

APRENDENDO COM OS DADOS

UMA ABORDAGEM DE CIÊNCIA DE DADOS E APRENDIZADO DE MÁQUINA UTILIZANDO R (PARTE 2)

Prof. Rafael G. Mantovani

04/09/2019



Dois Vizinhos - PR, Brasil
Setembro, 2019

Universidade Tecnológica Federal do Paraná (UTFPR)
IV Semana Acadêmica do curso de Engenharia de Software

Material

Link: https://github.com/rgmantovani/saes2019_dataScience

The screenshot shows the GitHub interface for the repository 'rgmantovani / saes2019_dataScience'. At the top, there are tabs for 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Security', and 'Insights'. Below the tabs, a message states 'No description, website, or topics provided.' with a link to 'Manage topics'. A summary bar indicates '4 commits', '1 branch', and '0 releases'. Below this, there are buttons for 'Branch: master', 'New pull request', and 'Create new file'. The main content area shows a commit by 'rgmantovani' titled 'adding examples'. Below the commit, there is a list of files and folders: 'datasets' (adding config scripts), 'scripts' (adding examples), 'ggplot2-cheatsheet.pdf' (adding cheat sheets), and 'r-cheat-sheet-3.pdf' (adding cheat sheets).

rgmantovani / saes2019_dataScience

<> Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights

No description, website, or topics provided.

[Manage topics](#)

4 commits 1 branch 0 releases

Branch: master New pull request Create new file

rgmantovani adding examples

datasets	adding config scripts
scripts	adding examples
ggplot2-cheatsheet.pdf	adding cheat sheets
r-cheat-sheet-3.pdf	adding cheat sheets

