Binary classifier regression model

Yograj Karki

5/19/2021

Loading libraries

```
library(caTools)
```

Loading the dataset

Here I created csv file by copying the contents from the arff file, and then fixed the attributes of certain columns.

```
setwd("~/MSDS/DSC520/dsc520/Binary_classifier")

data <- read.csv("binary-classifier-data.csv")
str(data)
```

```
## 'data.frame':    1498 obs. of  3 variables:
##  $ label: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ x    : num  70.9 75 73.8 66.4 69.1 ...
##  $ y    : num  83.2 87.9 92.2 81.1 84.5 ...
```

Splitting the data into train and test datasets

```
# Splitting the data into train and test with the 80:20 ratio
dt = sort(sample(nrow(data), nrow(data)*0.8))

# Training dataset
train<-data[dt,]

# Testing dataset
test<-data[-dt,]
```

Fitting the data to a logistic regression model

Here I used the glm() function to fit the model and the method was binomial logistic regression.

```r
# fitting the model
logistic <- glm(label ~ x+y, data = train, family="binomial")

# summary of the model
summary(logistic)
```

```
##
## Call:
## glm(formula = label ~ x + y, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3672  -1.1709  -0.9487   1.1677   1.3814
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.414583   0.131630   3.150 0.001635 **
## x           -0.003410   0.002036  -1.675 0.093941 .
## y           -0.007243   0.002100  -3.449 0.000563 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1659.4  on 1197  degrees of freedom
## Residual deviance: 1641.5  on 1195  degrees of freedom
## AIC: 1647.5
##
## Number of Fisher Scoring iterations: 4
```

