Análise de Crédito

Lucas Braga

Mar 20, 2022

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.

Etapa 1 - Coletando os Dados

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

Etapa 2 - Normalizando os Dados

```r
## 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)
```

```
## Carregando pacotes exigidos: ggplot2
```

```
## Carregando pacotes exigidos: lattice
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.1.3
```

```
## randomForest 4.7-1
```

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

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

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

```
# Função para a 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 0.005795    0.05077 0.03166
##          2   0.7369 0.322169    0.05689 0.15270
##          3   0.7702 0.402612    0.06628 0.17154        *
##          4   0.7301 0.302089    0.04830 0.14685
##          5   0.7217 0.276644    0.05930 0.15716
##          6   0.7332 0.304894    0.06145 0.17548
##          7   0.7550 0.367032    0.05193 0.15350
##          8   0.7583 0.377159    0.06302 0.17158
##          9   0.7684 0.408660    0.06511 0.16532
##         10   0.7617 0.388719    0.08424 0.21816
##         20   0.7534 0.343770    0.06004 0.17636
##
## The top 3 variables (out of 3):
##    account.balance, credit.duration.months, previous.credit.payment.status
```

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

```
##                                  Overall
## account.balance                20.461113
## credit.duration.months         11.750240
## previous.credit.payment.status  8.721829
```

Etapa 5 - Criando e Avaliando a Primeira VersÃ£o do Modelo

```
# Criando e Avaliando o modelo

library(caret)
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 4.1.3
```

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

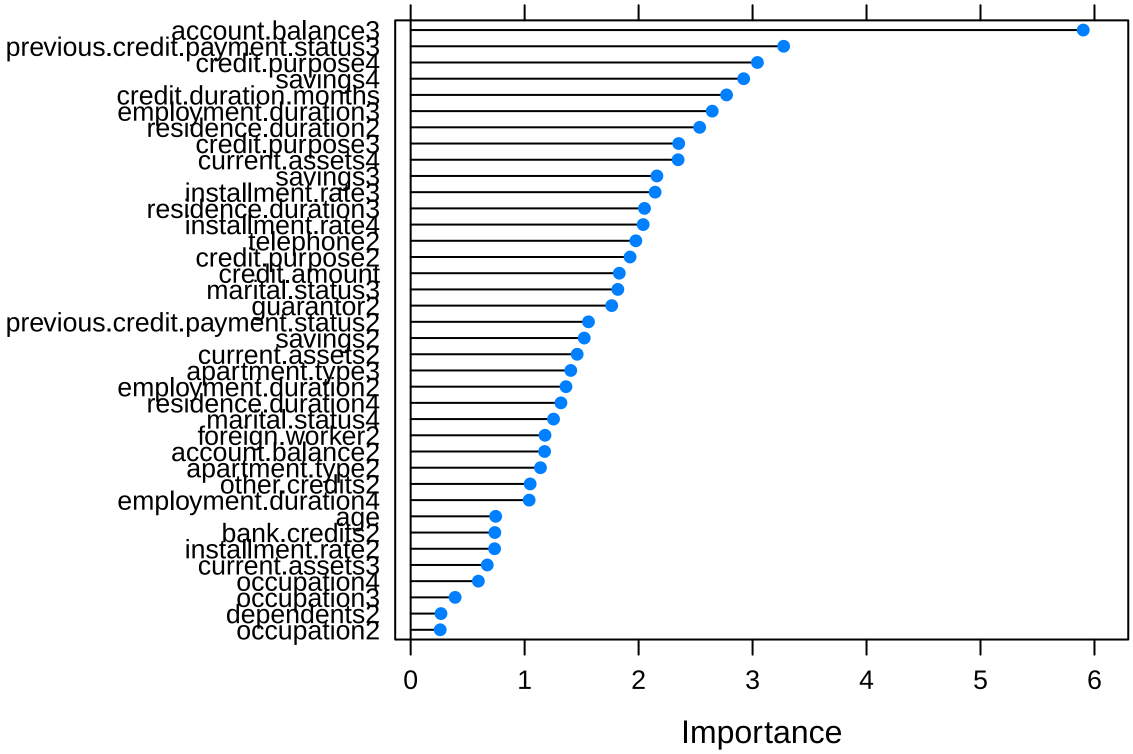
```r
## separate feature and class variables
test.feature.vars <- test.data[,-1]
test.class.var <- test.data[,1]

# 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
## -2.6096  -0.6960   0.3963   0.6975   1.9790
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      -0.006347   0.995828  -0.006  0.99491
## account.balance2                  0.330464   0.281196   1.175  0.23991
## account.balance3                  1.669280   0.282855   5.902  3.6e-09 ***
## credit.duration.months           -0.425459   0.153491  -2.772  0.00557 **
## previous.credit.payment.status2  0.606395   0.388679   1.560  0.11873
## previous.credit.payment.status3  1.364615   0.416978   3.273  0.00107 **
## credit.purpose2                  -0.944895   0.490800  -1.925  0.05420 .
## credit.purpose3                  -1.108599   0.471360  -2.352  0.01868 *
## credit.purpose4                  -1.405951   0.462085  -3.043  0.00235 **
## credit.amount                    -0.313153   0.171055  -1.831  0.06714 .
## savings2                          0.564717   0.370827   1.523  0.12779
## savings3                          0.937349   0.433851   2.161  0.03073 *
## savings4                          1.006356   0.344423   2.922  0.00348 **
## employment.duration2              0.419085   0.307527   1.363  0.17296
## employment.duration3              0.994730   0.375926   2.646  0.00814 **
## employment.duration4              0.380205   0.365692   1.040  0.29849
## installment.rate2                -0.300607   0.408265  -0.736  0.46155
## installment.rate3                -0.968825   0.451710  -2.145  0.03197 *
## installment.rate4                -0.805408   0.394807  -2.040  0.04135 *
## marital.status3                   0.481333   0.264784   1.818  0.06909 .
## marital.status4                   0.516715   0.412208   1.254  0.21001
## guarantor2                        0.653375   0.370372   1.764  0.07771 .
## residence.duration2              -0.940638   0.370980  -2.536  0.01123 *
## residence.duration3              -0.850315   0.414432  -2.052  0.04019 *
## residence.duration4              -0.497069   0.377014  -1.318  0.18736
## current.assets2                  -0.470373   0.321938  -1.461  0.14400
## current.assets3                  -0.209109   0.311130  -0.672  0.50152
## current.assets4                  -1.317287   0.561311  -2.347  0.01894 *
## age                               0.100947   0.135326   0.746  0.45570
## other.credits2                    0.303300   0.289329   1.048  0.29451
## apartment.type2                   0.362240   0.318029   1.139  0.25470
## apartment.type3                  0.895125   0.637239   1.405  0.16011
## bank.credits2                   -0.225750   0.305796  -0.738  0.46037
## occupation2                      0.197890   0.763729   0.259  0.79555
## occupation3                      0.289355   0.741472   0.390  0.69636
## occupation4                      0.478035   0.804834   0.594  0.55254
## dependents2                      0.090961   0.342057   0.266  0.79030
## telephone2                       0.518679   0.262534   1.976  0.04819 *
## foreign.worker2                  0.936148   0.794173   1.179  0.23849
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 734.72  on 599  degrees of freedom
## Residual deviance: 539.22  on 561  degrees of freedom
## AIC: 617.22
##
## Number of Fisher Scoring iterations: 5
```

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

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)
```



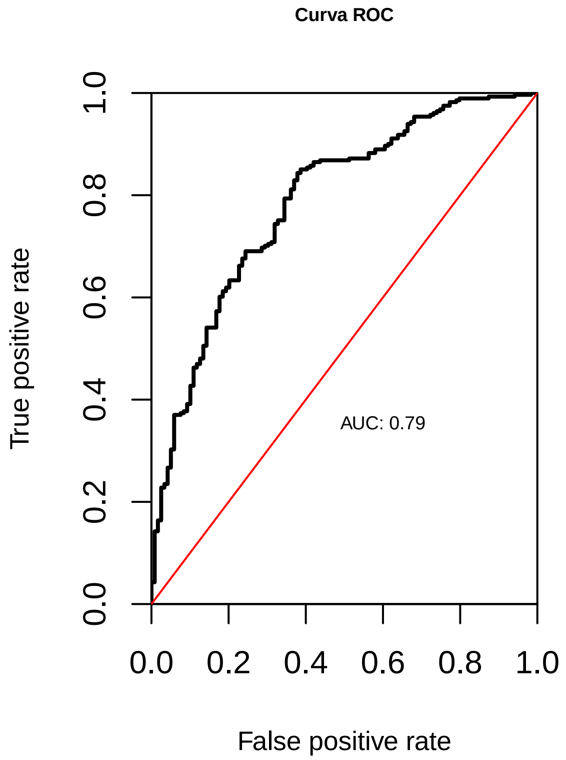
```
# Construindo o modelo com as variáveis selecionadas
formula.new <- "credit.rating ~ account.balance + credit.purpose + previous.credit.payment.status + savi
formula.new <- as.formula(formula.new)
lr.model.new <- glm(formula = formula.new, data = train.data, family = "binomial")

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

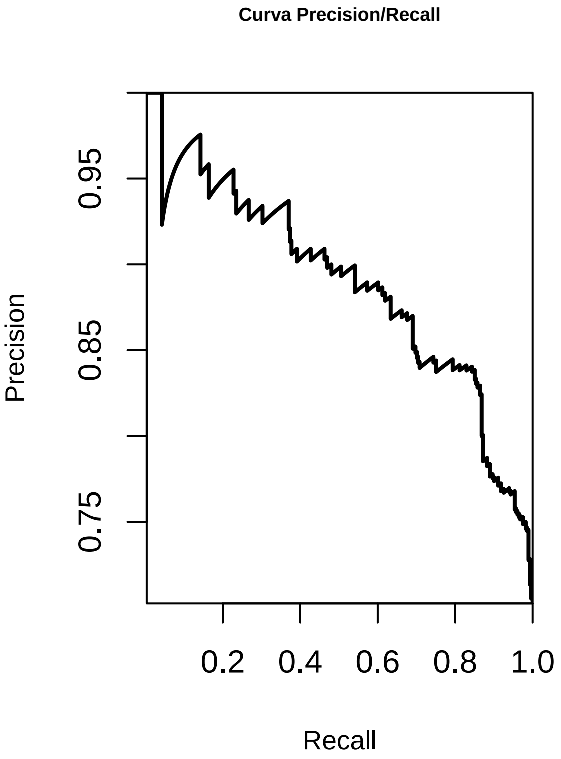
```
##
## Call:
## glm(formula = formula.new, family = "binomial", data = train.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5039  -0.8142   0.4552   0.7607   1.8337
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       -0.1450     0.4827  -0.300 0.763965
## account.balance2                   0.2441     0.2509   0.973 0.330581
## account.balance3                   1.5905     0.2618   6.075 1.24e-09 ***
## credit.purpose2                   -0.6826     0.4394  -1.553 0.120336
## credit.purpose3                   -0.6992     0.4091  -1.709 0.087391 .
## credit.purpose4                   -0.9440     0.4055  -2.328 0.019904 *
## previous.credit.payment.status2    0.7560     0.3344   2.261 0.023761 *
## previous.credit.payment.status3    1.3727     0.3589   3.825 0.000131 ***
## savings2                          0.4238     0.3366   1.259 0.207990
## savings3                          0.7440     0.4017   1.852 0.064031 .
## savings4                          0.7128     0.3119   2.286 0.022270 *
## credit.duration.months           -0.5279     0.1031  -5.120 3.06e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 734.72  on 599  degrees of freedom
## Residual deviance: 589.60  on 588  degrees of freedom
## AIC: 613.6
##
## Number of Fisher Scoring iterations: 5
```

```
# 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)
```

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")
```


Curva ROC
True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
AUC: 0.79


Curva Precision/Recall
Precision
0.75
0.85
0.95
Recall
0.2
0.4
0.6
0.8
1.0