degrees of Freedom: 449 Total (i.e. Null); 445 Residual  
## Null Deviance: 612.9   
## Residual Deviance: 340.2 AIC: 350.2

###################################################  
#Step 5: Rerun Logistic Regression  
###################################################  
  
#7. Create mylogit4 by building a logistic regression model using the variables you selected as a result of #6  
#Warning: because the variables are too small, variable selection does not influence the model  
#performance in this case. But we do this as a practice.   
  
mylogit4 <- glm(MYDEPV ~ pricefactor + Income + Age, data = trainData, family = binomial(link = "logit"), na.action = na.pass)  
  
summary(mylogit4)

##   
## Call:  
## glm(formula = MYDEPV ~ pricefactor + Income + Age, family = binomial(link = "logit"),   
## data = trainData, na.action = na.pass)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.9104 -0.5958 -0.2825 0.4839 2.9244   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -7.69002 0.80689 -9.530 < 2e-16 \*\*\*  
## pricefactor10 2.01914 0.38402 5.258 1.46e-07 \*\*\*  
## pricefactor20 1.44282 0.37020 3.897 9.72e-05 \*\*\*  
## Income 0.11991 0.01107 10.832 < 2e-16 \*\*\*  
## Age 0.03160 0.01433 2.206 0.0274 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 612.90 on 449 degrees of freedom  
## Residual deviance: 340.16 on 445 degrees of freedom  
## AIC: 350.16  
##   
## Number of Fisher Scoring iterations: 5

###################################################  
#Step 6: Model Evaluation using Confusion Matrix  
###################################################  
##Evaluate the model using confusion matrix. Think about using the right threshold based on the goal of the project  
#load the library caret  
library(caret)

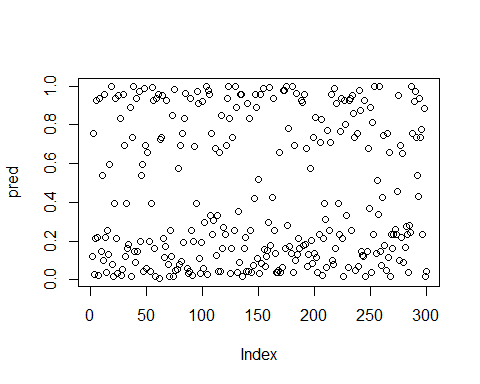
## Warning: package 'caret' was built under R version 4.2.1

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.2.1

## Loading required package: lattice

#8: predict the value for the validation dataset using the predict function and mylogit4 model you just built. You should set the type="response" to have probability value predicted.  
pred <- predict(mylogit4, validationData, type = "response")  
  
#9: build confusion matrix using confusionMatrix() function. Set the threshold as 0.5 (and with different values). What is the accuracy of the model?   
plot(pred)



confusionMatrix(as.factor(ifelse(pred < 0.5, "0", "1")), as.factor(validationData$MYDEPV))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 152 24  
## 1 14 110  
##   
## Accuracy : 0.8733   
## 95% CI : (0.8303, 0.9088)  
## No Information Rate : 0.5533   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.7419   
##   
## Mcnemar's Test P-Value : 0.1443   
##   
## Sensitivity : 0.9157   
## Specificity : 0.8209   
## Pos Pred Value : 0.8636   
## Neg Pred Value : 0.8871   
## Prevalence : 0.5533   
## Detection Rate : 0.5067   
## Detection Prevalence : 0.5867   
## Balanced Accuracy : 0.8683   
##   
## 'Positive' Class : 0   
##

###################################################  
#Step 7: Use Logistic Regression as a Classifier  
###################################################  
#10: Classify new data using the model you just built  
set.seed(2)  
#create a new data   
newdata2 <- data.frame(Age= round(runif(10,min(myData$Age),max(myData$Age))),  
 Income= round(runif(10,min(myData$Income),max(myData$Income))),  
 pricefactor = as.factor(round((runif(10,10,30)/10))\*10))  
  
newdata2$Prob <- predict(mylogit4, newdata2, type = "response")