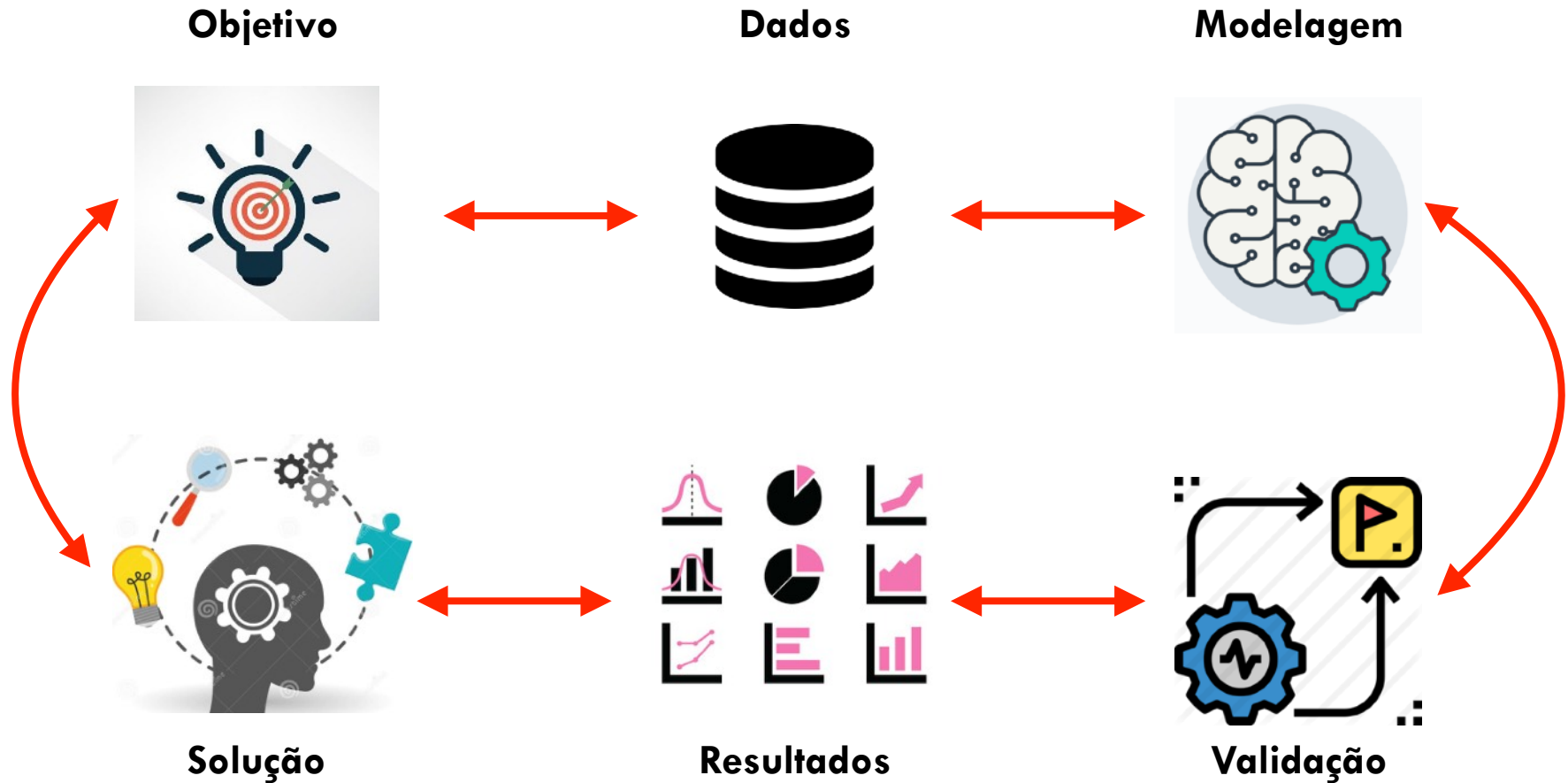
Roteiro

- 1 Revisão
- 2 Visualização dos dados (ggplot2)
- 3 Limpeza dos dados
- 4 Classificação holdout / CV
- 5 Benchmark
- 6 Titanic
- 7 Referências

