Data Science Academy - Mini-Projeto 4

Equipe DSA

15 Julho, 2018

Mini-Projeto 4 - Avaliação de Risco de Crédito

Para esta análise, vamos usar um conjunto de dados German Credit Data, já devidamente limpo e organizado para a criação do modelo preditivo.

Todo o projeto será descrito de acordo com suas etapas.

Etapa 1 - Coletando os Dados

Aqui está a coleta de dados, neste caso um arquivo csv.

```
# Coletando dados
credit.df <- read.csv("credit_dataset.csv", header = TRUE, sep = ",")
```

Etapa 2 - Normalizando os Dados

```
## Convertendo as variáveis para o tipo fator (categórica)
to.factors <- function(df, variables){
  for (variable in variables){
    df[[variable]] <- as.factor(df[[variable]])
  }
  return(df)
}

## Normalização
scale.features <- function(df, variables){
  for (variable in variables){
    df[[variable]] <- scale(df[[variable]], center=T, scale=T)
  }
  return(df)
}

# Normalizando as variáveis
numeric.vars <- c("credit.duration.months", "age", "credit.amount")
credit.df <- scale.features(credit.df, numeric.vars)

# Variáveis do tipo fator
categorical.vars <- c('credit.rating', 'account.balance', 'previous.credit.payment.status',
                      'credit.purpose', 'savings', 'employment.duration', 'installment.rate',
                      'marital.status', 'guarantor', 'residence.duration', 'current.assets',
                      'other.credits', 'apartment.type', 'bank.credits', 'occupation',
                      'dependents', 'telephone', 'foreign.worker')

credit.df <- to.factors(df = credit.df, variables = categorical.vars)
```

Etapa 3 - Dividindo os dados em dados de treino e de teste

```
# Dividindo os dados em treino e teste - 60:40 ratio
indexes <- sample(1:nrow(credit.df), size = 0.6 * nrow(credit.df))
train.data <- credit.df[indexes,]
test.data <- credit.df[-indexes,]
```

Etapa 4 - Feature Selection

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
# Função para seleção de variáveis
run.feature.selection <- function(num.iters=20, feature.vars, class.var){
  set.seed(10)
  variable.sizes <- 1:10
  control <- rfeControl(functions = rfFuncs, method = "cv",
                        verbose = FALSE, returnResamp = "all",
                        number = num.iters)
  results.rfe <- rfe(x = feature.vars, y = class.var,
                     sizes = variable.sizes,
                     rfeControl = control)
  return(results.rfe)
}

# Executando a função
rfe.results <- run.feature.selection(feature.vars = train.data[,-1],
                                     class.var = train.data[,1])


# Visualizando os resultados
rfe.results
```

```
##
## Recursive feature selection
##
## Outer resampling method: Cross-Validated (20 fold)
##
## Resampling performance over subset size:
##
```

```
##  Variables Accuracy  Kappa AccuracySD KappaSD Selected
##          1   0.6685 0.1027    0.04232  0.1138
##          2   0.7221 0.1611    0.04378  0.1392
##          3   0.7239 0.2430    0.07239  0.1977
##          4   0.7455 0.3573    0.08945  0.2224
##          5   0.7604 0.3898    0.07114  0.1819
##          6   0.7482 0.3630    0.07089  0.1664
##          7   0.7500 0.3465    0.06335  0.1709
##          8   0.7532 0.3538    0.07345  0.2020
##          9   0.7618 0.3876    0.06833  0.1688
##         10   0.7468 0.3427    0.08000  0.2132
##         20   0.7701 0.3809    0.05679  0.1675        *
##
## The top 5 variables (out of 20):
##    account.balance, previous.credit.payment.status, credit.duration.months, credit.amount, employment
```

```
varImp((rfe.results))
```

```
##                                 Overall
## account.balance                19.222687
## previous.credit.payment.status 11.608962
## credit.duration.months          9.197917
## credit.amount                   7.399089
## employment.duration             5.318123
## current.assets                  5.068656
## age                             4.927403
## savings                         4.403029
## marital.status                  3.657635
## guarantor                       3.409746
## installment.rate                3.251218
## other.credits                   2.800881
## residence.duration              1.801696
## apartment.type                  1.791801
## bank.credits                    1.709373
## foreign.worker                  1.401358
## occupation                      1.379986
## telephone                       1.361878
## dependents                      1.256409
## credit.purpose                  1.220637
```

Etapa 5 - Criando e Avaliando a Primeira Versão do Modelo

```
# Criando e Avaliando o Modelo
library(caret)
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

