

Trabalho 1 – Aprendizado de Máquina I

Componentes do grupo

Nome: DYANNA C. SANTOS

matrícula 51352121015

Nome: GUILHERME R. CAMBLOR

matrícula 51352121028

Nome: RENATO G. CAMPOS

matrícula 51352121021

Exercício 1:

Pesquisar dois exemplos de conjuntos de dados, um para um problema de classificação e outro para um problema de regressão, identificando os atributos e o atributo alvo. Explicar cada um dos conjuntos. Inserir nos exemplos as devidas referências bibliográficas.

1. DATASET PARA CLASSIFICAÇÃO

1.1 EXPLICAÇÃO

Este *dataset* contém dados de 802 Pokémons provenientes das sete primeiras gerações. As informações contidas neste conjunto de dados incluem características básicas, como, altura, peso, classificação, pontos de experiência (XPs), habilidades e muito mais.

Para contextualizar esse universo, Pokémon é nada mais do que uma franquia de mídia da “The Pokémon Company”, que foi criada por Satoshi Tajiri em 1995. A companhia é focada na criação de criaturas ficticiais denominadas “Pokémon”, que supostamente os seres humanos podem capturá-los e os treiná-los para lutarem entre si e para evoluir em diversas áreas.

Timeline das gerações:

- Generation I: Kanto (1996–1999)
- Generation II: Johto (1999–2002)
- Generation III: Hoenn (2002–2006)
- Generation IV: Sinnoh (2006–2010)
- Generation V: Unova (2010–2013)
- Generation VI: Kalos (2013–2016)
- Generation VII: Alola (2016–2019) - até onde o dataset abrange
- Generation VIII: Galar (2019–2023)
- Generation IX: Paldea (2022–)

1.2 ATRIBUTOS

- Nome americano: O nome em inglês;

- Nome japonês: O original;
- Número pokedex: A quantidade de Pokémons na pokédex nacional (registros);
- Porcentagem por gênero: Porcentagem de espécies masculinas. Se não tem genero, então em branco;
- Tipos: Pokémon principal (tipo 1) e Pokémon secundário (tipo 2). Cada Pokémon pode pertencer a até dois tipos.

A classificação do Pokémon conforme descrito pela Pokédex do Sol e da Lua.

Figura 1: Atributos do dataset de classificação.

ATRIBUTO	DESCRICAO
altura	Em m
peso	Em kg
capture_rate	Taxa de captura do Pokémon
base_egg_steps	O número de etapas necessárias para chocar um ovo do Pokémon
habilidades	Uma lista restrita de habilidades que o Pokémon é capaz de ter
experience_growth	O crescimento da experiência do Pokémon
base_happiness	Felicidade básica do Pokémon
against_?	Dezoito características que denotam a quantidade de dano sofrido contra um ataque de um tipo específico
hp	O HP base do Pokémon
ataque	O ataque base do Pokémon
defesa	A Defesa Base do Pokémon
sp_attack	O ataque especial básico do Pokémon
sp_defense	A defesa especial básica do Pokémon
speed	A velocidade base do Pokémon
geração	A geração numerada em que o Pokémon foi introduzido pela primeira vez
is_legendary	Indica se o Pokémon é lendário

Fonte: Autores, 2023.

1.4 EXEMPLO

Figura 2: Print do dataset.