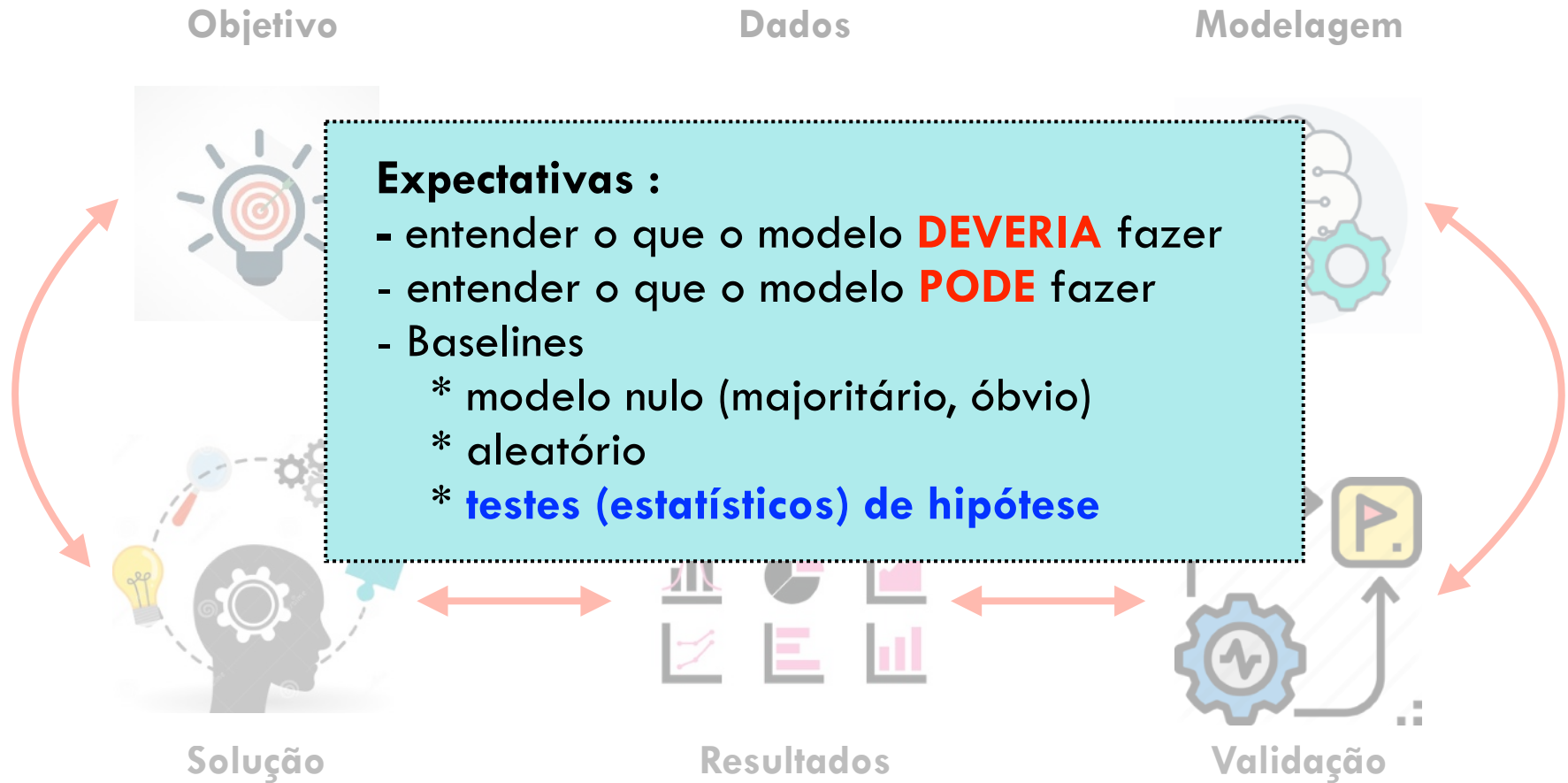
Roteiro

- 1 Revisão**
- 2 Visualização dos dados (ggplot2)**
- 3 Limpeza dos dados**
- 4 Classificação holdout / CV**
- 5 Benchmark**
- 6 Titanic**
- 7 Referências**

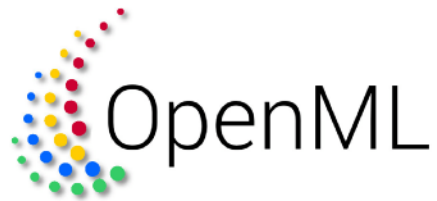
Fluxo de Ciência de Dados



Fluxo de Ciência de Dados



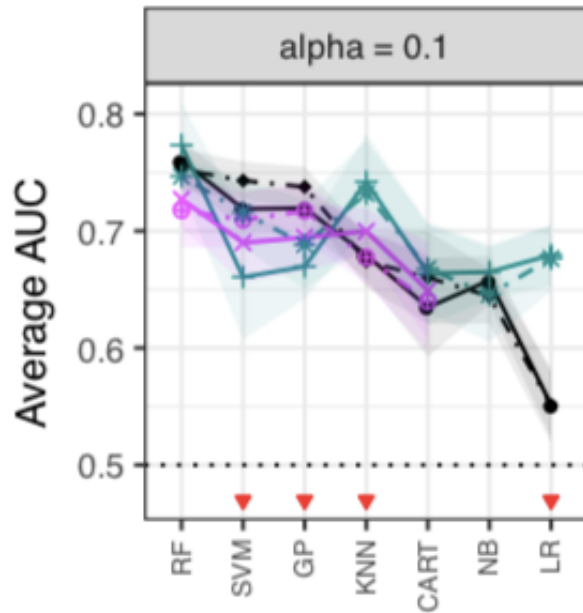
Ferramentas



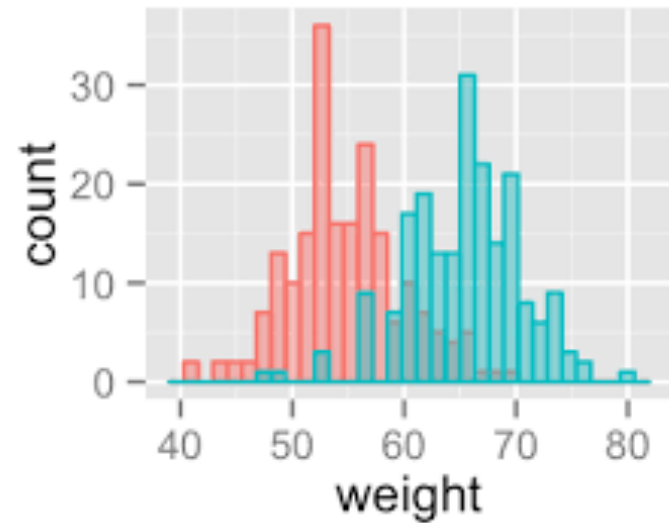
Roteiro

- 1 Revisão
- 2 Visualização dos dados (ggplot2)
- 3 Limpeza dos dados
- 4 Classificação holdout / CV
- 5 Benchmark
- 6 Titanic
- 7 Referências

