Métodos Multivariados de Análise de Dados*

2ª Atividade

Alberson da Silva Miranda

16 de setembro de 2024

*Código disponível em https://github.com/albersonmiranda/analise_multivariada.

Índice

1 INTRODUÇÃO

A atividade consiste na classificação de municípios do Espírito Santo em relação à transparência. O *dataset* é composto por 6 variáveis:

- `ID_PCP`: Índice de divulgação dos Procedimentos Contábeis Patrimoniais;
- `ITGP`: Índice de Transparência e Governança Pública;
- `LEG`: Dimensão Legislativa;
- `PLAT`: Dimensão Plataformas;
- `AG`: Dimensão Administrativo e Governança;
- `TFO`: Dimensão Transparência Financeira Orçamentária;
- `CEP`: Dimensão Comunicação, Engajamento e Participação Social.

E a variável resposta `Classificação` tem os valores 1, 2 e 3, que representam as classificações `Ótimo`, `Regular` e `Ruim`, respectivamente.

2 IMPORTAÇÃO DOS DADOS

```r
# importando dados
dados <- readxl::read_excel(
  "data-raw/analise_discriminante/transparencia.xlsx",
  sheet = "ID_PCP_2021"
)

# corrigindo nome das colunas
dados <- janitor::clean_names(dados)

# corrigindo formatos
dados$classificacao <- factor(
  dados$classificacao,
  levels = 1:3,
  labels = c("Ótimo", "Regular", "Ruim"),
  ordered = TRUE
)

print(dados)
```

```
# A tibble: 78 x 10
   municipio id_pcp  itgp   leg  plat    ag   tfo   cep classificacao habitantes
   <chr>      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <ord>              <dbl>
 1 AFONSO C~   58.1  92.9  93.8  94.7 100    85.1  90.9 Ótimo             30684
 2 AGUA DOC~   54.8  29.1  25    63.2  23.5  34.0   0   Ruim              12042
 3 AGUIA BR~   58.1  71.8  25    89.5  76.5  81.9  86.4 Regular            9711
 4 ALEGRE      64.5  51.9  50    60.5  41.2  51.1  56.8 Regular           29177
 5 ALFREDO ~   61.3  52.2  25    76.3  58.8  55.3  45.5 Ótimo             13836
 6 ALTO RIO~   48.4  40.1   0    84.2  41.2  47.9  27.3 Ruim               7434
 7 ANCHIETA    71.0  86.3  68.8 100    94.1  77.7  90.9 Regular           29984
 8 APIACA      58.1  22.8   0    44.7  11.8  57.4   0   Ótimo              7223
 9 ARACRUZ     80.6  89.0  62.5  94.7  94.1  93.6 100   Regular           94765
10 ATILIO V~   51.6  37.9  25    47.4  35.3  63.8  18.2 Regular           10540
# i 68 more rows
```

3 MODELAGEM

Utilizaremos o *framework* {mlr3} para realizar toda a esteira de modelagem.

```
# metapacote
library(mlr3verse)
```

```
Loading required package: mlr3
```

```
# criação da tarefa
task <- TaskClassif$new(
  id = "transparencia",
  backend = dados[, !names(dados) %in% "municipio"],
  target = "classificacao"
)

# split em treino e teste (80%/20%)
ids <- partition(task, ratio = 0.8)

# pipeline
pipeline <- po(
  # normalização das variáveis numéricas
  "scale",
  # média zero
  center = TRUE,
  # desvio padrão unitário
  scale = TRUE
) %>>%
  # linear discriminant analysis
  po("learner", lrn("classif.lda"))

# convertendo para learner
learner <- as_learner(pipeline)

# treinamento
learner$train(task, row_ids = ids$train)
```

```
Warning in lda.default(x, grouping, ...): variables are collinear
This happened PipeOp classif.lda's $train()
```

```r
# modelo
print(learner$model)
```

```
$scale
$scale$center
         ag         cep  habitantes      id_pcp        itgp         leg
   54.79127    47.02307 50057.72581    61.29032    54.62519    36.08871
       plat         tfo
   71.30730    63.91558

$scale$scale
          ag          cep   habitantes       id_pcp         itgp          leg
   24.114200    27.572472 96734.822613     9.382061    19.904517    28.029079
        plat          tfo
   19.558206    17.230682

$scale$dt_columns
[1] "ag"         "cep"        "habitantes" "id_pcp"     "itgp"
[6] "leg"        "plat"       "tfo"

$scale$affected_cols
[1] "ag"         "cep"        "habitantes" "id_pcp"     "itgp"
[6] "leg"        "plat"       "tfo"

$scale$intasklayout
Key: <id>
           id    type
       <char>  <char>
1:         ag numeric
2:        cep numeric
3: habitantes numeric
4:     id_pcp numeric
5:       itgp numeric
6:        leg numeric
7:       plat numeric
8:        tfo numeric

$scale$outtasklayout
Key: <id>
```

```
           id    type
       <char>  <char>
1:         ag numeric
2:        cep numeric
3: habitantes numeric
4:     id_pcp numeric
5:       itgp numeric
6:        leg numeric
7:       plat numeric
8:        tfo numeric

$scale$outtaskshell
Empty data.table (0 rows and 9 cols): classificacao,ag,cep,habitantes,id_pcp,itgp ...


$classif.lda
$model
Call:
lda(formula, data = task$data())

Prior probabilities of groups:
    Ótimo   Regular      Ruim
0.2096774 0.6290323 0.1612903

Group means:
                ag        cep habitantes      id_pcp       itgp        leg
Ótimo    1.0116134  0.8244517  0.9704202  0.81989532  1.1614774  1.2853321
Regular -0.1173784 -0.1228517 -0.2393207 -0.07934471 -0.1872664 -0.2012193
Ruim    -0.8573216 -0.5926656 -0.3281954 -0.75641955 -0.7795817 -0.8861764
               plat         tfo
Ótimo    0.74253533  1.03985907
Regular -0.26832324 -0.08888274
Ruim     0.08116472 -1.00517412

Coefficients of linear discriminants:
                  LD1        LD2
ag         -0.4801392 -0.5793724
cep         0.6173942 -0.1777774
habitantes -0.2571259  0.6195270
id_pcp     -0.2453124 -0.3523578
itgp       -0.1069328  0.1035361
leg        -0.8816338  0.4431217
plat        0.5119244  1.1581933
```

```
tfo        -0.8057597 -0.5801239

Proportion of trace:
   LD1    LD2
0.8552 0.1448

$log
Empty data.table (0 rows and 3 cols): stage,class,msg

$train_time
[1] 0.012

$param_vals
named list()

$task_hash
[1] "f4bf8ba7054c13fb"

$feature_names
[1] "ag"         "cep"        "habitantes" "id_pcp"     "itgp"
[6] "leg"        "plat"       "tfo"

$validate
NULL

$mlr3_version
[1] '0.20.2'

$data_prototype
Empty data.table (0 rows and 9 cols): classificacao,ag,cep,habitantes,id_pcp,itgp ...

$task_prototype
Empty data.table (0 rows and 9 cols): classificacao,ag,cep,habitantes,id_pcp,itgp ...

$train_task
<TaskClassif:transparencia> (62 x 9)
* Target: classificacao
* Properties: multiclass
* Features (8):
  - dbl (8): ag, cep, habitantes, id_pcp, itgp, leg, plat, tfo

attr(,"class")
[1] "learner_state" "list"
```

```
attr(,"class")
[1] "graph_learner_model" "list"
```

Com o modelo treinado, vamos realizar a predição.

```r
# predição
preds <- learner$predict(task, row_ids = ids$test)
as.data.table(preds)
```

```
    row_ids   truth response
      <int>   <ord>   <fctr>
 1:       1   Ótimo    Ótimo
 2:      17   Ótimo    Ótimo
 3:      63   Ótimo    Ótimo
 4:       2    Ruim     Ruim
 5:      50    Ruim  Regular
 6:      51    Ruim     Ruim
 7:      10 Regular  Regular
 8:      18 Regular  Regular
 9:      21 Regular  Regular
10:      26 Regular  Regular
11:      29 Regular  Regular
12:      32 Regular  Regular
13:      33 Regular  Regular
14:      59 Regular  Regular
15:      61 Regular  Regular
16:      76 Regular  Regular
```

E, por fim, a avaliação do modelo.

```r
# matriz de confusão
confusao <- table(preds$data$truth, preds$data$response)

# acurácia
acuracia <- sum(diag(confusao) / sum(confusao))

# precisão
precisao <- diag(confusao) / colSums(confusao)

# recall
```

```r
recall <- diag(confusao) / rowSums(confusao)

# métricas
confusao
```

```
          Ótimo Regular Ruim
  Ótimo       3       0    0
  Regular     0      10    0
  Ruim        0       1    2
```

```r
acuracia
```

```
[1] 0.9375
```

```r
precisao
```

```
    Ótimo   Regular      Ruim
1.0000000 0.9090909 1.0000000
```

```r
recall
```

```
    Ótimo   Regular      Ruim
1.0000000 1.0000000 0.6666667
```

Como resultado, o modelo se mostrou muito eficiente, com acurácia de 94%, tendo errado apenas uma observação, classificando como `Regular` um município de *actual* `Ruim`. Isso significa uma precisão de 100% nas classes `Ótimo` e `Ruim`, e 91% na classe `Regular`, e recall de 100% nas classes `Ótimo` e `Regular`, e 67% na classe `Ruim`.