3.Aprendizado de Máquina - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

pokemon

Filter

	abilities	against_bug	against_dark	against_dragon	against_electric	against_fairy	against_fight	against_fire	against_flying	against_ground
1	[Overgrow, 'Chlorophyll]	1.00	1.0	1.0	0.5	0.5	0.50	2.0	2.00	
2	[Overgrow, 'Chlorophyll]	1.00	1.0	1.0	0.5	0.5	0.50	2.0	2.00	
3	[Overgrow, 'Chlorophyll]	1.00	1.0	1.0	0.5	0.5	0.50	2.0	2.00	
4	[Blaze, 'Solar Power]	0.50	1.0	1.0	1.0	0.5	1.00	0.5	1.00	
5	[Blaze, 'Solar Power]	0.50	1.0	1.0	1.0	0.5	1.00	0.5	1.00	
6	[Blaze, 'Solar Power]	0.25	1.0	1.0	2.0	0.5	0.50	0.5	1.00	
7	[Torrent, 'Rain Dish]	1.00	1.0	1.0	2.0	1.0	1.00	0.5	1.00	
8	[Torrent, 'Rain Dish]	1.00	1.0	1.0	2.0	1.0	1.00	0.5	1.00	
9	[Torrent, 'Rain Dish]	1.00	1.0	1.0	2.0	1.0	1.00	0.5	1.00	
10	[Shield Dust, 'Run Away]	1.00	1.0	1.0	1.0	1.0	0.50	2.0	2.00	
11	[Shed Skin]	1.00	1.0	1.0	1.0	1.0	0.50	2.0	2.00	

Showing 1 to 11 of 801 entries, 41 total columns

```
R 4.3.1 - C:/Users/dyann/OneDrive/DYANNA/D4_FATEC/SEMESTRES/5º SEMESTRE/3.Aprendizado de Máquina/
> library(readr)
> pokemon <- read_csv("ATIVIDADES - R/pokemon.csv")
Rows: 801 Columns: 41
Column specification:
 Delimiter: ","
chr (7): abilities, capture_rate, classification, japanese_name, name, type1, type2
dbl (34): against_bug, against_dark, against_dragon, against_electric, against_fairy, against_fight, against_fire, against_flying, against...
```

i use 'spec()' to retrieve the full column specification for this data.
i specify the column types or set 'show_col_types = FALSE' to quiet this message.

```
> View(pokemon)
```

Fonte: Autores, 2023.

Figura 3: Exemplo de knn utilizando o dataset.

```
> classificacao <- classificacao$type1
> resultado <- knn(treino, teste, classificacao, 3)
> resultado
[1] normal water  water  normal water  normal
Levels: bug normal water
> |
```

Fonte: Autores, 2023.

Algumas questões podem ser levantadas através das análises, como por exemplo: é possível construir um classificador para identificar Pokémons lendários? Como a altura e o peso de um Pokémon estão correlacionados com suas estatísticas básicas? Quais fatores exercem influência no crescimento da experiência? Qual é o tipo mais forte no geral? E o mais fraco? Qual tipo tem maior chance de ser um Pokémon lendário? É possível construir um time dos sonhos? Uma equipe de 6 Pokémons que causa mais dano enquanto permanece imune a outra equipe de mesma dimensão.

2. DATASET PARA REGRESSÃO

2.1 EXPLICAÇÃO

Este *dataset* contém dados da movimentação de carga do Porto de Santos, no período de 2005 a 2022, sendo publicado mensalmente no site da Autoridade Portuária de Santos (APS).

2.3 ATRIBUTOS

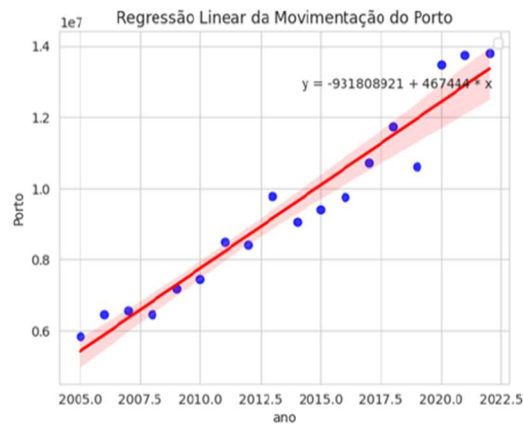
Figura 4: atributos do *dataset* de regressão.

REGRESSAO	
ATRIBUTO	DESCRIÇÃO
Ano	2005 - 2022 +
Mês	janeiro a dezembro
Tipo de instalação	torto Organizado
Terminal	TGG, Termag, DP...
Perfil de carga	tipo mais generico: granel sólido, carga containerizada...
Tipo de Operação	convencional, remoção...
Navegação	longo curso e cabotagem
Sentido de carga	embarque, desembarque e movimento a bordo
Carga	tipo, exemplo "grãos e farelo"...
Toneladas	unidade de medida
TEUs	unidade de medida
Unidades	unidade ligada aos TEUs

Fonte: Autores, 2023.