Visualização de dados

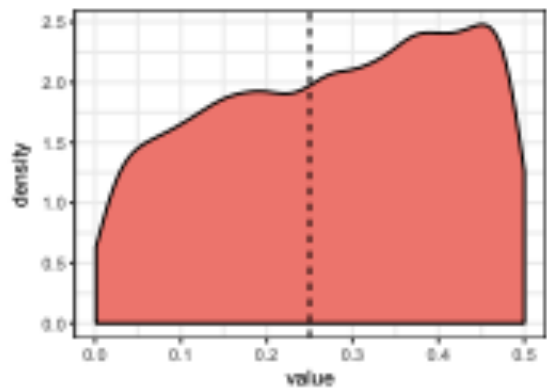


geom_line



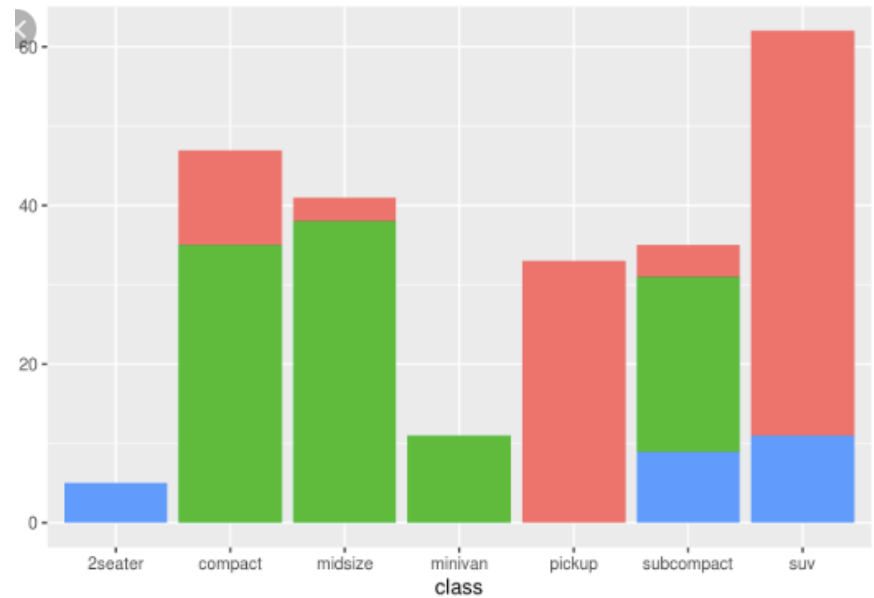
geom_histogram

Visualização de dados



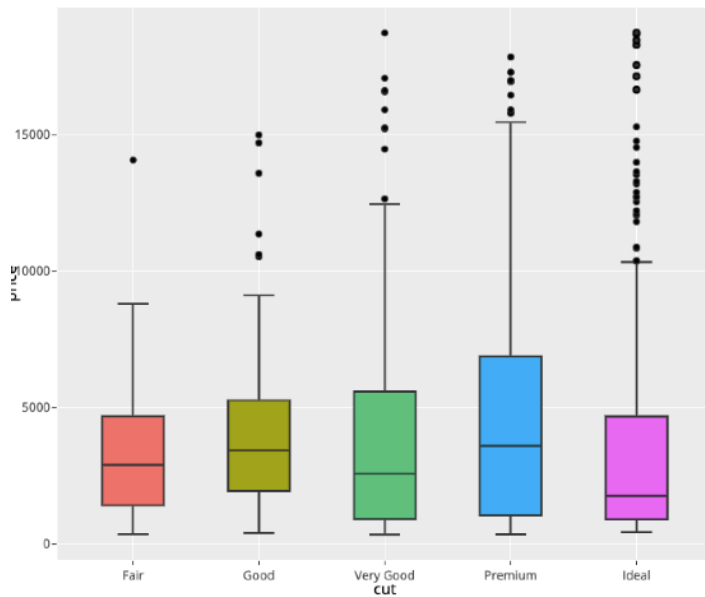
(a) C

geom_density

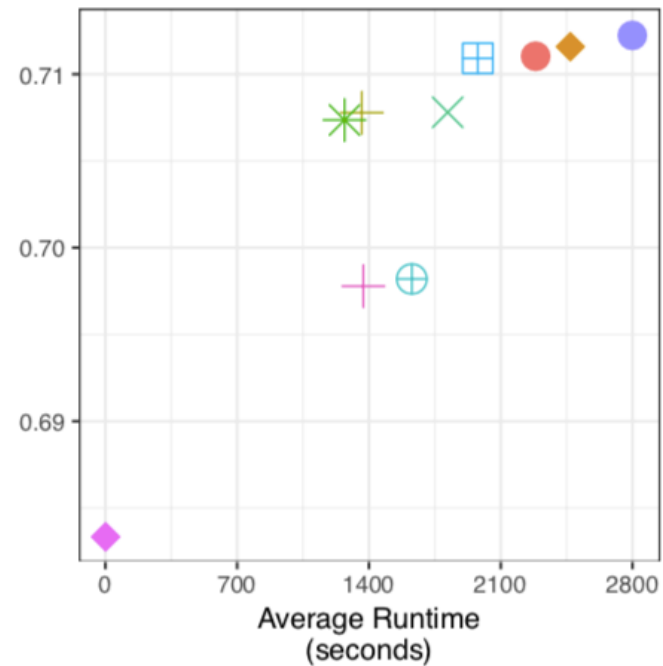


geom_bar

Visualização de dados

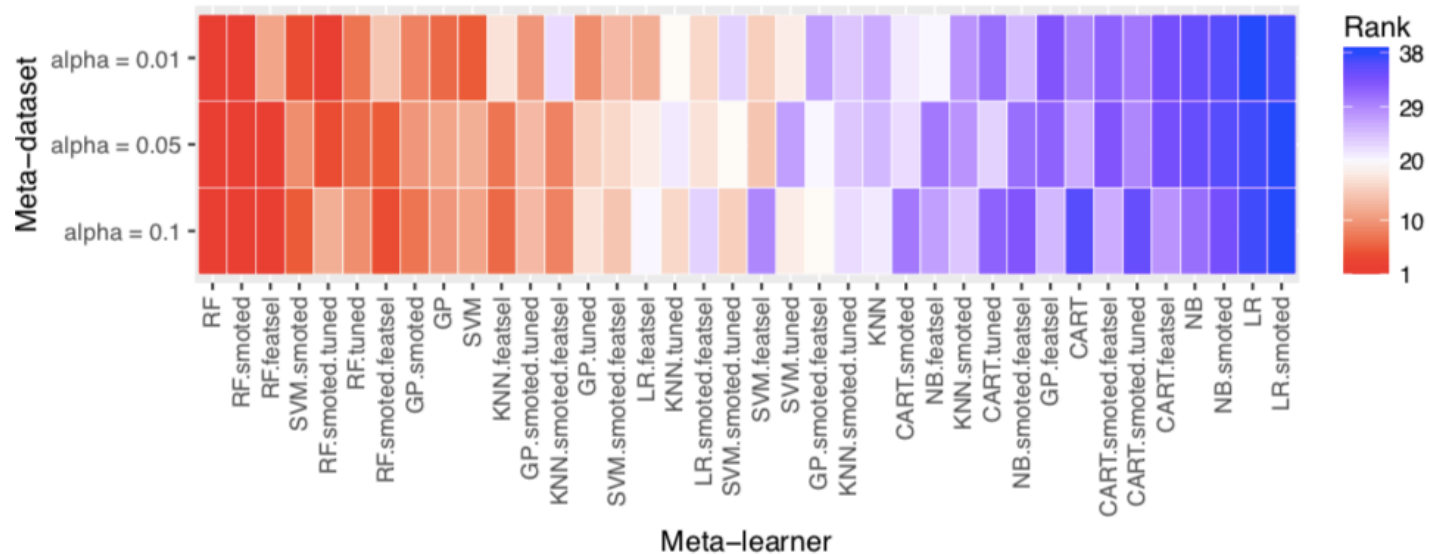


geom_boxplot



geom_point

Visualização de dados



heat maps (geom_tile)

plotting.R

```
1
2 library(ggplot2)
3
4 mpg
5
6 # plotar grafico com x = mpg$displ e y = mpg$hwy
7 ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy))
8
9 # adicionando a cor das classes
10 ggplot(data = mpg) +
11   geom_point(mapping = aes(x = displ, y = hwy, colour = class))
12
13 # tamanho dos pontos
14 ggplot(data = mpg) +
15   geom_point(mapping = aes(x = displ, y = hwy, size = class))
16
17 # tamanho dos pontos + cores
18 ggplot(data = mpg) +
19   geom_point(mapping = aes(x = displ, y = hwy, colour = class, size = class))
20
21 # shape -> forma dos pontos
22 # alpha -> transparencia
```

