Analise de Crédito - Machine Learning

Michel H S Nascimento

2 de fevereiro de 2018

Conhecendo os dados

Este documento trata de um estudo sobre machine learning, através da execução de exercícios de um curso de ML da Udemy que visa analisar a disponibilidade de crédito das pessoas através de um conjunto de dados.

A biblioteca usada para o algoritmo de ML foi a "e1071".

```
#install.packages("e1071")
library(e1071)

## Warning: package 'e1071' was built under R version 3.3.3
```

Importando os dados de créditos:

```
dados_creditos = read.csv(file.choose())

head(dados_creditos)

##   checking_status duration                   credit_history
## 1              <0        6 'critical/other existing credit'
## 2        0<=X<200       48                  'existing paid'
## 3   'no checking'       12 'critical/other existing credit'
## 4              <0       42                  'existing paid'
## 5              <0       24             'delayed previously'
## 6   'no checking'       36                  'existing paid'
##               purpose credit_amount     savings_status employment
## 1            radio/tv          1169 'no known savings'        >=7
## 2            radio/tv          5951               <100     1<=X<4
## 3           education          2096               <100     4<=X<7
## 4 furniture/equipment          7882               <100     4<=X<7
## 5           'new car'          4870               <100     1<=X<4
## 6           education          9055 'no known savings'     1<=X<4
##   installment_commitment      personal_status other_parties
## 1                      4        'male single'          none
## 2                      2 'female div/dep/mar'          none
## 3                      2        'male single'          none
## 4                      2        'male single'     guarantor
## 5                      3        'male single'          none
## 6                      2        'male single'          none
##   residence_since  property_magnitude age other_payment_plans
housing
## 1               4       'real estate'  67                none
```

```
own
## 2                2       'real estate'  22                none
own
## 3                3       'real estate'  49                none
own
## 4                4    'life insurance'  45                none 'for
free'
## 5                4 'no known property'  53                none 'for
free'
## 6                4 'no known property'  35                none 'for
free'
##   existing_credits                  job num_dependents own_telephone
## 1                2              skilled              1           yes
## 2                1              skilled              1          none
## 3                1 'unskilled resident'              2          none
## 4                1              skilled              2          none
## 5                2              skilled              2          none
## 6                1 'unskilled resident'              2           yes
##   foreign_worker class
## 1            yes  good
## 2            yes   bad
## 3            yes  good
## 4            yes  good
## 5            yes   bad
## 6            yes  good
```

Pode perceber no conjunto de dados as características de todas as instâncias na massa de dados, onde seu atributo final referente a classe descreve se tal instância é tida como de um bom ou mau pagador.