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

```r
# Biblioteca de utilitários para construção de gráficos
source("plot_utils.R")

## separate feature and class variables
test.feature.vars <- test.data[,-1]
test.class.var <- test.data[,1]

# Construindo um modelo de regressão logística
formula.init <- "credit.rating ~ ."
formula.init <- as.formula(formula.init)
lr.model <- glm(formula = formula.init, data = train.data, family = "binomial")

# Visualizando o modelo
summary(lr.model)
```

```
##
## Call:
## glm(formula = formula.init, family = "binomial", data = train.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6067  -0.6691   0.3670   0.7085   2.0151
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    -0.592826   0.990174  -0.599 0.549367
## account.balance2                0.523654   0.285035   1.837 0.066187 .
## account.balance3                1.659172   0.278307   5.962  2.5e-09 ***
## credit.duration.months         -0.305335   0.140018  -2.181 0.029207 *
## previous.credit.payment.status2  0.993798   0.397535   2.500 0.012423 *
## previous.credit.payment.status3  1.475400   0.416971   3.538 0.000403 ***
## credit.purpose2                -0.907978   0.524438  -1.731 0.083392 .
## credit.purpose3                -0.976871   0.511980  -1.908 0.056388 .
## credit.purpose4                -1.084367   0.499433  -2.171 0.029916 *
## credit.amount                  -0.376575   0.156547  -2.405 0.016150 *
## savings2                        0.151849   0.378481   0.401 0.688269
## savings3                        0.976683   0.439762   2.221 0.026355 *
## savings4                        0.730716   0.337878   2.163 0.030567 *
## employment.duration2            0.500298   0.305116   1.640 0.101068
## employment.duration3            1.028803   0.378634   2.717 0.006585 **
## employment.duration4            0.753333   0.353348   2.132 0.033008 *
## installment.rate2               0.003947   0.397921   0.010 0.992086
## installment.rate3              -0.587746   0.437452  -1.344 0.179088
## installment.rate4              -0.973771   0.381088  -2.555 0.010612 *
## marital.status3                 0.726825   0.263675   2.757 0.005842 **
## marital.status4                 0.483523   0.418321   1.156 0.247736
## guarantor2                      0.669917   0.388194   1.726 0.084396 .
## residence.duration2            -1.144972   0.396844  -2.885 0.003912 **
## residence.duration3            -0.713823   0.441376  -1.617 0.105821
## residence.duration4            -0.689571   0.399838  -1.725 0.084595 .
## current.assets2                -0.121938   0.332015  -0.367 0.713421
## current.assets3                -0.264286   0.304989  -0.867 0.386193
## current.assets4                -1.066976   0.523762  -2.037 0.041636 *
## age                             0.101935   0.132516   0.769 0.441756
```