Diante dos atributos identificados, é possível observar a possibilidade de correlacionar as colunas encontradas, como no exemplo abaixo.

Figura 5: exemplo de regressão.



Fonte: Autores, 2023.

2.4 EXEMPLO

Figura 6: print do *dataset*.

3.Aprendizado de Máquina - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

mensario

	Ano	Mês	Berço	Tipo da instalação	Terminal	Perfil da carga	Tipo de operação	Navegação	Sentido da carga	Carga	Toneladas	TEUs	Unidade
1	2006	08	SABOO 3	PORTO ORGANIZADO	RODRIMAR	CARGA CONTEINERIZADA	CONVENCIONAL	LONGO CURSO	DESEMBARQUE	ÓLEO DE ORIGEM VEGETAL	132.714	6	
2	2006	08	TECON 1	PORTO ORGANIZADO	SANTOS BRASIL	CARGA CONTEINERIZADA	REMOÇÃO	LONGO CURSO	EMBARQUE	SEM CARGAS	423.970	202	
3	2006	08	CORTE	PORTO ORGANIZADO	ECOPORTO	CARGA CONTEINERIZADA	CONVENCIONAL	LONGO CURSO	EMBARQUE	OUTRAS MERCADORIAS	77545.352	5376	
4	2006	08	ARM 35.2	PORTO ORGANIZADO	LIBRA 35	CARGA CONTEINERIZADA	CONVENCIONAL	LONGO CURSO	DESEMBARQUE	OUTRAS MERCADORIAS	519.554	33	
5	2006	08	ARM 35.2	PORTO ORGANIZADO	LIBRA 35	CARGA CONTEINERIZADA	CONVENCIONAL	LONGO CURSO	DESEMBARQUE	CARNES	27.460	2	
6	2006	08	ARM 37	PORTO ORGANIZADO	LIBRA 35	CARGA CONTEINERIZADA	REMOÇÃO	CABOTAGEM	DESEMBARQUE	OUTRAS MERCADORIAS	6277.912	535	
7	2006	08	ARM 37	PORTO ORGANIZADO	LIBRA 35	CARGA CONTEINERIZADA	REMOÇÃO	CABOTAGEM	EMBARQUE	OUTRAS MERCADORIAS	6277.912	535	
8	2006	08	TEV	PORTO ORGANIZADO	SANTOS BRASIL	CARGA CONTEINERIZADA	TRANSBORDO	CABOTAGEM	DESEMBARQUE	OUTRAS MERCADORIAS	1939.318	132	
9	2006	08	SABOO 4	PORTO ORGANIZADO	ECOPORTO	CARGA CONTEINERIZADA	CONVENCIONAL	LONGO CURSO	EMBARQUE	OUTRAS MERCADORIAS	34134.888	2578	
10	2006	08	SABOO 4	PORTO ORGANIZADO	ECOPORTO	CARGA CONTEINERIZADA	REMOÇÃO	LONGO CURSO	EMBARQUE	OUTRAS MERCADORIAS	2321.308	182	
11	2006	08	SABOO 4	PORTO ORGANIZADO	ECOPORTO	CARGA CONTEINERIZADA	CONVENCIONAL	LONGO CURSO	EMBARQUE	CARNES	94.813	6	
12	2006	08	TECON 1	PORTO ORGANIZADO	SANTOS BRASIL	CARGA CONTEINERIZADA	CONVENCIONAL	LONGO CURSO	DESEMBARQUE	OUTRAS MERCADORIAS	33783.973	2869	

Showing 1 to 12 of 291,296 entries, 13 total columns

```
R 4.3.1 C:/Users/dyann/OneDrive/DYANNA/04_FATEC/SEMESTRES/5º SEMESTRE/3.Aprendizado de Máquina/
> library(readr)
> mensario <- read_csv("ATIVIDADES - R/mensario.csv")
Rows: 291296 Columns: 13
Column specification:
Delimiter: ","
chr (9): Mês, Berço, Tipo da instalação, Terminal, Perfil da carga, Tipo de operação, Navegação, Sentido da carga, Carga
dbl (4): Ano, Toneladas, TEUs, Unidades
i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
> View(mensario)
```

Fonte: Autores, 2023.