Criando Amostras

Para uma análise mais adequada de resultados para posteriormente colocá-los em produção com uma certa confiança, se faz necessário a divisão dos dados em amostras para testes e para treino:

```
amostras = sample(2, 1000, replace = T, prob = c(0.7,0.3))
amostras

##    [1] 1 1 1 1 2 1 2 2 1 1 2 2 1 2 2 2 2 1 1 1 2 1 2 2 1 2 1 1 1 1 1 1
2 1
##   [35] 2 1 2 2 2 2 1 2 2 1 1 1 1 1 2 1 1 1 1 1 2 1 2 2 1 1 1 1 1 1 1 1
1 1
##   [69] 2 1 1 1 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 1 2 1 1 1 2 1 2
1 1
##  [103] 1 1 1 2 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 2
1 2
##  [137] 2 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 2 2 1 1 2 2 1 1 1 1 1 2 1 1 2
1 1
##  [171] 2 1 2 1 1 2 2 1 2 1 2 1 1 1 1 2 2 1 1 1 2 2 1 1 1 2 1 2 1 1 2 1
```

```
1 1
##  [205] 1 1 2 2 1 1 2 2 1 1 1 2 2 2 1 2 2 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1
1 1
##  [239] 1 1 1 1 1 1 1 2 1 1 1 2 1 2 1 1 1 2 2 1 2 2 1 1 1 2 1 2 1 1 1 1
1 2
##  [273] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 2 1 2 2 2 1 2 1 1 2
1 1
##  [307] 1 1 1 1 2 2 2 1 1 1 1 1 1 1 2 1 2 2 2 1 2 1 1 1 1 1 1 1 1 1 2 2
2 1
##  [341] 1 2 1 2 2 1 1 1 1 1 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 2 2 1 1 2 1 2
1 1
##  [375] 1 2 1 2 2 1 1 1 1 1 1 2 1 1 1 2 1 2 2 2 2 2 1 2 1 1 2 1 1 2 2 2
1 1
##  [409] 1 2 1 1 1 2 2 1 1 1 2 1 1 2 2 2 1 1 2 1 1 1 1 2 2 1 2 2 1 1 1 1
1 1
##  [443] 1 1 2 2 2 2 2 2 1 1 2 2 1 1 1 1 1 2 1 1 1 2 1 2 2 1 1 1 1 1 1 2
1 2
##  [477] 2 2 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 2 2 1 1 2 1 1 1 1 2 1 2 1
1 2
##  [511] 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2 1 1 1 1 1 1 1 1 2 1 2
2 1
##  [545] 2 1 1 1 2 2 1 1 1 1 2 1 2 2 1 1 2 2 1 1 1 1 1 2 1 1 1 1 1 1 2 1
1 1
##  [579] 1 1 1 2 1 1 1 1 1 1 1 2 1 2 1 1 1 2 1 2 2 1 1 1 2 1 1 1 1 1 1 1
2 1
##  [613] 2 1 2 2 2 2 2 1 2 2 1 1 2 1 1 1 1 1 1 2 1 2 1 1 2 1 1 1 1 1 1 1
1 2
##  [647] 1 1 2 1 2 1 1 2 1 1 1 1 1 2 2 2 1 2 1 1 1 1 1 2 1 2 2 2 1 1 2 1
1 1
##  [681] 1 1 1 1 1 1 1 1 1 2 1 1 2 2 1 2 2 2 2 1 1 1 1 1 1 1 1 1 2 2 1 1
1 1
##  [715] 1 1 1 1 2 2 2 2 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 2 1 1 1 2 1 2 1 1
1 1
##  [749] 2 2 2 2 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 2 1 1 2 2 1
2 1
##  [783] 2 1 1 1 2 1 1 2 1 1 2 2 2 1 2 1 1 1 1 2 1 1 2 1 1 1 1 1 1 2 2 1
2 1
##  [817] 2 1 1 1 1 1 1 2 1 1 2 1 2 2 2 1 2 2 1 1 2 1 1 2 1 2 1 1 1 1 1 1
1 2
##  [851] 2 1 1 2 1 1 2 1 1 2 1 1 2 2 1 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 1 1
2 1
##  [885] 1 1 2 2 2 1 1 1 1 2 2 1 2 2 1 1 1 1 1 2 1 1 1 1 2 1 2 2 2 1 1 1
2 1
##  [919] 1 2 1 1 2 2 1 2 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1
1 2
##  [953] 1 2 2 2 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 2 2 2 2 2 1 1 1 1 1
1 1
##  [987] 1 1 1 1 1 1 2 1 2 2 1 1 1 1
```

```r
dados_treino = dados_creditos[amostras==1,]
dados_teste = dados_creditos[amostras==2,]
```