Roteiro

- 1 Revisão
- 2 Visualização dos dados (ggplot2)
- 3 Limpeza dos dados
- 4 Classificação holdout / CV
- 5 Benchmark
- 6 Titanic
- 7 Referências

cleaning.R

```
6 # carregar os pacotes
7 library(OpenML)      # para acessar e obter o dataset
8 library(mlr)         # para manipularmos os dados
9 library(ggplot2)     # para visualizar o dataset
10
11 # loading dataset usando o pacote OpenML
12 data = getOMLDataSet(data.id = 510)
13
14 # tambem poderia ser
15 # data = OpenML::getOMLDataSet(data.id = 510)
16
17 # o objeto "data" retorna muita informacao, vamos pegar apenas o dataset no slot $data
18 dataset = data$data
19 View(dataset)
20
21 # Verificar os problemas que podemos ter com os dados
22 # 1 - atributos (features) desnecessarios -> ids
23 teste = apply(dataset, 2, unique)
24 lapply(teste, length)
25
26 # 'species' e um atributo do tipo identificador,
27 # ele nao adiciona nada na resolucao do problema
28 ggplot(data = dataset) +
29   geom_point(mapping = aes(x = species, y = overall_danger_index)) +
```

cleaning.R

□ Comandos úteis:

OpenML::getDataSet ()

Pegar dataset do site do OpenML

mlr::impute ()

Imputar dados ausentes

caret::findCorrelation()

Encontrar atributos correlacionados

mlr::normalizeFeatures ()

Normaliza as features

Roteiro

- 1 Revisão
- 2 Visualização dos dados (ggplot2)
- 3 Limpeza dos dados
- 4 Classificação holdout / CV
- 5 Benchmark
- 6 Titanic
- 7 Referências

ex01_holdout.R

```
1 # Carregando os pacotes necessarios para o script funcionar
2 library(OpenML)
3 library(mlr)
4 library(ggplot2)
5 library(reshape2)
6
7 # Pegar o dataset iris do OpenML (id = 61)
8 dataObj = getOMLDataSet(data.id = 61)
9 dataset = dataObj$data
10
11 # Caracteristicas gerais do dataset
12 summary(dataset)
13 colnames(dataset)[ncol(dataset)] = "Species"
14
15 # Plotar a distribuicao de classes
16 ggplot(data = dataset) + geom_bar(aes(x = Species,
17   colour = Species, fill = Species))
18
19 # criar uma taerfa de classificacao
20 task = makeClassifTask(data = dataset, target = "Species")
21 print(task)
```

ex01_holdout.R

□ Comandos úteis:

OpenML::getDataSet ()

Pegar dataset do site do OpenML

mlr::makeClassifTask()

Define uma tarefa preditiva

mlr::makeLearner ()

Define um learner / algoritmo

mlr::makeResampleDesc ()

Define método de avaliação

mlr::resample ()

Rodamos nosso experimento

mlr::getRRPredictions ()

Acessar o valor das predições

Exercício 01

- Replicar o script para outro dataset do OpenML
 - ler os dados via código / baixar o arquivo
 - definir algum algoritmo de classificação
 - ver aqui: https://mlr.mlr-org.com/articles/tutorial/integrated_learners.html#classification-82
 - plotar (predito x valor real)

ex02_CV.R

```
30 # Dividir os dados do dataset em treino e teste, e rodar varias permutacoes
31 rdesc = makeResampleDesc(method = "CV", iters = 10, stratify = TRUE)
32
33 # Medidas de desempenho para avaliar os resultados
34 measures = list(acc, bac)
35
36 # Rodar o algoritmo na tarefa e coletar os resultados
37 result = resample(learner = algo, task = task, resampling = rdesc,
38                  measures = measures, show.info = TRUE)
39 result
40
41 # mostrando o resultado
42 print(result$aggr)
43
44 # Checar as predicoes obtidas pelo algoritmo
45 result$pred
46 pred = getRRPredictions(res = result)
47 print(pred)
48
49 # tabela predicoes
50 table(pred$data[,2:3])
51
```

□ Comandos úteis:

OpenML::getDataSet ()

Pegar dataset do site do OpenML

mlr::makeClassifTask()

Define uma tarefa preditiva

mlr::makeLearner ()

Define um learner / algoritmo

mlr::makeResampleDesc ()

Define método de avaliação

mlr::resample ()

Rodamos nosso experimento

mlr::getRRPredictions ()

Acessar o valor das predições

Exercício 02

- Replicar o script e comparar os resultados com o exercício anterior
 - usar validação cruzada (10 partições /folds)
 - plotar (predito x valor real)

Roteiro

- 1 Revisão
- 2 Visualização dos dados (ggplot2)
- 3 Limpeza dos dados
- 4 Classificação holdout / CV
- 5 Benchmark
- 6 Titanic
- 7 Referências

ex03_benchmarkA.R

```
1 # carregando os pacotes necessarios
2 library(OpenML)
3 library(mlr)
4 library(ggplot2)
5
6 # Obter o dataset do OpenML
7 data_obj = getOMLDataSet(data.id=61)
8 dataset = data_obj$data
9
10 # Criar uma tarefa de classificacao
11 task = makeClassifTask(data = dataset, target = "class")
12
13 # Criar uma lista de algoritmos (learners)
14 lrns = list(
15   makeLearner("classif.lda", id = "lda"),           # LDA - algoritmo linear
16   makeLearner("classif.svm", id = "svm"),           # SVM
17   makeLearner("classif.rpart", id = "rpart"),       # DT - arvore de decisao
18   makeLearner("classif.randomForest", id = "randomForest") # RF - random Forest
19 )
20
```

ex03_benchmarkB.R

```
19
20 # Criar tarefas de classificacao, uma para cada dataset
21 task1 = makeClassifTask(id = data1$desc$name, data = dataset1, target = "class")
22 task2 = makeClassifTask(id = data2$desc$name, data = dataset2, target = "Class")
23 task3 = makeClassifTask(id = data3$desc$name, data = dataset3, target = "Class")
24 task4 = makeClassifTask(id = data4$desc$name, data = dataset4, target = "Class")
25 tasks = list(task1, task2, task3, task4)
26
27 # Criando uma lista de algoritmos (learners)
28 lrns = list(
29   makeLearner("classif.lda", id = "lda"),
30   makeLearner("classif.svm", id = "svm"),
31   makeLearner("classif.rpart", id = "rpart"),
32   makeLearner("classif.randomForest", id = "randomForest")
33 )
34
35 # Criando uma estrategia de validacao
36 # rdesc = makeResampleDesc("RepCV", folds = 10, reps = 10, stratify = TRUE)
37 rdesc = makeResampleDesc("CV", iters = 10, stratify = TRUE)
38
```

ex03_benchmark(A | B).R

□ Comandos úteis:

OpenML::getDataSet ()

Pegar dataset do site do OpenML

mlr::makeClassifTask()

Define uma tarefa preditiva

mlr::makeLearner ()

Define um learner / algoritmo

mlr::makeResampleDesc ()

Define método de avaliação

mlr::benchmark ()

Rodamos nosso experimento com vários algoritmos e tarefas

mlr::plotBMR... ()

Plota os resultados do benchmark

Exercício 03

- Replicar o script:
 - Selecionar vários datasets
 - Definir vários algoritmos
 - usar validação cruzada (10 partições /folds)
 - plotar os resultados

Roteiro

- 1 Revisão
- 2 Visualização dos dados (ggplot2)
- 3 Limpeza dos dados
- 4 Classificação holdout / CV
- 5 Benchmark
- 6 Titanic
- 7 Referências

Desafio :)



Desafio :)



Desafio :)

- <https://www.kaggle.com/c/titanic>
- Titanic (OpenML data id = 40945)
- **Tarefa:**
 - dados ausentes, ruídos, desbalanceamento, etc ...
 - classificação binária
 - quem sobreviveu e quem morreu?
 - quais perfis de pessoas tiveram uma chance maior de sobrevivência?