Algumas questões podem ser analisadas, tais como: Qual é a movimentação do Porto de Santos? Quais são os principais tipos de carga movimentados e que representam? Como se distribui a movimentação de carga entre os diferentes berços?

REFERÊNCIAS

<https://www.portodesantos.com.br/informacoes-operacionais/estatisticas/mensario-estatistico/tabela-dinamica/>

<https://www.kaggle.com/datasets/rounakbanik/pokemon?resource=download>

<https://pokemon.fandom.com/pt-br/wiki/Tipo>

<https://twitter.com/pokemongobra/status/752570109391171584?lang=el>

Exercício 2:

- Vizinhos mais próximos – k-nn

Estruturar uma tabela com três atributos em que as instâncias possam ser classificadas em duas classes.

Aplicar o algoritmo k-nn ao conjunto de dados, variando o parâmetro k.

Demonstrar os resultados e justificar em função da variação de k.

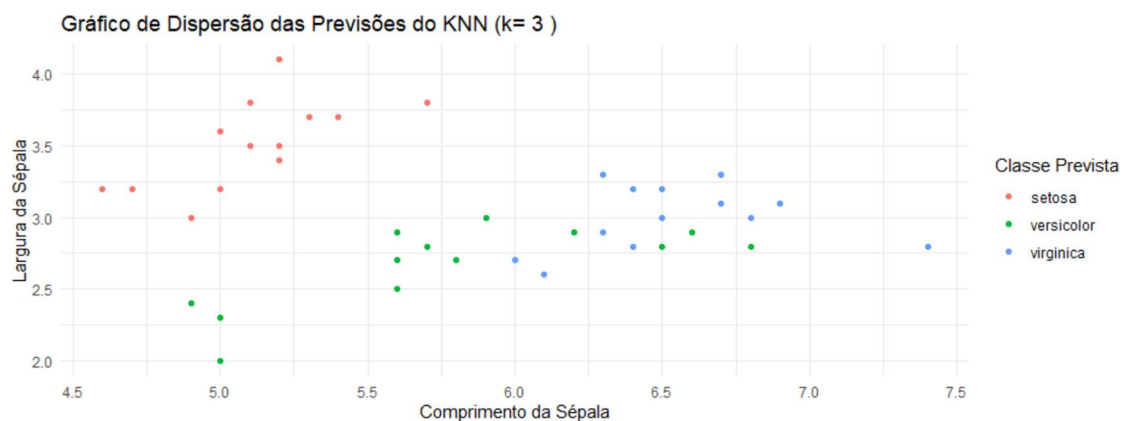
SCRIPT

```
library(class)
library(ggplot2)

set.seed(123)
indices <- sample(1:nrow(iris), nrow(iris) * 0.7)
treinamento <- iris[indices, ]
teste <- iris[-indices, ]
k_values <- c(1, 3, 6, 7)
plotKNN <- function(k_values) {
  for (k in k_values) {
    resultado_knn <- knn(treinamento[, 1:4], teste[, 1:4], treinamento$Species, k)
    teste$Previsto_knn <- resultado_knn
    p <- ggplot(teste, aes(x = Sepal.Length, y = Sepal.Width, color = Previsto)) +
      geom_point() +
      labs(title = paste("Gráfico de Dispersão das Previsões do KNN (k=", k, ")"),
           x = "Comprimento da Sépala",
           y = "Largura da Sépala") +
      scale_color_discrete(name = "Classe Prevista") +
      theme_minimal()
    print(p)
  }
}

plotknn(K_VALUES)
```

