Regressão Logística

> library(ROCR)

> library(sqldf)

> set.seed(1984)

> cc = read.csv("D:/academic-projects/regressao/qconlondon2016\_sample\_data.csv")

> cc <- sqldf("select case when fraudulent = 'True' then

+ 1 else 0 end fraudulent,

+ charge\_time, amount, card\_country, card\_use\_24h

+ from cc")

> #Cria valores de teste e treino

> teste = sample(1:nrow(cc),round(0.3\*nrow(cc)))

> ccteste = cc[teste,]

> cctrain = cc[-teste,]

> str(cctrain)

'data.frame': 62575 obs. of 5 variables:

$ fraudulent : int 0 0 0 0 0 0 0 0 0 0 ...

$ charge\_time : Factor w/ 45586 levels "2015-12-13T11:55:30Z",..: 45586 45586 45586 45586 45586 45586 45586 45586 45586 45586 ...

$ amount : int 20484 1211 8396 1480 535 1632 10305 2783 939 2256 ...

$ card\_country: Factor w/ 3 levels "AU","GB","US": 3 3 3 3 3 3 3 3 3 3 ...

$ card\_use\_24h: int 0 0 1 3 3 0 1 0 0 0 ...

> #Altera vaiaveis para tipo numerico

> ccteste$charge\_time = as.numeric(ccteste$charge\_time)

> cctrain$charge\_time = as.numeric(cctrain$charge\_time)

> cctrain$fraudulent = as.numeric(cctrain$fraudulent)

> table(cc$card\_country)

AU GB US

2145 2754 84494

> modelo = glm(fraudulent~., data=cctrain, family=binomial(link="logit"))

Warning message:

glm.fit: probabilidades ajustadas numericamente 0 ou 1 ocorreu

> summary(modelo)

Call:

glm(formula = fraudulent ~ ., family = binomial(link = "logit"),

data = cctrain)

Deviance Residuals:

Min 1Q Median 3Q Max

-3.9344 -0.0338 -0.0021 0.0000 5.0567

Coefficients:

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

(Intercept) 1.852e+02 4.164e+00 44.469 <2e-16 \*\*\*

charge\_time -4.139e-03 9.305e-05 -44.479 <2e-16 \*\*\*

amount 6.851e-07 5.433e-07 1.261 0.207

card\_countryGB -9.187e+00 3.397e-01 -27.048 <2e-16 \*\*\*

card\_countryUS -9.672e+00 2.309e-01 -41.888 <2e-16 \*\*\*

card\_use\_24h 2.424e-03 9.754e-04 2.485 0.013 \*

---

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 86739.3 on 62574 degrees of freedom

Residual deviance: 4352.6 on 62569 degrees of freedom

AIC: 4364.6

Number of Fisher Scoring iterations: 13

> predict\_teste = predict(modelo, newdata=ccteste, type="response")>0.5

> c\_matrix=table(ccteste$fraudulent,predict\_test)

> c\_matrix=table(ccteste$fraudulent,predict\_teste)

> print(c\_matrix)

predict\_teste

FALSE TRUE

0 13481 51

1 184 13102

> cat('Accuracy: ', sum(diag(c\_matrix))/sum(c\_matrix)\*100, ' %')

Accuracy: 99.12372 %

Árvore de decisão

> cc = read.csv("D:/academic-projects/regressao/qconlondon2016\_sample\_data.csv")

> library(sqldf)

Carregando pacotes exigidos: gsubfn

Carregando pacotes exigidos: proto

Carregando pacotes exigidos: RSQLite

> cc = sqldf("SELECT \* FROM cc

+ ORDER BY RANDOM() LIMIT 1000")

> write.csv(cc, file = "D:/academic-projects/regressao/cc-1000.csv")

> rm(list=ls())

> cc = read.csv("D:/academic-projects/regressao/cc-1000.csv")

> L <- sample(1:nrow(cc),round(nrow(cc)/3))

> train <- cc[-L,]

> test <- cc[L,]

> library(party)

Carregando pacotes exigidos: grid

Carregando pacotes exigidos: mvtnorm

Carregando pacotes exigidos: modeltools

Carregando pacotes exigidos: stats4

Carregando pacotes exigidos: strucchange

Carregando pacotes exigidos: zoo

Attaching package: ‘zoo’

The following objects are masked from ‘package:base’:

as.Date, as.Date.numeric

Carregando pacotes exigidos: sandwich

> fit <- ctree(fraudulent ~., train)

> predict\_test = predict(fit, newdata=test)

> c\_matrix=table(test$fraudulent,predict\_test)

> print(c\_matrix)

predict\_test

False True

False 148 27

True 102 56

> cat('Accuracy: ', sum(diag(c\_matrix))/sum(c\_matrix)\*100, ' %')

Accuracy: 61.26126 %

> plot(fit)

> plot(fit)

Naive bayes

> library(e1071)

> cc = read.csv("D:/academic-projects/regressao/qconlondon2016\_sample\_data.csv")

> cc$charge\_time=as.numeric(cc$charge\_time)