Checking the accuracy of the model

Running the test data throught the model

```r
res <- predict(logistic, test, type="response")
res
```

```
##         8        20        21        26        28        33        42        43
## 0.3625456 0.3836847 0.3817814 0.3925229 0.3835418 0.3888461 0.3968957 0.3810827
##        45        46        55        62        70        71        88        90
## 0.3879913 0.3700902 0.3818652 0.4857248 0.4842660 0.4979590 0.4987704 0.4913943
##       101       102       106       111       115       116       121       129
## 0.4328497 0.4350709 0.4311646 0.4281034 0.4301366 0.4334429 0.4310221 0.4323467
##       132       133       134       148       169       170       182       201
## 0.4330148 0.4289590 0.4262658 0.4299770 0.4113155 0.4069813 0.4065361 0.4857431
##       209       214       215       223       228       234       256       258
## 0.4904686 0.4812145 0.4814063 0.3837392 0.3831597 0.3908117 0.3927166 0.3868925
##       259       264       275       276       287       289       290       291
## 0.3847808 0.5282703 0.5264802 0.5297633 0.5312839 0.5323758 0.5281247 0.5322792
##       294       296       299       302       308       311       315       318
## 0.5330437 0.5355375 0.5242682 0.5171013 0.5381503 0.5291678 0.4828010 0.4952438
```

```
##       323       324       329       330       334       335       336       337
## 0.5036609 0.4968335 0.4905004 0.5063241 0.4948204 0.5029410 0.4926432 0.5013913
##       341       342       343       344       348       351       353       355
## 0.5050975 0.5024371 0.4993732 0.5021283 0.5039077 0.4896620 0.4995032 0.5012443
##       368       376       384       396       400       401       413       414
## 0.5066867 0.4973724 0.5084361 0.5062471 0.5051836 0.5091586 0.5100057 0.5145879
##       415       417       425       429       435       439       443       445
## 0.5149277 0.5214801 0.5069030 0.5288005 0.5266297 0.5371753 0.5267915 0.5304192
##       452       456       459       469       481       483       484       496
## 0.5331213 0.5208522 0.5220760 0.5321491 0.5978583 0.5934059 0.5948171 0.5986013
##       497       507       508       515       519       537       541       547
## 0.5986763 0.6038127 0.6012547 0.5961682 0.6005822 0.3978305 0.3978343 0.3960208
##       555       559       571       572       580       585       586       587
## 0.3958613 0.4191917 0.3989136 0.3875721 0.5293695 0.5491352 0.5509571 0.5334108
##       589       590       593       595       597       604       605       611
## 0.5263939 0.5444975 0.5245611 0.5443852 0.5306618 0.5444920 0.5414994 0.5337899
##       620       622       624       627       630       640       660       664
## 0.5415742 0.5333783 0.5383492 0.5418083 0.5469386 0.5489273 0.5508396 0.5457490
##       668       670       672       673       688       689       700       716
## 0.5370280 0.5425286 0.5509906 0.5407891 0.4941760 0.4980661 0.4664466 0.3602761
##       719       720       723       729       738       744       745       746
## 0.3627822 0.3603877 0.3651183 0.3604539 0.3666744 0.3669258 0.3661145 0.3632764
##       748       750       752       757       768       770       773       774
## 0.3650092 0.3598891 0.3674481 0.3601592 0.4572280 0.4697694 0.4630275 0.4556623
##       775       788       793       796       798       810       813       819
## 0.4622980 0.4587810 0.4745558 0.4679458 0.4331647 0.4560585 0.4608187 0.5061314
##       821       822       829       833       836       838       849       851
## 0.4981720 0.5062066 0.5029046 0.5184127 0.5080530 0.5049594 0.5139421 0.5062448
##       856       867       868       873       874       877       887       890
## 0.4991055 0.5138775 0.4937028 0.5076435 0.5097097 0.5006558 0.5099633 0.5093218
##       906       912       917       919       921       930       931       933
## 0.5141549 0.5147914 0.5133462 0.5107586 0.5121171 0.5129092 0.5159085 0.5086590
##       934       941       942       943       947       953       956       960
## 0.5083248 0.4298715 0.4222117 0.4208704 0.4208084 0.4272206 0.4215521 0.4223893
##       966       969       973       985       992       997      1002      1008
## 0.4234367 0.4304863 0.4293402 0.4255680 0.5013253 0.5146116 0.5081951 0.4920941
##      1026      1028      1029      1043      1046      1049      1056      1057
## 0.5070490 0.5173333 0.5030806 0.4384160 0.4413300 0.4418902 0.4402768 0.4415272
##      1067      1080      1086      1094      1100      1103      1107      1108
## 0.4430571 0.4379423 0.4400112 0.4349297 0.4983043 0.4966241 0.4926153 0.4911797
##      1109      1114      1116      1122      1130      1133      1134      1136
## 0.5059150 0.4894258 0.4910371 0.4989528 0.4875577 0.4956439 0.5041504 0.4800624
##      1142      1145      1149      1156      1157      1160      1164      1165
## 0.5620928 0.5743033 0.5634846 0.5639533 0.5709616 0.5600754 0.5627967 0.5452482
##      1166      1169      1171      1176      1185      1187      1194      1201
## 0.5432486 0.5399490 0.5445573 0.5541981 0.5410993 0.5465145 0.5438635 0.5427862
##      1208      1209      1210      1211      1214      1223      1226      1228
## 0.5408628 0.5389164 0.5440522 0.5493306 0.5393668 0.5309835 0.5383420 0.5387686
##      1236      1237      1249      1250      1272      1273      1277      1285
## 0.5390393 0.5344045 0.5313807 0.5343407 0.4251372 0.4503554 0.4716141 0.4460766
##      1288      1291      1300      1311      1315      1322      1324      1326
## 0.4419224 0.4361341 0.4455285 0.4370353 0.4406064 0.4450718 0.4426651 0.4568298
##      1332      1337      1343      1354      1359      1376      1377      1379
## 0.4436335 0.4558297 0.4439650 0.5066484 0.5084932 0.5054681 0.5036271 0.5052927
```

```
##      1380      1385      1389      1391      1406      1409      1411      1413
## 0.5050466 0.5079036 0.5082385 0.5071468 0.5699128 0.5819132 0.5926642 0.5801939
##      1420      1422      1423      1426      1427      1429      1432      1434
## 0.5821137 0.5742338 0.5786444 0.5848733 0.5694679 0.5819199 0.5689976 0.5704493
##      1440      1449      1453      1466      1468      1471      1474      1479
## 0.5805381 0.5610006 0.5912337 0.4024900 0.3967351 0.3960997 0.4038193 0.4151920
##      1480      1484      1486      1495
## 0.4105144 0.4028730 0.4026011 0.3910919
```

Confusion matrix

```
confmatrix <- table(Actual_value = test$label, Predicted_value = res >0.5)
confmatrix
```

```
##             Predicted_value
## Actual_value FALSE TRUE
##            0    74   74
##            1    69   83
```

Accuracy percentage

```
#calculating accuracy percentage
(confmatrix[[1]]+confmatrix[[2]])/sum(confmatrix)
```

```
## [1] 0.4766667
```

Our model turned out to be only 52% accurate.