Criação do Modelo

Cria-se o modelo para análise dos dados, este é feito com base na função da biblioteca que foi carregada, neste caso naivebayes; Tal modelo permite identificar relações de proporcionalidade entre os atributos definidos na massa de dados:

```r
modelo = naiveBayes(class ~., dados_treino)
modelo

##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##       bad      good
## 0.3064516 0.6935484
##
## Conditional probabilities:
##       checking_status
## Y       'no checking'         <0      >=200   0<=X<200
##   bad      0.14832536 0.44497608 0.04784689 0.35885167
##   good     0.51585624 0.20718816 0.06131078 0.21564482
##
##       duration
## Y           [,1]     [,2]
##   bad  25.14833 14.13132
##   good 19.28753 10.97946
##
##       credit_history
## Y       'all paid' 'critical/other existing credit' 'delayed
previously'
##   bad  0.08133971                       0.15789474
0.08133971
##   good 0.03382664                       0.33403805
0.08245243
##       credit_history
## Y       'existing paid' 'no credits/all paid'
##   bad        0.60287081            0.07655502
##   good       0.52854123            0.02114165
##
##       purpose
## Y       'domestic appliance'   'new car'  'used car'    business
##   bad            0.009569378 0.296650718 0.057416268 0.110047847
##   good           0.010570825 0.207188161 0.118393235 0.086680761
```

```
##       purpose
## Y        education furniture/equipment       other    radio/tv
repairs
##   bad  0.076555024         0.191387560 0.014354067 0.205741627
0.033492823
##   good 0.035940803         0.179704017 0.010570825 0.319238901
0.027484144
##       purpose
## Y       retraining
##   bad  0.004784689
##   good 0.004228330
##
##       credit_amount
## Y          [,1]     [,2]
##   bad  4018.254 3615.574
##   good 2989.211 2469.153
##
##       savings_status
## Y      'no known savings'       <100     >=1000 100<=X<500 500<=X<1000
##   bad          0.09090909 0.73684211 0.01435407 0.11961722  0.03827751
##   good         0.21987315 0.56448203 0.06131078 0.09302326  0.06131078
##
##       employment
## Y              <1        >=7     1<=X<4     4<=X<7 unemployed
##   bad  0.22488038 0.21052632 0.36842105 0.11961722 0.07655502
##   good 0.16701903 0.26427061 0.33192389 0.17970402 0.05708245
##
##       installment_commitment
## Y          [,1]     [,2]
##   bad  3.095694 1.096511
##   good 2.938689 1.134234
##
##       personal_status
## Y      'female div/dep/mar' 'male div/sep' 'male mar/wid' 'male
single'
##   bad            0.34928230     0.09090909     0.07655502
0.48325359
##   good           0.30232558     0.04862579     0.10993658
0.53911205
##
##       other_parties
## Y      'co applicant'  guarantor       none
##   bad      0.04306220 0.02870813 0.92822967
##   good     0.03171247 0.05919662 0.90909091
##
##       residence_since
## Y          [,1]     [,2]
##   bad  2.827751 1.095882
##   good 2.824524 1.101266
##
```

```
##      property_magnitude
## Y      'life insurance' 'no known property' 'real estate'       car
##   bad         0.2344498           0.2392344     0.2200957 0.3062201
##   good        0.2367865           0.1162791     0.3234672 0.3234672
##
##      age
## Y         [,1]     [,2]
##   bad  33.77033 10.91862
##   good 35.83087 11.73114
##
##      other_payment_plans
## Y           bank       none     stores
##   bad  0.17703349 0.75598086 0.06698565
##   good 0.10993658 0.84778013 0.04228330
##
##      housing
## Y      'for free'        own       rent
##   bad  0.14354067 0.62679426 0.22966507
##   good 0.09090909 0.74630021 0.16279070
##
##      existing_credits
## Y         [,1]      [,2]
##   bad  1.306220 0.4823914
##   good 1.418605 0.5985793
##
##      job
## Y      'high qualif/self emp/mgmt' 'unemp/unskilled non res'
##   bad                   0.15789474                0.03349282
##   good                  0.13530655                0.02114165
##      job
## Y      'unskilled resident'    skilled
##   bad            0.19138756 0.61722488
##   good           0.20718816 0.63636364
##
##      num_dependents
## Y         [,1]      [,2]
##   bad  1.157895 0.3655178
##   good 1.158562 0.3656542
##
##      own_telephone
## Y          none       yes
##   bad  0.6315789 0.3684211
##   good 0.5940803 0.4059197
##
##      foreign_worker
## Y             no        yes
##   bad  0.01435407 0.98564593
##   good 0.05496829 0.94503171
```