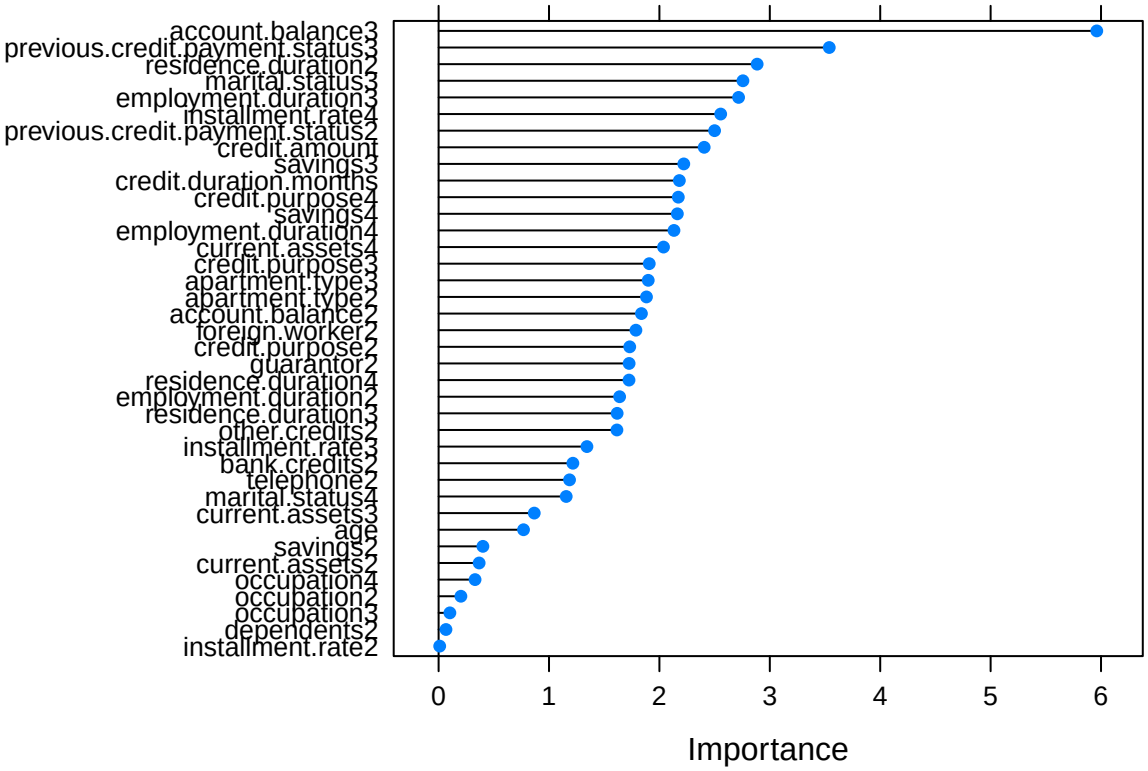
```
## other.credits2                   0.486325   0.301082   1.615 0.106255
## apartment.type2                  0.559441   0.297121   1.883 0.059717 .
## apartment.type3                  1.143958   0.602486   1.899 0.057600 .
## bank.credits2                   -0.361162   0.296962  -1.216 0.223913
## occupation2                     -0.147575   0.730559  -0.202 0.839914
## occupation3                      0.071495   0.699782   0.102 0.918623
## occupation4                      0.246779   0.747099   0.330 0.741162
## dependents2                      0.022574   0.338547   0.067 0.946838
## telephone2                       0.310937   0.262135   1.186 0.235554
## foreign.worker2                  1.799731   1.007001   1.787 0.073902 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 739.69  on 599  degrees of freedom
## Residual deviance: 534.59  on 561  degrees of freedom
## AIC: 612.59
##
## Number of Fisher Scoring iterations: 5
```

```r
# Testando o modelo nos dados de teste
lr.predictions <- predict(lr.model, test.data, type="response")
lr.predictions <- round(lr.predictions)

# Avaliando o modelo
confusionMatrix(table(data = lr.predictions, reference = test.class.var), positive = '1')
```

```
## Confusion Matrix and Statistics
##
##     reference
## data   0   1
##    0  53  28
##    1  63 256
##
##                Accuracy : 0.7725
##                  95% CI : (0.7282, 0.8127)
##     No Information Rate : 0.71
##     P-Value [Acc > NIR] : 0.002932
##
##                   Kappa : 0.3934
##  Mcnemar's Test P-Value : 0.000365
##
##             Sensitivity : 0.9014
##             Specificity : 0.4569
##          Pos Pred Value : 0.8025
##          Neg Pred Value : 0.6543
##              Prevalence : 0.7100
##          Detection Rate : 0.6400
##    Detection Prevalence : 0.7975
##       Balanced Accuracy : 0.6792
##
##        'Positive' Class : 1
##
```

Etapa 6 - Otimizando o Modelo

```
## Feature selection
formula <- "credit.rating ~ ."
formula <- as.formula(formula)
control <- trainControl(method = "repeatedcv", number = 10, repeats = 2)
model <- train(formula, data = train.data, method = "glm", trControl = control)
importance <- varImp(model, scale = FALSE)
plot(importance)
```




```
# Construindo o modelo com as variáveis selecionadas
formula.new <- "credit.rating ~ account.balance + credit.purpose + previous.credit.payment.status + savi
formula.new <- as.formula(formula.new)
lr.model.new <- glm(formula = formula.new, data = train.data, family = "binomial")

# Visualizando o modelo
summary(lr.model.new)
```

```
##
## Call:
## glm(formula = formula.new, family = "binomial", data = train.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4675  -0.8558   0.4772   0.8072   2.0662
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -0.54701    0.52872  -1.035 0.300863
```