> # Treinamento e teste

> set.seed(1984)

> T = sample(1:nrow(cc),round(0.3\*nrow(cc)))

> ccteste = cc[T,]

> cctreino = cc[-T,]

> nb = naiveBayes(fraudulent~.,data=cctreino,laplace=1)

> nb$apriori

Y

False True

31642 30933

> nb$tables

$charge\_time

charge\_time

Y [,1] [,2]

False 44226.89 787.7086

True 21684.48 12591.2082

$amount

amount

Y [,1] [,2]

False 6758.131 42986.58

True 12117.250 40860.70

$card\_country

card\_country

Y AU GB US

False 0.005435298 0.017980724 0.976583979

True 0.043767779 0.042539436 0.913692785

$card\_use\_24h

card\_use\_24h

Y [,1] [,2]

False 2.180298 21.66117

True 5.582582 21.98128

> predict\_test = predict(nb, newdata=ccteste)

> head(predict\_test)

[1] True False False False True False

Levels: False True

> #probabilidade por classe

> predict\_test = predict(nb, newdata=ccteste, type="raw")

> head(predict\_test)

False True

[1,] 6.763667e-71 1.00000000

[2,] 9.725257e-01 0.02747434

[3,] 9.828931e-01 0.01710686

[4,] 9.865683e-01 0.01343167

[5,] 2.675537e-157 1.00000000

[6,] 9.791038e-01 0.02089618

> #matriz de confusão e acuricidade

> #predição

> predict\_test = predict(nb, newdata=ccteste)

> #matriz de confusão e acuricidade

> c\_matrix=table(ccteste$fraudulent, predict\_test)

> print(c\_matrix)

predict\_test

False True

False 13527 5

True 4680 8606

> cat('Accuracy: ', sum(diag(c\_matrix))/sum(c\_matrix)\*100, ' %')

Accuracy: 82.53039 %

> #Curva ROC

> pr=prediction(as.numeric(predict\_test),as.numeric(ccteste$fraudulent))

> prf=performance(pr, measure="tpr", x.measure="fpr")

> plot(prf ,colorize=TRUE)

> auc=performance(pr, measure="auc")

> auc=auc@y.values[[1]]

> auc

[1] 0.82369

Naive bayes (bônus com função cut(20))

> library(ROCR)

Carregando pacotes exigidos: gplots

Attaching package: ‘gplots’

The following object is masked from ‘package:stats’:

lowess

> library(e1071)

> cc = read.csv("D:/academic-projects/regressao/qconlondon2016\_sample\_data.csv")

> cc$charge\_time=as.numeric(cc$charge\_time)

> cc$amount = cut(cc$amount,20)

> cc$charge\_time=cut(as.numeric(cc$charge\_time),20)

> cc$card\_use\_24h = cut(cc$card\_use\_24h,20)

> set.seed(1984)

> T = sample(1:nrow(cc),round(0.3\*nrow(cc)))

> ccteste=cc[T,]

> cctreino=cc[-T,]

> nb = naiveBayes(fraudulent~.,data=cctreino,laplace=1)

> predict\_test = predict(nb, newdata=ccteste)

> c\_matrix=table(ccteste$fraudulent,predict\_test)

> print(c\_matrix)

predict\_test

False True

False 13527 5

True 635 12651

> cat('Accuracy: ', sum(diag(c\_matrix))/sum(c\_matrix)\*100, ' %')

Accuracy: 97.61354 %

> pr=prediction(as.numeric(predict\_test),as.numeric(ccteste$fraudulent))

> prf=performance(pr, measure="tpr", x.measure="fpr")

> plot(prf ,colorize=TRUE)

> auc=performance(pr, measure="auc")

> auc=auc@y.values[[1]]

> auc

[1] 0.9759179

Naive bayes (bônus com função cut(100))

> cc = read.csv("D:/academic-projects/regressao/qconlondon2016\_sample\_data.csv")

> cc$charge\_time=as.numeric(cc$charge\_time)

> cc$amount = cut(cc$amount,100)

> cc$charge\_time=cut(as.numeric(cc$charge\_time),100)

> cc$card\_use\_24h = cut(cc$card\_use\_24h,100)

> set.seed(1984)

> T = sample(1:nrow(cc),round(0.3\*nrow(cc)))

> ccteste=cc[T,]

> cctreino=cc[-T,]

> nb = naiveBayes(fraudulent~.,data=cctreino,laplace=1)

> predict\_test = predict(nb, newdata=ccteste)

> c\_matrix=table(ccteste$fraudulent,predict\_test)

> print(c\_matrix)

predict\_test

False True

False 13531 1

True 150 13136

> cat('Accuracy: ', sum(diag(c\_matrix))/sum(c\_matrix)\*100, ' %')

Accuracy: 99.43695 %

> pr=prediction(as.numeric(predict\_test),as.numeric(ccteste$fraudulent))

> prf=performance(pr, measure="tpr", x.measure="fpr")

> plot(prf ,colorize=TRUE)

> auc=performance(pr, measure="auc")

> auc=auc@y.values[[1]]

> auc

[1] 0.994318