Previsão

Usa-se o modelo obtido para a análise preditiva sobre a parcela de dados reservadas para o teste, assim os valores podem ser confrontados de modo que se possa verificar a confianãa do modelo:

```
previsao = predict(modelo, dados_teste)
previsao

##   [1] bad  good bad  bad  bad  good bad  good good good good good good
good
##  [15] good bad  good good good good good good bad  good good bad  good
good
##  [29] bad  good bad  good good good good good good bad  bad  good good
good
##  [43] good good good good bad  good bad  bad  good good good bad  bad
good
##  [57] bad  good bad  good good good bad  good good good good good good
good
##  [71] good bad  good good good good good good good good bad  good good
good
##  [85] good good good bad  good good good good good good good bad  good
good
##  [99] good good good good good good good bad  good good good good good
good
## [113] bad  good good good good good bad  good bad  good good good bad
good
## [127] good bad  good good good good good good good good good good good
good
## [141] good bad  good good good bad  good bad  good good good good good
good
## [155] good good bad  good bad  good good good bad  good bad  good bad
good
## [169] good good good good good bad  good good good good bad  good good
bad
## [183] good good bad  good good good good good bad  good good bad  bad
good
## [197] good bad  bad  good bad  good good good bad  good good good bad
bad
## [211] good good good good good good good bad  good good good good good
good
## [225] good good good good good good good good bad  good good good bad
bad
## [239] good good good bad  good good good good good good good good good
good
## [253] bad  good bad  good good good bad  good bad  bad  good good good
bad
## [267] bad  good bad  good good good good good good bad  good good good
good
## [281] good good bad  good good bad  good good good bad  good good good
```

```
good
## [295] bad  good good bad  bad  good bad  good good good good bad  good
good
## [309] good good good good good good good good good good
## Levels: bad good
```

Matriz Confusão

```
matriz_confusao = table(dados_teste$class, previsao)
matriz_confusao

##       previsao
##         bad good
##   bad    46   45
##   good   27  200

#taxa de acerto
taxa = (matriz_confusao[1]+matriz_confusao[4])/sum(matriz_confusao)
taxa

## [1] 0.7735849
```

Deployment

Faz-se a análise a partir de um arquivo externo de possíveis novas buscas por crédito com o modelo produzido:

```
novos_dados = read.csv(file.choose())
previsao_deployment = predict(modelo, novos_dados)
previsao_deployment

## [1] good
## Levels: bad good
```

Análise de Atributos

```
library(e1071)
# Verificando o nível de confiança do modelo sem alterar número de
atributos
modelo_svm = svm(class ~., dados_treino)

previsao_svm = predict(modelo_svm, dados_teste)

m_confusao_svm = table(dados_teste$class, previsao_svm)

taxa_good = (m_confusao_svm[1] + m_confusao_svm[4])/sum(m_confusao_svm)

library(FSelector)

## Warning: package 'FSelector' was built under R version 3.3.3

random.forest.importance(class ~., dados_creditos)
```

```
##                        attr_importance
## checking_status              47.250719
## duration                     28.132648
## credit_history               21.779147
## purpose                      12.949274
## credit_amount                18.165926
## savings_status               14.331658
## employment                    8.086395
## installment_commitment        5.563799
## personal_status               4.877668
## other_parties                10.917738
## residence_since               3.794998
## property_magnitude           10.270490
## age                          10.761404
## other_payment_plans           9.512547
## housing                       6.352225
## existing_credits              5.388015
## job                           4.172244
## num_dependents                2.707471
## own_telephone                 5.207729
## foreign_worker                2.620105
```

```
modelo_atributos = svm(class ~ checking_status + duration +
credit_history, dados_treino)

previsao_atributos = predict(modelo_atributos, dados_teste)

m_confusao_svm_att = table(dados_teste$class, previsao_atributos)

taxa_good_att = (m_confusao_svm_att[1] +
m_confusao_svm_att[4])/sum(m_confusao_svm)
```