Resultado plotado no gráfico:



Exercício 3:

1. Fazer o modelo de Naïve Bayes para a tabela “planejar-passeio”;
2. Criar mais 5 eventos teste relacionados à tabela, fazer os cálculos de probabilidade relacionados a cada atributo, estimar as respectivas classes com o modelo e comparar os resultados;
3. Inserir a “matriz de confusão” para estimar o modelo;

```
library(e1071)
library(readxl)
library(ggplot2)
library(caret)
caminho_arquivo <- "C:\\Users\\Guilherme\\Downloads\\Temperatura.xlsx"

abas_disponiveis <- excel_sheets(caminho_arquivo)

indice_abas <- 1

dados <- read_excel(caminho_arquivo, sheet = abas_disponiveis[indice_abas])

amostra = sample(2,14,replace=T,prob=c(0.7,0.3))

creditotreino = dados[which(amostra == 1), ]
creditoteste = dados[which(amostra == 2), ]
dim(creditotreino)
10 5
creditoteste
previsao temperatura umidade vento passeio
<chr> <chr> <chr> <chr> <chr>
1 sol quente alta sim orquidário
2 sol moderado alta não orquidário
3 nublado frio normal sim aquário
4 sol moderado normal sim aquário
modelo <- naiveBayes(passeio ~., data = creditotreino)
modelo
```

Naive Bayes Classifier for Discrete Predictors

Call:

```
naiveBayes.default(x = X, y = Y, laplace = laplace)
```

A-priori probabilities:

Y

	aquário	orquidário
	0.7	0.3

Conditional probabilities:

		previsao		
Y		chuva	nublado	sol
	aquário	0.4285714	0.4285714	0.1428571
	orquidário	0.6666667	0.0000000	0.3333333

		temperatura		
Y		frio	moderado	quente
	aquário	0.2857143	0.4285714	0.2857143
	orquidário	0.3333333	0.3333333	0.3333333

		umidade	
Y		alta	normal
	aquário	0.4285714	0.5714286
	orquidário	0.6666667	0.3333333

		vento	
Y		não	sim
	aquário	0.8571429	0.1428571
	orquidário	0.3333333	0.6666667

```
predicao <- predict(modelo, creditoteste)
```

```
confusao = table(creditoteste$passeio, predicao)
```

```
taxaacerto = ((confusao[1] + confusao[4]) / sum(confusao))
```

```
taxaerro = ((confusao[2] + confusao[3]) / sum(confusao))
```

```
taxaacerto
```

```
0.5
```

```
primeiroCenario <- data.frame(previsao="sol", temperatura="quente", umid  
ade="normal", vento="não")  
segundoCenario <- data.frame(previsao="chuva", temperatura="quente", umi  
dade="normal", vento="sim")  
terceiroCenario <- data.frame(previsao="sol", temperatura="moderado", um  
idade="alta", vento="sim")  
quartoCenario <- data.frame(previsao="sol", temperatura="frio", umidade=  
"normal", vento="sim")  
quintoCenario <- data.frame(previsao="chuva", temperatura="frio", umidad  
e="alta", vento="não")
```



```

predicao1 <- predict(modelo, primeiroCenario)
predicao2 <- predict(modelo, segundoCenario)
predicao3 <- predict(modelo, terceiroCenario)
predicao4 <- predict(modelo, quartoCenario)
predicao5 <- predict(modelo, quintoCenario)

valor1 <- as.character(predicao1)
valor2 <- as.character(predicao2)
valor3 <- as.character(predicao3)
valor4 <- as.character(predicao4)
valor5 <- as.character(predicao5)

primeiroCenario$passeio <- valor1
segundoCenario$passeio <- valor2
terceiroCenario$passeio <- valor3
quartoCenario$passeio <- valor4
quintoCenario$passeio <- valor5

creditoteste <- rbind(creditoteste, primeiroCenario)
creditoteste <- rbind(creditoteste, segundoCenario)
creditoteste <- rbind(creditoteste, terceiroCenario)
creditoteste <- rbind(creditoteste, quartoCenario)
creditoteste <- rbind(creditoteste, quintoCenario)
creditoteste
  previsao temperatura umidade vento passeio
  <chr>      <chr>      <chr>  <chr> <chr>
1 sol       quente     alta   sim  orquidário
2 sol       moderado   alta   não  orquidário
3 nublado   frio       normal  sim  aquário
4 sol       moderado   normal  sim  aquário
5 sol       quente     normal  não  aquário
6 chuva     quente     normal  sim  orquidário
7 sol       moderado   alta   sim  orquidário
8 sol       frio       normal  sim  orquidário
9 chuva     frio       alta   não  aquário

predicao2 <- predict(modelo, creditoteste)
confusao2 = table(creditoteste$passeio, predicao2)
predicao2
orquidário aquário  aquário  orquidário aquário  orquidário orquid
ário orquidário aquário
Levels: aquário orquidário
confusao2
      predicao2
      aquário orquidário
aquário      3          1
orquidário    1          4
taxaacerto2 = ((confusao2[1] + confusao2[4]) / sum(confusao2))
taxaerro2 = ((confusao2[2] + confusao2[3]) / sum(confusao2))
taxaacerto2
0.7777778