Desafio :)

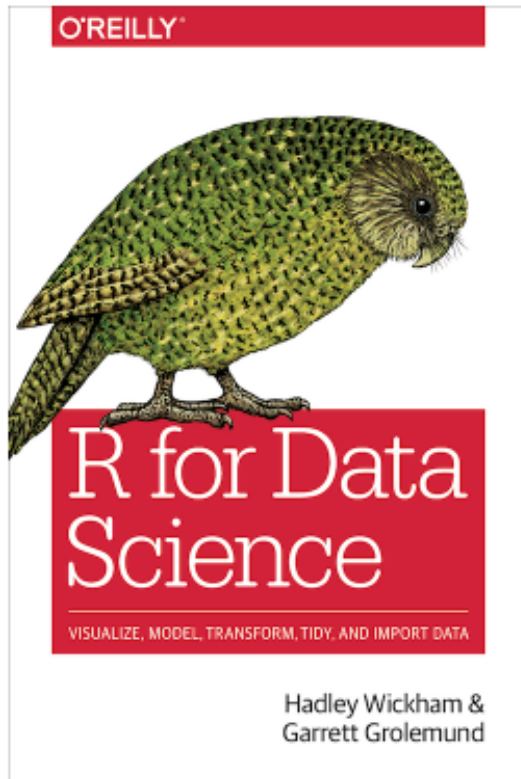
□ Passos:

- 01: ler os dados
- 02: limpar e entender os dados
- 03: definir um algoritmo
- 04: definir um processo de validação
- 05: avaliar o modelo (métricas)
- 06: mostrar os resultados e entender as predições

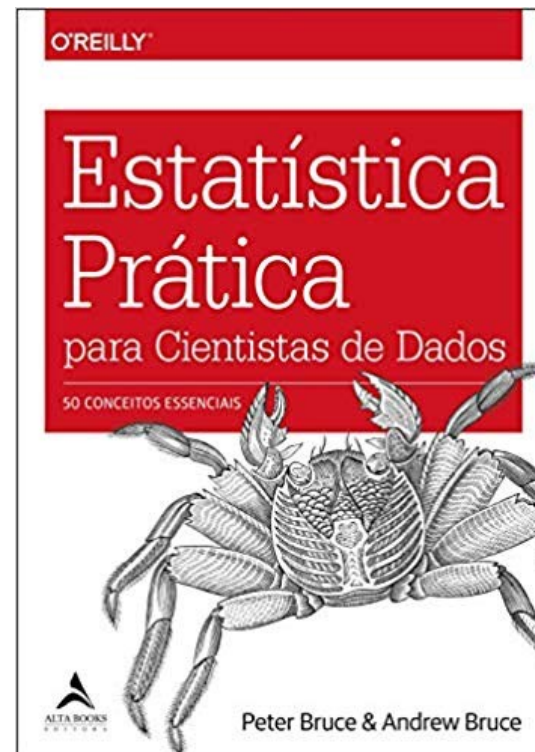
Roteiro

- 1 Revisão
- 2 Visualização dos dados (ggplot2)
- 3 Limpeza dos dados
- 4 Classificação holdout / CV
- 5 Benchmark
- 6 Titanic
- 7 Referências

Referências

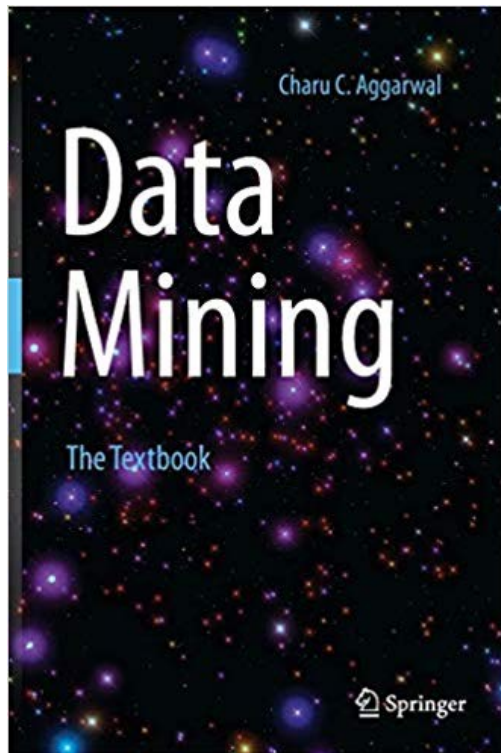


[Wickham & Grolemund, 2018]



[Bruce & Bruce, 2019]

Referências



[Aggarwal, 2015]



[Zumel and Mount, 2014]

Perguntas?

Prof. Rafael G. **Mantovani**

rafaelmantovani@utfpr.edu.br