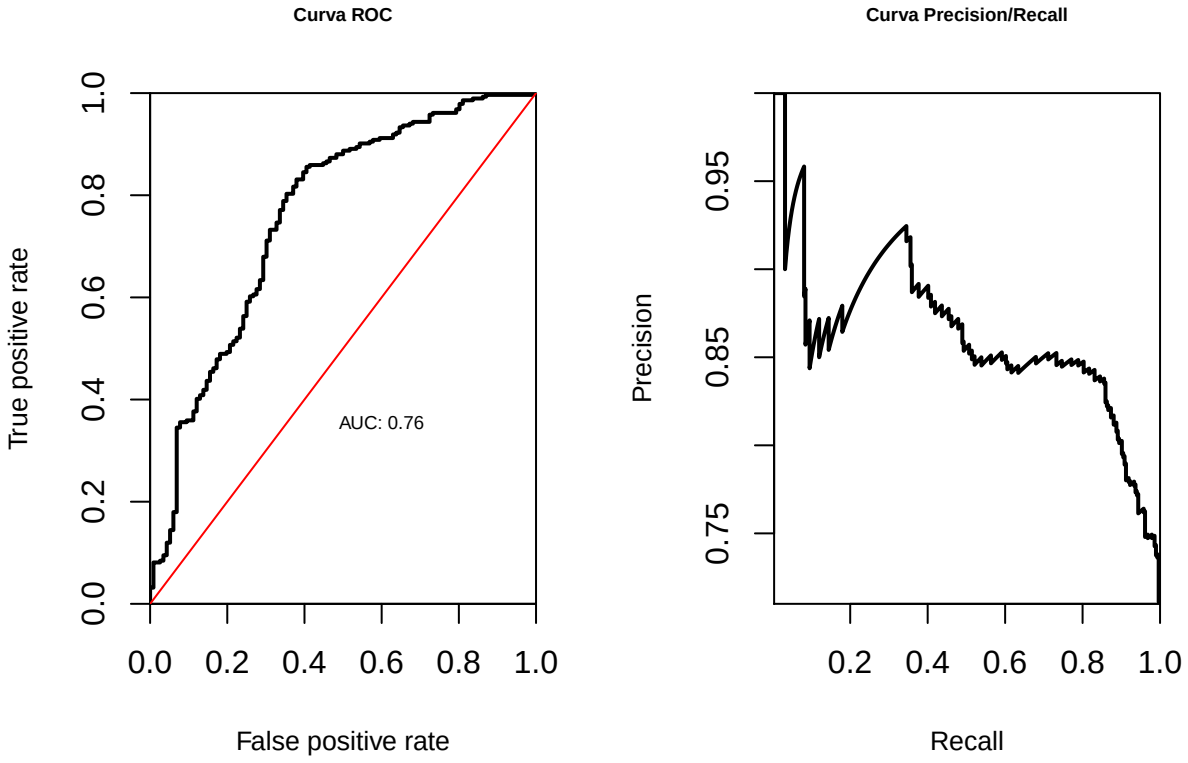
```
## account.balance2                 0.40516    0.25028   1.619 0.105485
## account.balance3                 1.58721    0.25215   6.295 3.08e-10 ***
## credit.purpose2                 -0.79596    0.46986  -1.694 0.090258 .
## credit.purpose3                 -0.65421    0.44928  -1.456 0.145360
## credit.purpose4                 -0.87482    0.44872  -1.950 0.051226 .
## previous.credit.payment.status2  1.19823    0.33963   3.528 0.000419 ***
## previous.credit.payment.status3  1.60602    0.35792   4.487 7.22e-06 ***
## savings2                         0.01151    0.33810   0.034 0.972838
## savings3                         0.83412    0.39744   2.099 0.035839 *
## savings4                         0.67366    0.30622   2.200 0.027810 *
## credit.duration.months          -0.44660    0.10049  -4.444 8.82e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 739.69  on 599  degrees of freedom
## Residual deviance: 604.05  on 588  degrees of freedom
## AIC: 628.05
##
## Number of Fisher Scoring iterations: 4
```

```
# Testando o modelo nos dados de teste
lr.predictions.new <- predict(lr.model.new, test.data, type="response")
lr.predictions.new <- round(lr.predictions.new)

# Avaliando o modelo
confusionMatrix(table(data=lr.predictions.new, reference=test.class.var), positive='1')
```

```
## Confusion Matrix and Statistics
##
##     reference
## data   0   1
##    0  40  26
##    1  76 258
##
##                Accuracy : 0.745
##                  95% CI : (0.6993, 0.787)
##     No Information Rate : 0.71
##     P-Value [Acc > NIR] : 0.0671
##
##                   Kappa : 0.2903
##  Mcnemar's Test P-Value : 1.224e-06
##
##             Sensitivity : 0.9085
##             Specificity : 0.3448
##          Pos Pred Value : 0.7725
##          Neg Pred Value : 0.6061
##              Prevalence : 0.7100
##          Detection Rate : 0.6450
##    Detection Prevalence : 0.8350
##       Balanced Accuracy : 0.6266
##
##        'Positive' Class : 1
##
```

Etapa 7 - Curva ROC e Avaliação Final do Modelo

```
# Avaliando a performance do modelo

# Criando curvas ROC
lr.model.best <- lr.model
lr.prediction.values <- predict(lr.model.best, test.feature.vars, type = "response")
predictions <- prediction(lr.prediction.values, test.class.var)
par(mfrow = c(1,2))
plot.roc.curve(predictions, title.text = "Curva ROC")
plot.pr.curve(predictions, title.text = "Curva Precision/Recall")
```



Fim

www.datascienceacademy.com.br