```

Exercício 3:

4. Fazer o exemplo Naïve Bayes da tabela “credit-g.arff” contido no arquivo “Preparação de Dados de treinamento e teste - Naïve-Bayes”.

modelo

Naive Bayes Classifier for Discrete Predictors

Call:

naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:

Y

	bad	good
Y	0.3057971	0.6942029

Conditional probabilities:

checking_status

Y	<0	>=200	0<=X<200	no checking
bad	0.46919431	0.04739336	0.33175355	0.15165877
good	0.19415449	0.06680585	0.24008351	0.49895616

duration

Y	[,1]	[,2]
bad	24.56398	13.23557
good	19.07098	11.05990

credit_history

Y	all paid	critical/other existing credit delayed previously existing paid no credits/all paid			
bad	0.09952607		0.18483412	0.09004739	0
	.54502370	0.08056872			
good	0.03340292		0.31941545	0.10020877	0
	.52192067	0.02505219			

purpose

Y	business	domestic appliance	education	furniture/equipment	
new car					
bad	0.109004739	0.018957346	0.071090047	0.213270142	0.
	293838863	0.004739336	0.189573460	0.033175355	0.004739336
good	0.093945720	0.010438413	0.048016701	0.169102296	0.
	212943633	0.006263048	0.306889353	0.018789144	0.014613779
					0.118997912

credit_amount

Y	[,1]	[,2]
bad	3673.882	3169.021
good	2950.457	2439.872

savings_status

Y	<100	>=1000	100<=X<500	500<=X<1000	no known savings
bad	0.74881517	0.01895735	0.10426540	0.02843602	0.09952607
good	0.55114823	0.06889353	0.09812109	0.07098121	0.21085595

employment

Y	<1	>=7	1<=X<4	4<=X<7	unemployed
---	----	-----	--------	--------	------------

	bad	0.22274882	0.21327014	0.34597156	0.13744076	0.08056872
	good	0.15240084	0.25052192	0.34655532	0.20250522	0.04801670

	installment_commitment	
Y	[,1]	[,2]
	bad	3.127962 1.081324
	good	2.870564 1.142535

	personal_status							
Y	female	div/dep/mar	male	div/sep	male	mar/wid	male	single
	bad	0.35545024	0.08530806	0.08056872	0.47867299			
	good	0.31315240	0.04384134	0.10020877	0.54279749			

	other_parties		
Y	co applicant	guarantor	none
	bad	0.05213270	0.01895735 0.92890995
	good	0.03131524	0.04592902 0.92275574

	residence_since	
Y	[,1]	[,2]
	bad	2.900474 1.118907
	good	2.824635 1.097038

	property_magnitude				
Y	car	life	insurance	no known property	real estate
	bad	0.3364929	0.2322275	0.2227488	0.2085308
	good	0.3402923	0.2275574	0.1022965	0.3298539

	age	
Y	[,1]	[,2]
	bad	34.29858 11.14155
	good	36.01670 11.50094

	other_payment_plans		
Y	bank	none	stores
	bad	0.18009479 0.74881517	0.07109005
	good	0.12526096 0.83924843	0.03549061

	housing		
Y	for free	own	rent
	bad	0.15165877 0.59715640	0.25118483
	good	0.07724426 0.75365344	0.16910230

	existing_credits	
Y	[,1]	[,2]
	bad	1.398104 0.5958557
	good	1.405010 0.5553618

	job			
Y	high qualif/self	emp/mgmt	skilled unemp/unskilled	non res unsk
	illed resident			
	bad	0.14691943	0.63507109	0.02369668
	0.19431280			
	good	0.12317328	0.63883090	0.01878914
	0.21920668			

	num_dependents	
Y	[,1]	[,2]
	bad	1.156398 0.3640962
	good	1.160752 0.3676857

```

      own_telephone
Y      none      yes
bad  0.6682464 0.3317536
good 0.6075157 0.3924843

      foreign_worker
Y      no      yes
bad  0.01421801 0.98578199
good 0.04592902 0.95407098

head(predicao)
good bad bad good good
Levels: bad good
confusao
      predicao
      bad good
bad    50   39
good   38  183
taxaacerto
0.7516129
taxaerro
0.2483871

```

Exercício 4:

→ Criar uma tabela de classificação utilizando 3 atributos e duas classificações; fazer a classificação utilizando [árvore de decisão com Rpart](#)

```

library("rpart.plot")
library(rpart)
set.seed(123) # Para tornar os resultados reproduzíveis
idade <- sample(18:60, 20, replace = TRUE)

renda <- sample(3000:12000, 20, replace = TRUE)

historico_credito <- sample(c("Bom", "Ruim"), 20, replace = TRUE)

empregado <- sample(c("Sim", "Não"), 20, replace = TRUE)

estado_civil <- sample(c("Solteiro", "Casado", "Divorciado"), 20, replace = TRUE)

elegibilidade <- sample(c("Elegível", "Não Elegível"), 20, replace = TRUE)

dados <- data.frame(
  Idade = idade,
  Renda = renda,
  Historico_de_Credito = historico_credito,
  Empregado = empregado,
  Estado_Civil = estado_civil,
  Elegibilidade = elegibilidade
)

```

```

modelo_arvore <- rpart(Elegibilidade ~ Idade + Renda + Historico_de_Credito + Empregado + Estado_Civil, data = dados, method = "class", control = rpart.control(minsplit = 1), parms = list(split = 'Information'))

modelo_arvore
n= 20

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 20 5 Não Elegível (0.2500000 0.7500000)
  2) Idade>=42.5 11 5 Não Elegível (0.4545455 0.5454545)
    4) Idade< 43.5 2 0 Elegível (1.0000000 0.0000000) *
    5) Idade>=43.5 9 3 Não Elegível (0.3333333 0.6666667)
      10) Empregado=Não 5 2 Elegível (0.6000000 0.4000000)
        20) Estado_Civil=Casado,Solteiro 3 0 Elegível (1.0000000 0.0000000)
      00) *
        21) Estado_Civil=Divorciado 2 0 Não Elegível (0.0000000 1.0000000)
      0) *
        11) Empregado=Sim 4 0 Não Elegível (0.0000000 1.0000000) *
      3) Idade< 42.5 9 0 Não Elegível (0.0000000 1.0000000) *

plota_arvore <- rpart.plot(modelo_arvore, 1, 4)

predicao <- predict(modelo_arvore)
predicao
  Elegível  Não Elegível
1         0             1
2         0             1
3         0             1
4         0             1
5         0             1
6         1             0
7         0             1
8         0             1
9         0             1
10        1             0
11        0             1
12        0             1
13        0             1
14        1             0
15        0             1
16        0             1
17        1             0
18        0             1
19        1             0
20        0             1

```

Resultado do gráfico no formato em árvore:

