

Código

```
1 library(faraway)
2 library(visdat)
3 library(corrplot)
4 library(MASS)
5 library(class)
6 library(dplyr)
7 library(ISLR)
8 library(leaps)
9 install.packages("caret")
10 library(caret)
11
12 ?diabetes
13 vis_miss(diabetes[,-1])
14 ## Retirar o id, visto n ser importante
15 diabetes
16 nossoDiabetes <- diabetes[,-1]
17 nossoDiabetes
18
19 ## Gerar a variável binária que indica se tem ou não diabetes
20 diabetesB <- rep(NA,403)
21 diabetesB[nossoDiabetes$glyhb > 7]=1
22 diabetesB[nossoDiabetes$glyhb <= 7] = 0
23 # por como fator (é qualitativa)
24 diabetesB <- factor(diabetesB)
25 summary(diabetesB)
26
27 ##Adicionar a variavel ao dataset
28 nossoDiabetes$diabetesB <- diabetesB
29
30 ##Ordenar
31 nossoDiabetes <- nossoDiabetes[,c(19,5,1,2,3,4,6,7,8,9,10,11,12,13,16,17,18)]
32 summary(nossoDiabetes)
33
34 ##Visualizar as correlações
35 cor(nossoDiabetes[,c(1,7,9,12)],use = "complete.obs")[1,] #apenas da hemoglobina glicada
36 rel = cor(nossoDiabetes[,c(1,7,9,12)],use = "complete.obs")
37 corrplot(rel, type="upper", tl.col = "black",sig.level = 0.01)
```

Análise Exploratória

Análise à influência

```
1 #-----#
2 # Teste à influência /
3 #-----#
4
5 ##Gerar os gráficos das variáveis em função das outras
6 pairs(nossoDiabetes[,c(1:6)])
7 pairs(nossoDiabetes[,c(1:2,7:10)])
8 pairs(nossoDiabetes[,c(1:2,11:14)])
9 pairs(nossoDiabetes[,c(1:2,15:19)])
10
11
12 ### Variável Binária:
13 par(mfrow=c(1,1))
14
15 ### ANLISE COLESTROL ##
```

```

16 colestrol <- diabetes$chol
17 table(diabetesB)
18 table(diabetesB,colestrol)
19 table(diabetesB,colestrol)
20 boxplot(table(diabetesB,colestrol))
21 boxplot(colestrol~diabetesB)
22 barplot(colestrol~diabetesB) ##Pode ter influencia maior o colestrol diabetes bitch, mediana ligeiramente a
    cima
23 table(diabetesB)
24 fivenum(colestrol)
25 #Min: 78, Max:443, Media 179, 25%->78 75%->230
26
27 ### ANLISE STAB-GLU ##
28 #FIVENUM()
29 stab = nossoDiabetes$stab.glu
30 fivenum(stab)
31 #Min: 48, Max:385, Media 89, 25%->81 75%->106
32 table(diabetesB,stab)
33 boxplot(stab)
34 boxplot(stab~diabetesB) #STAB-GLU CLARAMENTE AFETA
35
36 ##ANALISE RACIO ##
37 racio <- diabetes$ratio
38 summary(racio)
39 boxplot(racio~diabetesB) ## Parece ter algum
40
41 ### ANLISE HDL ##
42 hdl <- diabetes$hdl
43 table(diabetesB,hdl)
44 fivenum(hdl)
45 #Min: 12, Max:120, Media 46, 25%->38 75%->59
46 boxplot(hdl)
47 boxplot(hdl~diabetesB) ##Mais baixo HDL menor os "diabetes"
48
49 ### ANLISE location ##
50 loc <- diabetes$location
51 table(diabetesB,loc)
52 barplot(table(diabetesB,loc),beside=T, legend.text=c("Nao_ter","Ter")) ## Nao parece ter influencia nenhuma
    a localizacao
53
54 ##ANALISE AGE
55 age <- diabetes$age
56 summary(age)
57 table(diabetesB,age)
58 barplot(table(diabetesB,age))
59 boxplot(age)
60 boxplot(age~diabetesB)
61
62
63 # Age
64 boxplot(nossoDiabetes$age~nossoDiabetes$diabetesB) ## parece afetar um pouco
65 # Height
66 boxplot(nossoDiabetes$height~nossoDiabetes$diabetesB) ## não parece afetar praticamente nada
67 # Weight
68 boxplot(nossoDiabetes$weight~nossoDiabetes$diabetesB) ## parece afetar um pouco (menos que
69 ## a primeira)
70
71 # Gender
72 table(nossoDiabetes$diabetesB, nossoDiabetes$gender) ## Na tabela podemos verificar que 18.2% dos

```

```

73 26/162 #1 em homens ## homens do dataset possuem diabetes. Já nas
74 34/228 #1 em mulheres ## mulheres a percentagem é de 17%. Tendo isto
75
76 136/(136+194) #homens em 0 ## conta o género parece não afetar
77 26/(26+34) #homens em 1
78
79 # Frame
80 table(nossoDiabetes$diabetesB, nossoDiabetes$frame) ## Na tabela podemos verificar que quanto maior
81 9/102 #1 em small ## é o frame, mais incidências de diabetes
82 26/178 #1 em medium ## são registadas. Parece portanto afetar
83 23/99 #1 em large
84
85 93/(93+152+76) #small em 0
86 152/(93+152+76) #medium em 0
87
88 9/(9+26+23) #small em 1
89 26/(9+26+23) #medium em 1
90 par(mfrow=c(1,1))
91
92 summary(nossoDiabetes$diabetesB)
93 boxplot(nossoDiabetes$time.ppn~nossoDiabetes$diabetesB) ##não parece afetar a
94 ##variavel binaria
95 boxplot(nossoDiabetes$waist~nossoDiabetes$diabetesB) #parece afetar ligeiramente
96 boxplot(nossoDiabetes$hip~nossoDiabetes$diabetesB) ##parece afetar ligeiramente
97 boxplot(nossoDiabetes$bp.1s~nossoDiabetes$diabetesB) ##parece afetar mais, mas
98 ## nao muito
99 boxplot(nossoDiabetes$bp.1d~nossoDiabetes$diabetesB) ##parece afetar ligeiramente (mas
100 ## parece afetar mais do que a primeira, menos do que as restantes)
101
102 ### Variável glyhb:
103 # Gender
104 boxplot(nossoDiabetes$glyhb~nossoDiabetes$gender) ## não parece afetar praticamente nada
105
106 # Frame
107 boxplot(nossoDiabetes$glyhb~nossoDiabetes$frame) ## não parece afetar praticamente nada
108
109 # Location
110 boxplot(nossoDiabetes$glyhb~nossoDiabetes$location) ## não parece afetar praticamente nada
111
112 plot(nossoDiabetes$glyhb ~ nossoDiabetes$time.ppn, type = "p") ##parece n afetar
113 abline(lm(nossoDiabetes$glyhb ~ nossoDiabetes$time.ppn)) ##tendencia constante
114 plot(nossoDiabetes$glyhb ~ nossoDiabetes$waist, type = "p") ##difícil de
115 ##analisar este grafico, no entanto ao fazer o fit parece ter tendencia
116 abline(lm(nossoDiabetes$glyhb ~ nossoDiabetes$waist)) ##crescente
117 plot(nossoDiabetes$glyhb ~ nossoDiabetes$hip, type = "p") ##o mesmo do anterior
118 abline(lm(nossoDiabetes$glyhb ~ nossoDiabetes$hip))
119 plot(nossoDiabetes$glyhb ~ nossoDiabetes$bp.1s, type = "p")##parece afetar com
120 ##tendencia crescente
121 abline(lm(nossoDiabetes$glyhb ~ nossoDiabetes$bp.1s)) ##confirma
122 plot(nossoDiabetes$glyhb ~ nossoDiabetes$bp.1d, type = "p")##nao parece afetar
123 abline(lm(nossoDiabetes$glyhb ~ nossoDiabetes$bp.1d)) ##tendencia constante
124
125 par(mfrow=c(2,2))
126 plot(nossoDiabetes$glyhb ~ nossoDiabetes$chol, type = "p")
127 plot(nossoDiabetes$glyhb ~ nossoDiabetes$stab.glu, type = "p")
128 plot(nossoDiabetes$glyhb ~ nossoDiabetes$hdl, type = "p")
129 plot(nossoDiabetes$glyhb ~ nossoDiabetes$ratio, type = "p")
130 plot(nossoDiabetes$glyhb ~ nossoDiabetes$age, type = "p")
131 plot(nossoDiabetes$glyhb ~ nossoDiabetes$height, type = "p")

```

```

132 plot(nossoDiabetes$glyhb ~ nossoDiabetes$weight, type = "p")
133 plot(nossoDiabetes$glyhb ~ nossoDiabetes$time.ppn, type = "p")
134 plot(nossoDiabetes$glyhb ~ nossoDiabetes$waist, type = "p")
135 plot(nossoDiabetes$glyhb ~ nossoDiabetes$hip, type = "p")
136 plot(nossoDiabetes$glyhb ~ nossoDiabetes$bp.ls, type = "p")
137 plot(nossoDiabetes$glyhb ~ nossoDiabetes$bp.1d, type = "p")

```

Regsubsets

```

1 par(mfrow=c(1,1))
2 selecao = regsubsets(diabetesB~.-glyhb,data=nossoDiabetes,nvmax=16)
3 summary(selecao)
4 plot(selecao,scale="adjr2")
5
6
7 selecao = regsubsets(diabetesB~.-glyhb,data=nossoDiabetes,nvmax=16, method = "forward")
8 summary(selecao)
9 plot(selecao,scale="adjr2")
10
11 selecao = regsubsets(diabetesB~.-glyhb,data=nossoDiabetes,nvmax=16, method = "backward")
12 summary(selecao)
13 plot(selecao,scale="adjr2")
14
15 selecao = regsubsets(diabetesB~.-glyhb,data=nossoDiabetes,nvmax=16, method = "seqrep")
16 summary(selecao)
17 plot(selecao,scale="adjr2")

```

Análise à normalidade

```

1 #-----#
2 # Teste à normalidade /
3 #-----#
4
5
6 ### ANLISE COLESTROL ##
7 colestrol <- diabetes$chol
8 qqnorm(colestrol) ##Parece normal ao contrario do ze
9 qqline(colestrol, col = 2)
10 hist(colestrol)
11 shapiro.test(colestrol)
12 valoresNormal <- rnorm(400,mean=mean(colestrol, na.rm = TRUE), sd=sd(colestrol, na.rm = TRUE))
13 hist(valoresNormal) ##PARECE NORMAL
14 fivenum(colestrol)
15 #Min: 78, Max:443, Media 179, 25%->78 75%->230
16
17
18
19 ### ANLISE STAB-GLU ##
20 stab <- diabetes$stab.glu
21 hist(stab) #
22 #FIVENUM()
23 fivenum(stab)
24 #Min: 48, Max:385, Media 89, 25%->81 75%->106
25 valoresNormal <- rnorm(400,mean=mean(stab, na.rm = TRUE), sd=sd(stab, na.rm = TRUE))
26 hist(valoresNormal) ##PARECE +- NORMAL
27
28 ##ANALISE RACIO ##
29 racio <- diabetes$ratio
30 hist(racio)
31 summary(racio)

```

```

32 valoresNormal <- rnorm(400,mean=mean(racio, na.rm = TRUE), sd=sd(racio, na.rm = TRUE))
33 hist(valoresNormal)
34 qqnorm(racio)
35 qqline(racio, col = 2) ##Difícil de analisar
36
37
38
39 ### ANLISE HDL ##
40
41 hdl <- diabetes$hdl
42 #Histograma
43 hist(hdl)
44 valoresNormal <- rnorm(400,mean=mean(hdl, na.rm = TRUE), sd=sd(hdl, na.rm = TRUE))
45 hist(valoresNormal) ##PARECE +- NORMAL
46 fivenum(hdl)
47 #Min: 12, Max:120, Media 46, 25%>38 75%>59
48
49
50 ##ANALISE AGE
51 age <- diabetes$age
52 summary(age)
53 hist(age)
54 qqnorm(age)
55 qqline(age, col="2")
56 valoresNormal <- rnorm(400,mean=mean(age, na.rm = TRUE), sd=sd(age, na.rm = TRUE)) ## Difícil de analisar
57 hist(valoresNormal) ##PARECE NORMAL
58
59
60 # Age
61
62 par(mfrow=c(2,1))
63 hist(nossoDiabetes$age) ## Aproxima-se da distribuição normal, contudo do lado esquerdo apresenta
64 ## um crescimento bastante mais acentuado que do lado direito.
65 mean(nossoDiabetes$age,na.rm=TRUE)
66 sd(nossoDiabetes$age,na.rm=TRUE)
67 valoresNormal <- rnorm(400,mean=mean(nossoDiabetes$age, na.rm = TRUE), sd=sd(nossoDiabetes$age, na.rm = TRUE
  ))
68 hist(valoresNormal) ## Observando os dois diagramas podemos afirmar que são parecidos. é possível
69 ## notar valores ligeiramente mais elevados à esquerda da média no de cima
70 summary(nossoDiabetes$age)
71 shapiro.test(valoresNormal) ## O diagrama inferior tem um pvalor alto pelo que não se rejeita
72 shapiro.test(nossoDiabetes$age) ## O diagrama superior tem um pvalor muito reduzido pelo que se
73 ## rejeita a hipótese nula (H0: dados normalmente distribuídos)
74 par(mfrow=c(1,1))
75 qqnorm(nossoDiabetes$age)
76 qqline(nossoDiabetes$age, col = 2) ## Do gráfico resultante podemos ver que a variável "Age"
77 ## não parece ser normalmente distribuída. Apesar de entre os
78 ## 35 e os 45 anos mostrar bastante convergência, fora deste
79 ## intervalo é os valores divergem bastante
80
81 # Height
82 par(mfrow=c(2,1))
83 hist(nossoDiabetes$height) ## Verifica-se algumas semelhanças à distribuição normal (maior
84 ## probabilidade em torno do centro), contudo é possível verificar
85 ## que na zona central a frequência mostra-se relativamente constante
86 ## ao longo de um largo intervalo.
87 mean(nossoDiabetes$height,na.rm=TRUE)
88 sd(nossoDiabetes$height,na.rm=TRUE)
89 valoresNormal <- rnorm(400,mean=mean(nossoDiabetes$height, na.rm = TRUE), sd=sd(nossoDiabetes$height, na.rm

```

```

    = TRUE))
90 hist(valoresNormal) ## Observando os dois diagramas é possível verificar algumas diferenças com
91 ## as freqüências mais elevadas a mostrarem-se em torno das 67 polegadas no
92 ## diagrama inferior ao passo que no diagrama superior as freqüências mais
93 ## elevadas se encontravam em torno das 63 polegadas
94 summary(nossoDiabetes$height)
95 shapiro.test(valoresNormal) ## O diagrama inferior tem um pvalor alto (30%) pelo que não se rejeita
96 shapiro.test(nossoDiabetes$height) ## O diagrama superior tem um pvalor muito reduzido (0.2%) pelo
97 ## que se rejeita a hipótese nula.
98 par(mfrow=c(1,1))
99 qqnorm(nossoDiabetes$height)
100 qqline(nossoDiabetes$height, col = 2) ## Do gráfico resultante podemos ver que a variável "Height"
101 ## não parece ser normalmente distribuída. Para alturas
102 ## inferiores a 62 e superiores a 72 polegadas verifica-se
103 ## uma tendência a divergir dos valores teóricos.
104
105
106 # Weight
107 set.seed(1)
108 par(mfrow=c(2,1))
109 hist(nossoDiabetes$weight) ## Olhando para o histograma é possível retirar bastantes parecências a
110 ## uma distribuição normal. Do lado esquerda dos 170 pounds verifica-se
111 ## uma subida mais constante da frequência relativamente ao lado direito
112 ## (os dois lados diferem ligeiramente)
113 mean(nossoDiabetes$weight, na.rm=TRUE)
114 sd(nossoDiabetes$weight, na.rm=TRUE)
115 valoresNormal <- rnorm(400, mean=mean(nossoDiabetes$weight, na.rm = TRUE), sd=sd(nossoDiabetes$weight, na.rm
    = TRUE))
116 hist(valoresNormal) ## Observando os dois diagramas é possível verificar bastantes semelhanças
117 ## entre eles nomeadamente no valor onde a frequência é mais elevada (170 pounds)
118 ## e também a nível da distribuição à esquerda e à direita deste valor.
119 summary(nossoDiabetes$weight)
120 shapiro.test(valoresNormal) ## O diagrama inferior tem um pvalor alto (80%) pelo que não se rejeita
121 shapiro.test(nossoDiabetes$weight) ## O diagrama superior tem um pvalor muito reduzido (<0.01%) pelo
122 ## que se rejeita a hipótese nula.
123 par(mfrow=c(1,1))
124 qqnorm(nossoDiabetes$weight)
125 qqline(nossoDiabetes$weight, col = 2) ## Do gráfico resultante podemos ver que a variável "Weight"
126 ## não parece ser normalmente distribuída. Para pesos
127 ## inferiores a 130 pounds e superiores a 220 pounds
128 ## verifica-se uma tendência a divergir superiormente dos
129 ## valores teóricos.
130
131
132 par(mfrow=c(2,1))
133 hist(nossoDiabetes$glyhb) ##tem algumas parecencias com distribuição normal
134 ## no entanto tem varios valores para o lado de lá (apesar de serem muito poucos)
135 ## o melhor será verificar com uma linha da distribuição normal
136 valoresNormal <- rnorm(400, mean=mean(nossoDiabetes$glyhb, na.rm = TRUE), sd=sd(nossoDiabetes$glyhb, na.rm =
    TRUE))
137 hist(valoresNormal)
138 summary(nossoDiabetes$glyhb)
139 shapiro.test(valoresNormal) ##este tem um pvalor alto e n se rejeita
140 shapiro.test(nossoDiabetes$glyhb) ##este teste tem um pvalor baixo pelo que se
141 ##rejeita a hipotese nula (de q os dados sao normalmente distribuidos)
142 par(mfrow=c(1,1))
143 qqnorm(valoresNormal)
144 qqline(valoresNormal, col = 2)
145 qqnorm(nossoDiabetes$glyhb)

```

```

146 qqline(nossoDiabetes$glyhb, col = 2) ##do grafico do teste da normal vemos que
147 ##a variavel se afasta muito e não parece normalmente distribuida
148
149
150 par(mfrow=c(2,1))
151 hist(nossoDiabetes$time.ppn) ##não se aproxima da distribuição normal
152 valoresNormal <- rnorm(400,mean=mean(nossoDiabetes$time.ppn, na.rm = TRUE), sd=sd(nossoDiabetes$time.ppn, na
    .rm = TRUE))
153 hist(valoresNormal)
154 summary(nossoDiabetes$time.ppn)
155 shapiro.test(valoresNormal) ##este tem um pvalor alto e n se rejeita
156 shapiro.test(nossoDiabetes$time.ppn) ##este teste tem um pvalor baixo pelo que se
157 ##rejeita a hipotese nula (de q os dados sao normalmente distribuidos)
158 par(mfrow=c(1,1))
159 qqnorm(nossoDiabetes$time.ppn)
160 qqline(nossoDiabetes$time.ppn, col = 2) ##do grafico do teste da normal vemos que
161 ##a variavel se afasta muito e não parece normalmente distribuida
162
163
164 par(mfrow=c(2,1))
165 hist(nossoDiabetes$waist) ##aproxima-se algo da distribuição normal
166 valoresNormal <- rnorm(400,mean=mean(nossoDiabetes$waist, na.rm = TRUE), sd=sd(nossoDiabetes$waist, na.rm =
    TRUE))
167 hist(valoresNormal) ##os histogramas são parecidos
168 summary(nossoDiabetes$waist)
169 shapiro.test(valoresNormal) ##este tem um pvalor alto e n se rejeita
170 shapiro.test(nossoDiabetes$waist) ##este teste tem um pvalor baixo pelo que se
171 ##rejeita a hipotese nula (de q os dados sao normalmente distribuidos)
172 par(mfrow=c(1,1))
173 qqnorm(valoresNormal)
174 qqline(valoresNormal, col = 2)
175 qqnorm(nossoDiabetes$waist)
176 qqline(nossoDiabetes$waist, col = 2) ##do grafico do teste da normal vemos que
177 ##a variavel nao se afasta muito e parece normalmente distribuida (no meio
178 ## até se comporta bem, mas nos extremos começa a afastar um pouco embora
179 ## aparenta afastar-se menosdo que o seguinte, nomeadamente no extremo superior)
180
181
182 par(mfrow=c(2,1))
183 hist(nossoDiabetes$hip) ##aproxma-se ligeiramente da distribuição normal
184 mean(nossoDiabetes$hip,na.rm=TRUE)
185 sd(nossoDiabetes$hip,na.rm=TRUE)
186 valoresNormal <- rnorm(400,mean=mean(nossoDiabetes$hip, na.rm = TRUE), sd=sd(nossoDiabetes$hip, na.rm = TRUE
    ))
187 hist(valoresNormal) ##comparando os 2 histogramas vemos que é parecido
188 summary(nossoDiabetes$hip)
189 shapiro.test(valoresNormal) ##este tem um pvalor alto e n se rejeita
190 shapiro.test(nossoDiabetes$hip) ##este teste tem um pvalor baixo pelo que se
191 ##rejeita a hipotese nula (de q os dados sao normalmente distribuidos)
192 par(mfrow=c(1,1))
193 qqnorm(nossoDiabetes$hip)
194 qqline(nossoDiabetes$hip, col = 2) ##do grafico do teste da normal vemos que
195 ##a variavel se afasta muito e não parece normalmente distribuida (no meio
196 ## até se comporta bem, mas nos extremos começa a afastar muito, embora
197 ## aparenta ser menos do queo seguinte)
198
199
200 par(mfrow=c(2,1))
201 hist(nossoDiabetes$bp.1s) ##tem algumas parecencas, mas tem o mm problema da

```

```

202 ##primeira, embora com menos amplitude para lá (diferença do maior valor)
203 valoresNormal <- rnorm(400,mean=mean(nossoDiabetes$bp.1s, na.rm = TRUE), sd=sd(nossoDiabetes$bp.1s, na.rm =
    TRUE))
204 hist(valoresNormal) ##não são parecidos os histogramas, as freqs sao diferentes
205 summary(nossoDiabetes$bp.1s)
206 shapiro.test(valoresNormal) ##este tem um pvalor alto e n se rejeita
207 shapiro.test(nossoDiabetes$bp.1s) ##este teste tem um pvalor baixo pelo que se
208 ##rejeita a hipotese nula (de q os dados sao normalmente distribuidos)
209 par(mfrow=c(1,1))
210 qqnorm(nossoDiabetes$bp.1s)
211 qqline(nossoDiabetes$bp.1s, col = 2) ##do grafico do teste da normal vemos que
212 ##a variavel se afasta muito e não parece normalmente distribuida (no meio
213 ## até se comporta bem, mas nos extremos começa a afastar muito)
214
215
216 par(mfrow=c(2,1))
217 hist(nossoDiabetes$bp.1d) ##parece normalmente distribuído
218 valoresNormal <- rnorm(400,mean=mean(nossoDiabetes$bp.1d, na.rm = TRUE), sd=sd(nossoDiabetes$bp.1d, na.rm =
    TRUE))
219 hist(valoresNormal)
220 summary(nossoDiabetes$bp.1d)
221 shapiro.test(valoresNormal) ##este tem um pvalor alto e n se rejeita
222 shapiro.test(nossoDiabetes$bp.1d) ##este teste tem um pvalor baixo pelo que se
223 ##rejeita a hipotese nula (de q os dados sao normalmente distribuidos)
224 ##no entanto este p-valor é bastante superior a todos os outros sendo de 0.01
225 par(mfrow=c(1,1))
226 qqnorm(nossoDiabetes$bp.1d)
227 qqline(nossoDiabetes$bp.1d, col = 2) ##do grafico do teste da normal vemos que
228 ##a variavel se afasta um pouco, mas não muito pelo que não conseguimos
229 ## concluir nada em concreto (aparenta ser normal)

```

Seleção do Modelo

Regressão Linear

```

1
2 ## NOSSA ANALISE ##
3
4 ## Teste com 1 var do regsubsets ## stab.glu
5 treino = dplyr::select(nossoDiabetes, glyhb,stab.glu)
6 treino = treino[complete.cases(treino),]
7
8 lm.fit.Stab.glu = lm(glyhb~stab.glu, data = treino)
9 par(mfrow=c(2,2))
10 plot(lm.fit.Stab.glu)
11 ## Explicar o plot aqui
12 # Olhando para o grafico Residuals vs Fited podemos verificar que os resíduos estão
13 # espalhados em volta de uma linha horizontal, o que indica que em princípio não
14 # existem relações não lineares
15 # Olhando para o grafico Normal Q-Q podemos verificar que os resíduos parecem
16 # seguir uma distribuição normal
17
18 lm.fit1 <- lm.fit.Stab.glu
19 summary(lm.fit1) # Adjusted R-squared: 0.5602
20
21 ## Verificar validação cruzada
22 train.control <- trainControl(method = "cv", number = 10)
23
24 set.seed(1)

```



```

84  ## O MAE apresenta-se mais baixo no model1.
85  ## O RMSE apresenta-se mais baixo no model1.
86  ## O Adjusted R-squared apresenta-se mais elevado no model1.
87  ## Como as diferenças são consideráveis, o melhor modelo é então o model1:
88  ## glyhb~stab.glu + age
89
90
91  #-----
92
93
94  # 3 variáveis -> stab.glu + age + time.ppn
95  treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,age,time.ppn)
96  treino = treino[complete.cases(treino),]
97
98  lm.fit.Time.ppn = lm(glyhb~time.ppn, data = treino)
99  par(mfrow=c(2,2))
100 plot(lm.fit.Time.ppn)
101 ## Explicar o plot aqui
102 # Olhando para o grafico Residuals vs Fited podemos verificar que os resíduos
103 # estão uniformemente espalhados em volta de uma linha reta, o que indica
104 # que em princípio não existem relações não lineares.
105 # Olhando para o grafico Normal Q-Q podemos verificar que os resíduos não parecem
106 # seguir uma distribuição normal
107 summary(lm.fit.Time.ppn) # Adjusted R-squared: -0.001485
108 # Sozinha parece não adicionar precisão ao modelo
109
110
111 lm.fitAnt <- lm(glyhb~stab.glu + age, data = treino)
112 summary(lm.fitAnt) # Adjusted R-squared: 0.5739
113
114 lm.fit1 <- lm(glyhb~stab.glu + age + time.ppn, data = treino)
115 summary(lm.fit1) # Adjusted R-squared: 0.5786 -> Melhorou muito pouco
116
117 # F-test parcial:
118 anova(lm.fitAnt, lm.fit1)
119 # Rejeitamos a hipótese nula de que o modelo lm.fit2 não é significativamente melhor
120 # que o modelo lm.fit1
121
122 # Não podemos assumir que lm.fit1 é significativamente melhor que lm.fitAnt devido a
123 # questões de overfitting.
124 # É necessário verificar por validação cruzada!
125
126
127 ## Verificar validação cruzada
128 train.control <- trainControl(method = "cv", number = 10)
129
130 ## Melhor modelo anterior:
131 set.seed(1)
132 modelAnt <- train(glyhb~stab.glu + age, data = treino, method = "lm", trControl = train.control)
133 print(modelAnt)
134 # Adjusted R-squared: 0.615430
135 1-(1-mean(modelAnt$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)
136
137 # Modelos a testar:
138 set.seed(1)
139 model1 <- train(glyhb~stab.glu + age + time.ppn, data = treino, method = "lm", trControl = train.control)
140 print(model1)
141 # Adjusted R-squared: 0.615881 -> Melhorou muito pouco. Não vale a pena aumentar
142 # a complexidade

```

```

143 1-(1-mean(model1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)
144
145 # Melhor modelo:
146 models <- resamples(list("modelAnt"=modelAnt,
147                           "model1"=model1))
148 summary(models)
149 ## O MAE apresenta-se mais baixo no modelAnt.
150 ## O RMSE apresenta-se mais baixo no model1 mas a diferença é muito reduzida.
151 ## O Adjusted R-squared apresenta-se mais elevado no model1 mas melhorou muito pouco.
152 ## O melhor modelo é então o model5 -> glyhb~stab.glu + age
153
154
155
156 #-----
157
158 # 3 variáveis -> stab.glu + age + hip
159 treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,age,hip)
160 treino = treino[complete.cases(treino),]
161
162 lm.fit.hip = lm(glyhb~hip, data = treino)
163 par(mfrow=c(2,2))
164 plot(lm.fit.hip)
165 ## Explicar o plot aqui
166 # Olhando para o grafico Residuals vs Fited podemos verificar que os resíduos
167 # estão uniformemente espalhados em volta de uma linha reta, o que indica
168 # que em princípio não existem relações não lineares.
169 # Olhando para o grafico Normal Q-Q podemos verificar que os resíduos não parecem
170 # seguir uma distribuição normal mas também não se
171 summary(lm.fit.hip) # Adjusted R-squared: 0.01746
172 # Sozinha parece não adicionar precisão ao modelo
173
174
175 lm.fitAnt <- lm(glyhb~stab.glu + age, data = treino)
176 summary(lm.fitAnt) # Adjusted R-squared: 0.5739
177
178 lm.fit1 <- lm(glyhb~stab.glu + age + hip, data = treino)
179 summary(lm.fit1) # Adjusted R-squared: 0.5748 -> Melhorou muito pouco
180
181 # F-test parcial:
182 anova(lm.fitAnt, lm.fit1)
183 # Aceitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
184 # que o modelo lm.fitAnt
185
186 ## O melhor modelo é então o modelAnt -> glyhb~stab.glu + age
187
188
189 #-----
190
191 # 3 Var -> stab.glu + age + bp.1s
192 treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,age,bp.1s)
193 treino = treino[complete.cases(treino),]
194
195 lm.fit.bp.1s = lm(glyhb~bp.1s, data = treino)
196 par(mfrow=c(2,2))
197 plot(lm.fit.bp.1s)
198 ## Explicar o plot aqui
199 # Olhando para o grafico Residuals vs Fited podemos verificar que os resíduos
200 # estão uniformemente espalhados em volta de uma linha reta, o que indica
201 # que em princípio não existem relações não lineares.

```

```

202 # Olhando para o grafico Normal Q-Q podemos verificar que os resíduos não parecem
203 # seguir uma distribuição normal.
204 summary(lm.fit.bp.1s) # Adjusted R-squared: 0.03667
205 # Sozinha parece não adicionar precisão ao modelo
206
207
208
209
210 lm.fitAnt <- lm(glyhb~stab.glu + age, data = treino)
211 summary(lm.fitAnt) # Adjusted R-squared: 0.5739
212
213 lm.fit1 <- lm(glyhb~stab.glu + age + bp.1s, data = treino)
214 summary(lm.fit1) # Adjusted R-squared: 0.5742 -> Melhorou muito pouco
215
216 # F-test parcial:
217 anova(lm.fitAnt, lm.fit1)
218 # Aceitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
219 # que o modelo lm.fitAnt
220
221 ## O melhor modelo é então o modelAnt -> glyhb~stab.glu + age
222
223
224 #-----
225
226 #####
227 # Melhor modelo!!! #
228 #####
229 # 2 variáveis -> glyhb~stab.glu + age
230 treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,age,time.ppn,hip,bp.1s)
231 treino = treino[complete.cases(treino),]
232
233 lm.fitAnt <- lm(glyhb~stab.glu + age, data = treino)
234 summary(lm.fitAnt) # Adjusted R-squared: 0.5735
235
236 set.seed(1)
237 train.control <- trainControl(method = "cv", number = 10)
238 melhorModelo <- train(glyhb~stab.glu + age, data = treino, method = "lm", trControl = train.control)
239 print(melhorModelo)
240 ## Para este dataset (treino) -> Adjusted R-squared: 0.579255
241 1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)
242
243
244 #-----
245
246
247 #####
248 # Modelos com combinações #
249 #####
250
251 ### REGSUBSETS
252 # 2 Var -> stab.glu + chol
253 # 3 Var -> stab.glu + ratio + age
254 # 4 Var -> stab.glu + ratio + age + time.ppn
255 # 5 Var -> stab.glu + ratio + age + time.ppn + chol
256
257 treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,age,time.ppn,hip,bp.1s)
258 treino = treino[complete.cases(treino),]
259
260 # 2 Var -> stab.glu + age

```

```

261 lm.fitAnt <- lm(glyhb~stab.glu + age, data = treino)
262 summary(lm.fitAnt) # Adjusted R-squared: 0.5735
263 lm.fit1 <- lm(glyhb~stab.glu + age + stab.glu:age, data = treino)
264 summary(lm.fit1) # Adjusted R-squared: 0.5792 -> melhorou muito pouco
265
266 ## Verificar validação cruzada
267 train.control <- trainControl(method = "cv", number = 10)
268
269 set.seed(1)
270 modeloAnt <- train(glyhb~stab.glu + age, data = treino, method = "lm", trControl = train.control)
271 print(modeloAnt)
272 # Adjusted R-squared: 0.579255
273 1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)
274
275 set.seed(1)
276 modelo1 <- train(glyhb~stab.glu + age + stab.glu:age, data = treino, method = "lm", trControl = train.
      control)
277 print(modelo1)
278 # Adjusted R-squared: 0.571200 -> Pior
279 1-(1-mean(modelo1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)
280
281 ## Melhor modelo até agora: glyhb~stab.glu + age
282
283 #-----
284
285 # 3 Var -> stab.glu + age + time.ppn
286 lm.fitAnt <- lm(glyhb~stab.glu + age, data = treino)
287 summary(lm.fitAnt) # Adjusted R-squared: 0.5735
288
289 lm.fit1 <- lm(glyhb~stab.glu + age + time.ppn + stab.glu:time.ppn, data = treino)
290 summary(lm.fit1) # Adjusted R-squared: 0.5935 -> melhorou bastante
291 lm.fit2 <- lm(glyhb~stab.glu + age + time.ppn + age:time.ppn, data = treino)
292 summary(lm.fit2) # Adjusted R-squared: 0.5834 -> melhorou um pouco
293
294 # F-test parcial:
295 anova(lm.fitAnt, lm.fit1)
296 # Rejeitamos a hipótese nula de que o modelo lm.fit1 não é significativamente melhor
297 # que o modelo lm.fitAnt
298 anova(lm.fitAnt, lm.fit2)
299 # Rejeitamos a hipótese nula de que o modelo lm.fit2 não é significativamente melhor
300 # que o modelo lm.fitAnt
301
302
303 ## Verificar validação cruzada
304 train.control <- trainControl(method = "cv", number = 10)
305
306 set.seed(1)
307 modeloAnt <- train(glyhb~stab.glu + age, data = treino, method = "lm", trControl = train.control)
308 print(modeloAnt)
309 # Adjusted R-squared: 0.579255
310 1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)
311
312 set.seed(1)
313 modelo1 <- train(glyhb~stab.glu + age + time.ppn + stab.glu:time.ppn, data = treino, method = "lm",
      trControl = train.control)
314 print(modelo1)
315 # Adjusted R-squared: 0.590397 -> Melhorou um pouco
316 1-(1-mean(modelo1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-4-1)
317

```

```

318 set.seed(1)
319 modelo2 <- train(glyhb~stab.glu + age + time.ppn + age:time.ppn, data = treino, method = "lm", trControl =
    train.control)
320 print(modelo2)
321 # Adjusted R-squared: 0.5789102 -> Piorou
322 1-(1-mean(modelo2$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-4-1)
323
324 ## Não compensa adicionar 2 preditores para melhorar a previsão em apenas 1%
325 ## O melhor modelo continua a ser o modeloAnt -> glyhb~stab.glu + age
326
327
328 #-----
329
330 # 4 Var -> stab.glu + age + time.ppn + hip
331 lm.fitAnt <- lm(glyhb~stab.glu + age, data = treino)
332 summary(lm.fitAnt) # Adjusted R-squared: 0.5735
333 lm.fit1 <- lm(glyhb~stab.glu + age + hip + stab.glu:hip, data = treino)
334 summary(lm.fit1) # Adjusted R-squared: 0.5736 -> igual
335 lm.fit2 <- lm(glyhb~stab.glu + age + hip + age:hip, data = treino)
336 summary(lm.fit2) # Adjusted R-squared: 0.5741 -> melhorou muito pouco
337 lm.fit3 <- lm(glyhb~stab.glu + age + time.ppn + hip + time.ppn:hip, data = treino)
338 summary(lm.fit3) # Adjusted R-squared: 0.5882 -> melhorou um pouco
339
340 # F-test parcial:
341 anova(lm.fitAnt, lm.fit1)
342 # Aceitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
343 # que o modelo lm.fitAnt
344 anova(lm.fitAnt, lm.fit2)
345 # Aceitamos a hipotese nula de que o modelo lm.fit2 não é significativamente melhor
346 # que o modelo lm.fitAnt
347 anova(lm.fitAnt, lm.fit3)
348 # Rejeitamos a hipotese nula de que o modelo lm.fit3 não é significativamente melhor
349 # que o modelo lm.fitAnt
350
351 # Não podemos assumir que lm.fit3 é significativamente
352 # melhor que lm.fitAnt devido a questões de overfitting.
353 # É necessário verificar por validação cruzada!
354
355
356 ## Verificar validação cruzada
357 train.control <- trainControl(method = "cv", number = 10)
358
359 set.seed(1)
360 modeloAnt <- train(glyhb~stab.glu + age, data = treino, method = "lm", trControl = train.control)
361 print(modeloAnt)
362 # Adjusted R-squared: 0.579255
363 1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)
364
365 set.seed(1)
366 modelo1 <- train(glyhb~stab.glu + age + time.ppn + hip + time.ppn:hip, data = treino, method = "lm",
    trControl = train.control)
367 print(modelo1)
368 # Adjusted R-squared: 0.5910418 -> melhorou um pouco
369 1-(1-mean(modelo1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-5-1)
370
371
372 ## Não compensa adicionar 2 preditores para melhorar a previsão em apenas 1%
373 ## O melhor modelo continua a ser o modeloAnt -> glyhb~stab.glu + age
374

```

```

375 #-----
376
377 # 5 Var -> stab.glu + age + time.ppn + hip + bp.1s
378 lm.fitAnt <- lm(glyhb~stab.glu + age, data = treino)
379 summary(lm.fitAnt) # Adjusted R-squared: 0.5735
380 lm.fit1 <- lm(glyhb~stab.glu + age + bp.1s + stab.glu:bp.1s, data = treino)
381 summary(lm.fit1) # Adjusted R-squared: 0.5719 -> Piorou
382 lm.fit2 <- lm(glyhb~stab.glu + age + bp.1s + age:bp.1s, data = treino)
383 summary(lm.fit2) # Adjusted R-squared: 0.5719 -> Piorou
384 lm.fit3 <- lm(glyhb~stab.glu + age + bp.1s + time.ppn + time.ppn:bp.1s, data = treino)
385 summary(lm.fit3) # Adjusted R-squared: 0.5789 -> Melhorou muito pouco
386 lm.fit4 <- lm(glyhb~stab.glu + age + bp.1s + hip + hip:bp.1s, data = treino)
387 summary(lm.fit4) # Adjusted R-squared: 0.5728 -> Piorou
388
389 # F-test parcial:
390 anova(lm.fitAnt, lm.fit1)
391 # Aceitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
392 # que o modelo lm.fitAnt
393 anova(lm.fitAnt, lm.fit2)
394 # Aceitamos a hipotese nula de que o modelo lm.fit2 não é significativamente melhor
395 # que o modelo lm.fitAnt
396 anova(lm.fitAnt, lm.fit3)
397 # Aceitamos a hipotese nula de que o modelo lm.fit3 não é significativamente melhor
398 # que o modelo lm.fitAnt
399 anova(lm.fitAnt, lm.fit4)
400 # Aceitamos a hipotese nula de que o modelo lm.fit4 não é significativamente melhor
401 # que o modelo lm.fitAnt
402
403 # Não podemos assumir que lm.fit1 e lm.fit2 são significativamente
404 # melhores que lm.fitAnt devido a questões de overfitting.
405 # É necessário verificar por validação cruzada!
406
407
408 ## O melhor modelo continua a ser o modeloAnt -> glyhb~stab.glu + age
409
410 #-----
411
412 #####
413 # Melhor modelo com combinações!!! #
414 #####
415
416 # Combinações não parecem favorecer o modelo. Deste modo o melhor modelo é:
417 lm.fitAnt <- lm(glyhb~stab.glu + age, data = treino)
418 summary(lm.fitAnt) # Adjusted R-squared: 0.5735
419
420 train.control <- trainControl(method = "cv", number = 10)
421
422 set.seed(1)
423 modeloAnt <- train(glyhb~stab.glu + age, data = treino, method = "lm", trControl = train.control)
424 print(modeloAnt)
425 # Adjusted R-squared: 0.579255
426 1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)
427
428 #-----
429
430
431
432 ## REGSUBSETS ##
433

```

```

434 lm.fit <- lm(glyhb ~ chol + stab.glu + hdl + ratio + age + gender + height + weight + frame + waist + hip +
    time.ppn, data = nossoDiabetes, x = TRUE, y = TRUE)
435 summary(lm.fit)
436
437
438 reg = regsubsets(glyhb~.-diabetesB, data=nossoDiabetes, nvmax=10)
439 summary(reg)
440 # 1 Var -> stab.glu
441 # 2 Var -> stab.glu + chol
442 # 3 Var -> stab.glu + ratio + age
443 # 4 Var -> stab.glu + ratio + age + time.ppn
444 # 5 Var -> stab.glu + ratio + age + time.ppn + chol
445
446
447 ## Teste com 1 var do regsubsets ## stab.glu
448 set.seed(1)
449 treino = dplyr::select(nossoDiabetes, glyhb,stab.glu)
450 treino = treino[complete.cases(treino),]
451
452 lm.fit.Stab.glu = lm(glyhb~stab.glu, data = treino)
453 par(mfrow=c(2,2))
454 plot(lm.fit.Stab.glu)
455 ## Explicar o plot aqui
456 # Olhando para o grafico Residuals vs Fited podemos verificar que os resíduos estão
457 # espalhados em volta de uma linha horizontal, o que indica que em princípio não
458 # existem relações não lineares
459 # Olhando para o grafico Normal Q-Q podemos verificar que os resíduos parecem s
460 # seguir uma distribuição normal
461
462 lm.fit1 <- lm.fit.Stab.glu
463 summary(lm.fit1) # Adjusted R-squared: 0.5602
464
465 ## Elevar o grau de stab.glu (grau 2)
466 lm.fit2 <- lm(glyhb~poly(stab.glu, 2, raw=TRUE), data = treino)
467 summary(lm.fit2) # Adjusted R-squared: 0.5678 -> Melhorou um pouco
468
469 # F-test parcial
470 anova(lm.fit1, lm.fit2)
471 # Rejeitamos a hipotese nula de que o modelo lm.fit2 não é significativamente melhor
472 # que o modelo lm.fit1.
473
474 ## Elevar o grau de stab.glu (grau 3)
475 lm.fit3 <- lm(glyhb~poly(stab.glu, 3, raw=TRUE), data = treino)
476 summary(lm.fit3) # Adjusted R-squared: 0.5877 -> Melhorou bastante
477
478 # F-test parcial
479 anova(lm.fit1, lm.fit3)
480 # Rejeitamos também a hipotese nula de que o modelo lm.fit3 não é significativamente
481 # melhor que o modelo lm.fit1.
482
483 # Não podemos assumir que estes modelos são significativamente melhores que lm.fit1
484 # devido a questões de overfitting. É necessário verificar por validação cruzada!
485
486 ## Verificar validação cruzada
487 train.control <- trainControl(method = "cv", number = 10)
488
489 set.seed(1)
490 model1 <- train(glyhb~stab.glu, data = treino, method = "lm", trControl = train.control)
491 print(model1)

```



```

492 # Adjusted R-squared: 0.567815
493 1-(1-mean(model1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-1-1)
494
495 set.seed(1)
496 model2 <- train(glyhb~poly(stab.glu, 2, raw=TRUE), data = treino, method = "lm", trControl = train.control)
497 print(model2)
498 # Adjusted R-squared: 0.562181
499 1-(1-mean(model2$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)
500
501 set.seed(1)
502 model3 <- train(glyhb~poly(stab.glu, 3, raw=TRUE), data = treino, method = "lm", trControl = train.control)
503 print(model3)
504 # Adjusted R-squared: 0.558148
505 1-(1-mean(model3$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)
506
507 models <- resamples(list("stab.glu(grau1)"=model1,
508                          "stab.glu(grau2)"=model2,
509                          "stab.glu(grau3)"=model3))
510 summary(models)
511 ## O MAE apresenta-se mais baixo no model3.
512 ## O RMSE apresenta-se mais baixo no model1.
513 ## O Adjusted R-squared apresenta-se mais elevado no model1.
514 ## O melhor modelo é então o model1 -> glyhb~stab.glu
515
516 #-----
517
518 # 2 variáveis -> stab.glu + chol
519 set.seed(1)
520 treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,chol)
521 treino = treino[complete.cases(treino),]
522
523 lm.fit.Chol = lm(glyhb~chol, data = treino)
524 par(mfrow=c(2,2))
525 plot(lm.fit.Chol)
526 ## Explicar o plot aqui
527 # Olhando para o grafico Residuals vs Fited podemos verificar que os resíduos
528 # estão uniformemente espalhados em volta de uma linha "reta", o que indica
529 # que em princípio não existem relações não lineares.
530 # Olhando para o grafico Normal Q-Q podemos verificar que os resíduos não parecem
531 # seguir uma distribuição normal
532 summary(lm.fit.Chol) # Adjusted R-squared: 0.05863
533 # Parece adicionar precisão ao modelo
534
535 lm.fitAnt <- lm(glyhb~stab.glu, data = treino)
536 summary(lm.fitAnt) # Adjusted R-squared: 0.5602
537
538 lm.fit1 <- lm(glyhb~stab.glu + chol, data = treino)
539 summary(lm.fit1) # Adjusted R-squared: 0.5759 -> Melhorou bastante
540
541 # F-test parcial
542 anova(lm.fitAnt, lm.fit1)
543 # Rejeitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
544 # que o modelo lm.fitAnt.
545
546 lm.fit2 <- lm(glyhb~stab.glu + poly(chol, 2, raw=TRUE), data = treino)
547 summary(lm.fit2) # Adjusted R-squared: 0.5759 -> Não melhorou
548
549 # F-test parcial
550 anova(lm.fitAnt, lm.fit2)

```

```

551 # Aceitamos a hipotese nula de que o modelo lm.fit2 não é significativamente melhor
552 # que o modelo lm.fit1.
553
554 lm.fit3 <- lm(glyhb~stab.glu + poly(chol, 3, raw=TRUE), data = treino)
555 summary(lm.fit3) # Adjusted R-squared: 0.5826 -> Melhorou muito pouco
556
557 # F-test parcial
558 anova(lm.fit1, lm.fit3)
559 # Rejeitamos a hipotese nula de que o modelo lm.fit3 não é significativamente melhor
560 # que o modelo lm.fit1.
561
562 # Não podemos assumir que lm.fit1 e lm.fit3 são significativamente melhores que
563 # lm.fitAnt devido a questões de overfitting. É necessário verificar por validação cruzada!
564
565
566 ## Verificar validação cruzada
567 train.control <- trainControl(method = "cv", number = 10)
568
569 ## Melhor modelo anterior:
570 set.seed(1)
571 modelAnt <- train(glyhb~stab.glu, data = treino, method = "lm", trControl = train.control)
572 print(modelAnt)
573 # Adjusted R-squared: 0.567614
574 1-(1-mean(modelAnt$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-1-1)
575
576 # Modelos a testar:
577 set.seed(1)
578 model1 <- train(glyhb~stab.glu + chol, data = treino, method = "lm", trControl = train.control)
579 print(model1)
580 # Adjusted R-squared: 0.581092 -> Melhor
581 1-(1-mean(model1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)
582
583 set.seed(1)
584 model2 <- train(glyhb~stab.glu + poly(chol, 3, raw=TRUE), data = treino, method = "lm", trControl = train.
    control)
585 print(model2)
586 # Adjusted R-squared: 0.577321 -> Piorou (overfitting)
587 1-(1-mean(model2$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-4-1)
588
589 models <- resamples(list("modelAnt"=modelAnt,
590                          "model1"=model1,
591                          "model2"=model2))
592 summary(models)
593 ## O MAE apresenta-se mais baixo no model1.
594 ## O RMSE apresenta-se mais baixo no model2 mas o model1 fica muito próximo.
595 ## O Adjusted R-squared apresenta-se mais elevado no model1.
596 ## O melhor modelo é então o model1.
597
598
599 #-----
600
601
602 # 3 variáveis -> stab.glu + ratio + age
603 treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,chol,ratio,age)
604 treino = treino[complete.cases(treino),]
605
606 lm.fit.Ratio = lm(glyhb~ratio, data = treino)
607 par(mfrow=c(2,2))
608 plot(lm.fit.Ratio)

```

```

609 ## Explicar o plot aqui
610 # Olhando para o grafico Residuals vs Fited podemos verificar que os resíduos
611 # estão uniformemente espalhados em volta de uma linha um pouco curvada, o que
612 # indica que poderão existir relações não lineares.
613 # Olhando para o grafico Normal Q-Q podemos verificar que os resíduos não parecem
614 # seguir uma distribuição normal
615 summary(lm.fit.Ratio) # Adjusted R-squared: 0.1057
616 # Parece adicionar precisão ao modelo
617
618 lm.fit.Age = lm(glyhb~age, data = treino)
619 par(mfrow=c(2,2))
620 plot(lm.fit.Age)
621 ## Explicar o plot aqui
622 # Olhando para o grafico Residuals vs Fited podemos verificar que os resíduos
623 # estão uniformemente espalhados em volta de uma linha reta, o que indica
624 # que em princípio não existem relações não lineares.
625 # Olhando para o grafico Normal Q-Q podemos verificar que os resíduos não parecem
626 # seguir uma distribuição normal
627 summary(lm.fit.Age) # Adjusted R-squared: 0.1128
628 # Parece adicionar precisão ao modelo
629
630 lm.fitAnt <- lm(glyhb~stab.glu + chol, data = treino)
631 summary(lm.fitAnt) # Adjusted R-squared: 0.5759
632
633 lm.fit1 <- lm(glyhb~stab.glu + ratio, data = treino)
634 summary(lm.fit1) # Adjusted R-squared: 0.5738 -> Pior
635
636 lm.fit2 <- lm(glyhb~stab.glu + poly(ratio, 2, raw=TRUE), data = treino)
637 summary(lm.fit2) # Adjusted R-squared: 0.5810 -> Melhorou muito pouco
638
639 # F-test parcial:
640 anova(lm.fit1, lm.fit2)
641 # Rejeitamos a hipótese nula de que o modelo lm.fit2 não é significativamente melhor
642 # que o modelo lm.fit1
643
644 lm.fit3 <- lm(glyhb~stab.glu + poly(ratio, 3, raw=TRUE), data = treino)
645 summary(lm.fit3) # Adjusted R-squared: 0.5823 -> Melhorou muito pouco
646
647 # F-test parcial:
648 anova(lm.fit2, lm.fit3)
649 # Aceitamos a hipótese nula de que o modelo lm.fit3 não é significativamente melhor
650 # que o modelo lm.fit2.
651
652 lm.fit4 <- lm(glyhb~stab.glu + age, data = treino)
653 summary(lm.fit4) # Adjusted R-squared: 0.5742 -> Pior
654
655 lm.fit5 <- lm(glyhb~stab.glu + poly(age, 2, raw=TRUE), data = treino)
656 summary(lm.fit5) # Adjusted R-squared: 0.5754 -> Pior
657
658 # F-test parcial:
659 anova(lm.fit4, lm.fit5)
660 # Aceitamos a hipótese nula de que o modelo lm.fit5 não é significativamente melhor
661 # que o modelo lm.fit4.
662
663 lm.fit6 <- lm(glyhb~stab.glu + poly(age, 3, raw=TRUE), data = treino)
664 summary(lm.fit6) # Adjusted R-squared: 0.5798 -> Melhorou muito pouco
665
666 # F-test parcial:
667 anova(lm.fit4, lm.fit6)

```

```

668 # Rejeitamos a hipotese nula de que o modelo lm.fit6 não é significativamente melhor
669 # que o modelo lm.fit4
670
671 # Não podemos assumir que lm.fit2 é significativamente melhor que lm.fit1 e que
672 # lm.fit6 é significativamente melhor que lm.fit4 e lm.fit6 devido a questões de
673 # overfitting. Também não podemos comparar os modelos lm.fit1 a lm.fit6 com o modelo
674 # lm.fitAnt pois ao passar de duas variáveis para três variáveis foi removida uma
675 # variável (chol) e adicionadas duas novas (ratio e age).
676 # É necessário verificar por validação cruzada!
677
678
679 ## Verificar validação cruzada
680 train.control <- trainControl(method = "cv", number = 10)
681
682 ## Melhor modelo anterior:
683 set.seed(1)
684 modelAnt <- train(glyhb~stab.glu + chol, data = treino, method = "lm", trControl = train.control)
685 print(modelAnt)
686 # Adjusted R-squared: 0.581092
687 1-(1-mean(modelAnt$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)
688
689 # Modelos a testar:
690 set.seed(1)
691 model1 <- train(glyhb~stab.glu + ratio, data = treino, method = "lm", trControl = train.control)
692 print(model1)
693 # Adjusted R-squared: 0.578752 -> Pior
694 1-(1-mean(model1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)
695
696 set.seed(1)
697 model2 <- train(glyhb~stab.glu + poly(ratio, 2, raw=TRUE), data = treino, method = "lm", trControl = train.
        control)
698 print(model2)
699 # Adjusted R-squared: 0.575364 -> Pior (overfitting)
700 1-(1-mean(model2$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)
701
702 set.seed(1)
703 model3 <- train(glyhb~stab.glu + age, data = treino, method = "lm", trControl = train.control)
704 print(model3)
705 # Adjusted R-squared: 0.584609 -> Ligeiramente melhor
706 1-(1-mean(model3$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-2-1)
707
708 set.seed(1)
709 model4 <- train(glyhb~stab.glu + poly(age, 2, raw=TRUE), data = treino, method = "lm", trControl = train.
        control)
710 print(model4)
711 # Adjusted R-squared: 0.586733 -> Melhorou muito pouco (nao vale a pena)
712 1-(1-mean(model4$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)
713
714 ## É melhor usar ratio e age lineares.
715 set.seed(1)
716 model5 <- train(glyhb~stab.glu + ratio + age, data = treino, method = "lm", trControl = train.control)
717 print(model5)
718 # Adjusted R-squared: 0.594697 -> Melhorou ligeiramente
719 1-(1-mean(model5$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)
720
721 # Melhores modelos:
722 models <- resamples(list("modelAnt"=modelAnt,
723                          "model1"=model1,
724                          "model3"=model3,

```

```

725         "model5"=model5))
726 summary(models)
727 ## O MAE apresenta-se mais baixo no model5.
728 ## O RMSE apresenta-se mais baixo no model5.
729 ## O Adjusted R-squared apresenta-se mais elevado no model5.
730 ## O melhor modelo é então o model5 -> glyhb~stab.glu + ratio + age
731
732
733
734 #-----
735
736 # 4 variáveis -> stab.glu + ratio + age + time.ppn
737 treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,ratio,age,time.ppn)
738 treino = treino[complete.cases(treino),]
739
740 lm.fit.Time.ppn = lm(glyhb~time.ppn, data = treino)
741 par(mfrow=c(2,2))
742 plot(lm.fit.Time.ppn)
743 ## Explicar o plot aqui
744 # Olhando para o grafico Residuals vs Fited podemos verificar que os resíduos
745 # estão uniformemente espalhados em volta de uma linha reta, o que indica
746 # que em princípio não existem relações não lineares.
747 # Olhando para o grafico Normal Q-Q podemos verificar que os resíduos não parecem
748 # seguir uma distribuição normal
749 summary(lm.fit.Time.ppn) # Adjusted R-squared: -0.001485
750 # Sozinha parece não adicionar precisão ao modelo
751
752
753 lm.fitAnt <- lm(glyhb~stab.glu + ratio + age, data = treino)
754 summary(lm.fitAnt) # Adjusted R-squared: 0.5855
755
756 lm.fit1 <- lm(glyhb~stab.glu + ratio + age + time.ppn, data = treino)
757 summary(lm.fit1) # Adjusted R-squared: 0.5907 -> Melhorou ligeiramente
758
759 # F-test parcial:
760 anova(lm.fitAnt, lm.fit1)
761 # Rejeitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
762 # que o modelo lm.fitAnt
763
764 lm.fit2 <- lm(glyhb~stab.glu + ratio + age + poly(time.ppn, 2, raw=TRUE), data = treino)
765 summary(lm.fit2) # Adjusted R-squared: 0.5897 -> Piorou
766
767 # F-test parcial:
768 anova(lm.fit1, lm.fit2)
769 # Aceitamos a hipotese nula de que o modelo lm.fit2 não é significativamente melhor
770 # que o modelo lm.fit1.
771
772 lm.fit3 <- lm(glyhb~stab.glu + ratio + age + poly(time.ppn, 3, raw=TRUE), data = treino)
773 summary(lm.fit3) # Adjusted R-squared: 0.5915 -> Melhorou muito pouco
774
775 # F-test parcial:
776 anova(lm.fit1, lm.fit3)
777 # Aceitamos a hipotese nula de que o modelo lm.fit3 não é significativamente melhor
778 # que o modelo lm.fit1.
779
780 # Não podemos assumir que lm.fit1 é significativamente melhor que lm.fitAnt devido
781 # a questões de overfitting. É necessário verificar por validação cruzada!
782
783

```

```

784 ## Verificar validação cruzada
785 train.control <- trainControl(method = "cv", number = 10)
786
787 ## Melhor modelo anterior:
788 set.seed(1)
789 modelAnt <- train(glyhb~stab.glu + ratio + age, data = treino, method = "lm", trControl = train.control)
790 print(modelAnt)
791 # Adjusted R-squared: 0.624422 -> Ao remover os NAs do time.ppn melhorou o resultado
792 1-(1-mean(modelAnt$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)
793
794 # Modelos a testar:
795 set.seed(1)
796 model1 <- train(glyhb~stab.glu + ratio + age + time.ppn, data = treino, method = "lm", trControl = train.
      control)
797 print(model1)
798 # Adjusted R-squared: 0.626305 -> Melhorou muito pouco
799 1-(1-mean(model1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-4-1)
800
801
802 # Melhores modelos:
803 models <- resamples(list("modelAnt"=modelAnt,
804                          "model1"=model1))
805 summary(models)
806 ## O MAE apresenta-se ligeiramente mais baixo no model1.
807 ## O RMSE apresenta-se ligeiramente mais baixo no model1.
808 ## O Adjusted R-squared apresenta-se ligeiramente mais elevado no model1.
809 ## Não parece compensar aumentar a complexidade para ter resultados ligeiramente
810 ## melhores
811 ## O melhor modelo é então o modelAnt -> glyhb~stab.glu + ratio + age
812
813
814 #-----
815
816 # 5 variáveis -> stab.glu + ratio + age + time.ppn + chol
817 treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,ratio,age,time.ppn,chol)
818 treino = treino[complete.cases(treino),]
819
820 ## Análise do chol já foi feita em cima
821 summary(lm.fit.Chol) # Adjusted R-squared: 0.05863
822 # Parece adicionar precisão ao modelo
823
824
825 lm.fitAnt <- lm(glyhb~stab.glu + ratio + age, data = treino)
826 summary(lm.fitAnt) # Adjusted R-squared: 0.5855
827
828 lm.fit1 <- lm(glyhb~stab.glu + ratio + age + time.ppn + chol, data = treino)
829 summary(lm.fit1) # Adjusted R-squared: 0.5934 -> Melhorou ligeiramente
830
831 # F-test parcial:
832 anova(lm.fitAnt, lm.fit1)
833 # Rejeitamos a hipótese nula de que o modelo lm.fit1 não é significativamente melhor
834 # que o modelo lm.fitAnt
835
836 lm.fit2 <- lm(glyhb~stab.glu + ratio + age + time.ppn + poly(chol, 2, raw=TRUE), data = treino)
837 summary(lm.fit2) # Adjusted R-squared: 0.5934 -> Melhorou ligeiramente
838
839 # F-test parcial:
840 anova(lm.fit1, lm.fit2)
841 # Aceitamos a hipótese nula de que o modelo lm.fit2 não é significativamente melhor

```

```

842 # que o modelo lm.fit1
843
844 lm.fit3 <- lm(glyhb~stab.glu + ratio + age + time.ppn + poly(chol, 3, raw=TRUE), data = treino)
845 summary(lm.fit3) # Adjusted R-squared: 0.5984 -> Melhorou ligeiramente
846
847 # F-test parcial:
848 anova(lm.fit1, lm.fit3)
849 # Rejeitamos a hipotese nula de que o modelo lm.fit3 não é significativamente melhor
850 # que o modelo lm.fit1
851
852 # Não podemos assumir que lm.fit1 e lm.fit3 são significativamente melhores que
853 # lm.fitAnt devido a questões de overfitting.
854 # É necessário verificar por validação cruzada!
855
856
857 ## Verificar validação cruzada
858 train.control <- trainControl(method = "cv", number = 10)
859
860 ## Melhor modelo anterior:
861 set.seed(1)
862 modelAnt <- train(glyhb~stab.glu + ratio + age, data = treino, method = "lm", trControl = train.control)
863 print(modelAnt)
864 # Adjusted R-squared: 0.624422 -> Ao remover os NAs do time.ppn melhorou o resultado
865 1-(1-mean(modelAnt$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)
866
867 # Modelos a testar:
868 set.seed(1)
869 model1 <- train(glyhb~stab.glu + ratio + age + time.ppn + chol, data = treino, method = "lm", trControl =
      train.control)
870 print(model1)
871 # Adjusted R-squared: 0.626159 -> Melhorou muito pouco
872 1-(1-mean(model1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-5-1)
873
874 set.seed(1)
875 model2 <- train(glyhb~stab.glu + ratio + age + time.ppn + poly(chol, 3, raw=TRUE), data = treino, method = "
      lm", trControl = train.control)
876 print(model2)
877 # Adjusted R-squared: 0.620654 -> Piorou
878 1-(1-mean(model2$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-6-1)
879
880 # Melhores modelos:
881 models <- resamples(list("modelAnt"=modelAnt,
882                          "model1"=model1,
883                          "model2"=model2))
884 summary(models)
885 ## O MAE apresenta-se ligeiramente mais baixo no model1.
886 ## O RMSE apresenta-se ligeiramente mais baixo no modelAnt.
887 ## O Adjusted R-squared apresenta-se ligeiramente mais elevado no model1.
888 ## Não parece compensar aumentar a complexidade para ter resultados ligeiramente
889 ## melhores
890 ## O melhor modelo é então o modelAnt -> glyhb~stab.glu + ratio + age
891
892
893 #-----
894
895 #####
896 # Melhor modelo!!! #
897 #####
898 # 3 variáveis -> glyhb~stab.glu + ratio + age

```

```

899 treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,ratio,age,time.ppn,chol)
900 treino = treino[complete.cases(treino),]
901
902 lm.fitAnt <- lm(glyhb~stab.glu + ratio + age, data = treino)
903 summary(lm.fitAnt) # Adjusted R-squared: 0.5858
904
905 set.seed(1)
906 train.control <- trainControl(method = "cv", number = 10)
907 melhorModelo <- train(glyhb~stab.glu + ratio + age, data = treino, method = "lm", trControl = train.control)
908 print(melhorModelo)
909 ## Para este dataset (treino) -> Adjusted R-squared: 0.594697
910 # Adjusted R-squared: 0.624422 -> Ao remover os NAs do time.ppn melhorou o resultado
911 1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)
912
913
914 #-----
915
916
917 #####
918 # Modelos com combinações #
919 #####
920
921 ### REGSUBSETS
922 # 2 Var -> stab.glu + chol
923 # 3 Var -> stab.glu + ratio + age
924 # 4 Var -> stab.glu + ratio + age + time.ppn
925 # 5 Var -> stab.glu + ratio + age + time.ppn + chol
926
927 treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,ratio,age,time.ppn,chol)
928
929 treino = treino[complete.cases(treino),]
930
931 # 2 Var -> stab.glu + chol
932 lm.fitAnt <- lm(glyhb~stab.glu + ratio + age, data = treino)
933 summary(lm.fitAnt) # Adjusted R-squared: 0.5858
934 lm.fit1 <- lm(glyhb~stab.glu + chol + stab.glu:chol, data = treino)
935 summary(lm.fit1) # Adjusted R-squared: 0.5811 -> baixou
936
937 ## Verificar validação cruzada
938 train.control <- trainControl(method = "cv", number = 10)
939
940 set.seed(1)
941 modeloAnt <- train(glyhb~stab.glu + ratio + age, data = treino, method = "lm", trControl = train.control)
942 print(modeloAnt)
943 # Adjusted R-squared: 0.624422
944 1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)
945
946 set.seed(1)
947 modelo1 <- train(glyhb~stab.glu + chol + stab.glu:chol, data = treino, method = "lm", trControl = train.
          control)
948 print(modelo1)
949 # Adjusted R-squared: 0.612082 -> Pior
950 1-(1-mean(modelo1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)
951
952 #-----
953
954 # 3 Var -> stab.glu + ratio + age
955 lm.fitAnt <- lm(glyhb~stab.glu + ratio + age, data = treino)
956 summary(lm.fitAnt) # Adjusted R-squared: 0.5858

```



```

957
958 lm.fit1 <- lm(glyhb~stab.glu + ratio + age + stab.glu:ratio, data = treino)
959 summary(lm.fit1) # Adjusted R-squared: 0.5910 -> melhorou um pouco
960
961 # F-test parcial:
962 anova(lm.fitAnt, lm.fit1)
963 # Rejeitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
964 # que o modelo lm.fitAnt
965
966 lm.fit2 <- lm(glyhb~stab.glu + ratio + age + stab.glu:age, data = treino)
967 summary(lm.fit2) # Adjusted R-squared: 0.5896 -> melhorou muito pouco
968
969 # F-test parcial:
970 anova(lm.fitAnt, lm.fit2)
971 # Rejeitamos a hipotese nula de que o modelo lm.fit2 não é significativamente melhor
972 # que o modelo lm.fitAnt
973
974 lm.fit3 <- lm(glyhb~stab.glu + ratio + age + stab.glu:ratio + stab.glu:age, data = treino)
975 summary(lm.fit3) # Adjusted R-squared: 0.5959 -> melhorou um pouco
976
977 # F-test parcial:
978 anova(lm.fitAnt, lm.fit3)
979 # Rejeitamos a hipotese nula de que o modelo lm.fit3 não é significativamente melhor
980 # que o modelo lm.fitAnt
981
982 lm.fit4 <- lm(glyhb~stab.glu + ratio + age + ratio:age, data = treino)
983 summary(lm.fit4) # Adjusted R-squared: 0.5849 -> Pior
984
985 # F-test parcial:
986 anova(lm.fitAnt, lm.fit4)
987 # Aceitamos a hipotese nula de que o modelo lm.fit4 não é significativamente melhor
988 # que o modelo lm.fitAnt
989
990 ## Verificar validação cruzada
991 train.control <- trainControl(method = "cv", number = 10)
992
993 set.seed(1)
994 modeloAnt <- train(glyhb~stab.glu + ratio + age, data = treino, method = "lm", trControl = train.control)
995 print(modeloAnt)
996 # Adjusted R-squared: 0.624422
997 1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)
998
999 set.seed(1)
1000 modelo1 <- train(glyhb~stab.glu + ratio + age + stab.glu:ratio, data = treino, method = "lm", trControl =
      train.control)
1001 print(modelo1)
1002 # Adjusted R-squared: 0.613438 -> Pior
1003 1-(1-mean(modelo1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-4-1)
1004
1005 set.seed(1)
1006 modelo2 <- train(glyhb~stab.glu + ratio + age + stab.glu:age, data = treino, method = "lm", trControl =
      train.control)
1007 print(modelo2)
1008 # Adjusted R-squared: 0.614949 -> Pior
1009 1-(1-mean(modelo2$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-4-1)
1010
1011 set.seed(1)
1012 modelo3 <- train(glyhb~stab.glu + ratio + age + ratio:age, data = treino, method = "lm", trControl = train.
      control)

```

```

1013 print(modelo3)
1014 # Adjusted R-squared: 0.621185 -> Pior
1015 1-(1-mean(modelo3$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-4-1)
1016
1017 ## O melhor modelo continua a ser o modeloAnt -> glyhb~stab.glu + ratio + age
1018
1019 #-----
1020
1021 # 4 Var -> stab.glu + ratio + age + time.ppn
1022 lm.fitAnt <- lm(glyhb~stab.glu + ratio + age, data = treino)
1023 summary(lm.fitAnt) # Adjusted R-squared: 0.5858
1024 lm.fit1 <- lm(glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn, data = treino)
1025 summary(lm.fit1) # Adjusted R-squared: 0.6036 -> melhorou um pouco
1026 lm.fit2 <- lm(glyhb~stab.glu + ratio + age + time.ppn + ratio:time.ppn, data = treino)
1027 summary(lm.fit2) # Adjusted R-squared: 0.5917 -> melhorou muito pouco
1028 lm.fit3 <- lm(glyhb~stab.glu + ratio + age + time.ppn + age:time.ppn, data = treino)
1029 summary(lm.fit3) # Adjusted R-squared: 0.5973 -> melhorou um pouco
1030
1031 # F-test parcial:
1032 anova(lm.fitAnt, lm.fit1)
1033 # Rejeitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
1034 # que o modelo lm.fitAnt
1035 anova(lm.fitAnt, lm.fit2)
1036 # Rejeitamos a hipotese nula de que o modelo lm.fit2 não é significativamente melhor
1037 # que o modelo lm.fitAnt
1038 anova(lm.fitAnt, lm.fit3)
1039 # Rejeitamos a hipotese nula de que o modelo lm.fit3 não é significativamente melhor
1040 # que o modelo lm.fitAnt
1041
1042 # Não podemos assumir que lm.fit1, lm.fit2, lm.fit3 e lm.fit4 são significativamente
1043 # melhores que lm.fitAnt devido a questões de overfitting.
1044 # É necessário verificar por validação cruzada!
1045
1046
1047 ## Verificar validação cruzada
1048 train.control <- trainControl(method = "cv", number = 10)
1049
1050 set.seed(1)
1051 modeloAnt <- train(glyhb~stab.glu + ratio + age, data = treino, method = "lm", trControl = train.control)
1052 print(modeloAnt)
1053 # Adjusted R-squared: 0.624422
1054 1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)
1055
1056 set.seed(1)
1057 modelo1 <- train(glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn, data = treino, method = "lm",
1058                 trControl = train.control)
1059 print(modelo1)
1060 # Adjusted R-squared: 0.637304 -> melhorou um pouco
1061 1-(1-mean(modelo1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-5-1)
1062
1063 set.seed(1)
1064 modelo2 <- train(glyhb~stab.glu + ratio + age + time.ppn + ratio:time.ppn, data = treino, method = "lm",
1065                 trControl = train.control)
1066 print(modelo2)
1067 # Adjusted R-squared: 0.623103 -> Pior
1068 1-(1-mean(modelo2$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-5-1)
1069
1070 set.seed(1)
1071 modelo3 <- train(glyhb~stab.glu + ratio + age + time.ppn + age:time.ppn, data = treino, method = "lm",

```

```

    trControl = train.control)
1070 print(modelo3)
1071 # Adjusted R-squared: 0.625895 -> Melhorou muito pouco
1072 1-(1-mean(modelo3$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-5-1)
1073
1074 ## O melhor modelo passa a ser o modelo1 -> glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn
1075
1076 #-----
1077
1078 # 5 Var -> stab.glu + ratio + age + time.ppn + chol
1079 lm.fitAnt <- lm(glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn, data = treino)
1080 summary(lm.fitAnt) # Adjusted R-squared: 0.6036
1081 lm.fit1 <- lm(glyhb~stab.glu + ratio + age + time.ppn + chol + stab.glu:time.ppn + stab.glu:chol, data =
    treino)
1082 summary(lm.fit1) # Adjusted R-squared: 0.6083 -> Melhorou muito pouco
1083 lm.fit2 <- lm(glyhb~stab.glu + ratio + age + time.ppn + chol + stab.glu:time.ppn + ratio:chol, data = treino
    )
1084 summary(lm.fit2) # Adjusted R-squared: 0.6142 -> Melhorou um pouco
1085 lm.fit3 <- lm(glyhb~stab.glu + ratio + age + time.ppn + chol + stab.glu:time.ppn + age:chol, data = treino)
1086 summary(lm.fit3) # Adjusted R-squared: 0.6044 -> Melhorou muito pouco
1087 lm.fit4 <- lm(glyhb~stab.glu + ratio + age + time.ppn + chol + stab.glu:time.ppn + time.ppn:chol, data =
    treino)
1088 summary(lm.fit4) # Adjusted R-squared: 0.6042 -> Melhorou muito pouco
1089
1090 # F-test parcial:
1091 anova(lm.fitAnt, lm.fit1)
1092 # Rejeitamos a hipotese nula de que o modelo lm.fit1 não é significativamente melhor
1093 # que o modelo lm.fitAnt
1094 anova(lm.fitAnt, lm.fit2)
1095 # Rejeitamos a hipotese nula de que o modelo lm.fit2 não é significativamente melhor
1096 # que o modelo lm.fitAnt
1097 anova(lm.fitAnt, lm.fit3)
1098 # Aceitamos a hipotese nula de que o modelo lm.fit3 não é significativamente melhor
1099 # que o modelo lm.fitAnt
1100 anova(lm.fitAnt, lm.fit4)
1101 # Aceitamos a hipotese nula de que o modelo lm.fit4 não é significativamente melhor
1102 # que o modelo lm.fitAnt
1103
1104 # Não podemos assumir que lm.fit1 e lm.fit2 são significativamente
1105 # melhores que lm.fitAnt devido a questões de overfitting.
1106 # É necessário verificar por validação cruzada!
1107
1108
1109 ## Verificar validação cruzada
1110 train.control <- trainControl(method = "cv", number = 10)
1111
1112 set.seed(1)
1113 modeloAnt <- train(glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn, data = treino, method = "lm"
    , trControl = train.control)
1114 print(modeloAnt)
1115 # Adjusted R-squared: 0.637304
1116 1-(1-mean(modeloAnt$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-5-1)
1117
1118 set.seed(1)
1119 modelo1 <- train(glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn, data = treino, method = "lm",
    trControl = train.control)
1120 print(modelo1)
1121 # Adjusted R-squared: 0.637304 -> melhorou um pouco
1122 1-(1-mean(modelo1$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-5-1)

```

```

1123
1124 set.seed(1)
1125 modelo2 <- train(glyhb~stab.glu + ratio + age + time.ppn + chol + stab.glu:time.ppn + stab.glu:chol, data =
      treino, method = "lm", trControl = train.control)
1126 print(modelo2)
1127 # Adjusted R-squared: 0.635052 -> Pior (overfitting)
1128 1-(1-mean(modelo2$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-7-1)
1129
1130 set.seed(1)
1131 modelo3 <- train(glyhb~stab.glu + ratio + age + time.ppn + chol + stab.glu:time.ppn + ratio:chol, method = "
      lm", trControl = train.control)
1132 print(modelo3)
1133 # Adjusted R-squared: 0.623915 -> Pior (overfitting)
1134 1-(1-mean(modelo3$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-7-1)
1135
1136 ## O melhor modelo continua a ser o modeloAnt -> glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn
1137
1138 #-----
1139
1140 #####
1141 # Melhor modelo com combinações!!! #
1142 #####
1143 # 5 variáveis -> glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn
1144 treino = dplyr::select(nossoDiabetes, glyhb,stab.glu,ratio,age,time.ppn,chol)
1145 treino = treino[complete.cases(treino),]
1146
1147 lm.fitAnt <- lm(glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn, data = treino)
1148 summary(lm.fitAnt) # Adjusted R-squared: 0.6036
1149
1150 set.seed(1)
1151 melhorModelo <- train(glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn, data = treino, method = "
      lm", trControl = train.control)
1152 print(melhorModelo)
1153 # Adjusted R-squared: 0.637304
1154 1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-5-1)
1155
1156 #-----
1157
1158 # Contudo é necessário adicionar duas novas variáveis (time.ppn e stab.glu:time.ppn)
1159 # para subir o Adjusted R-squared de 0.624422 para 0.637304
1160
1161 set.seed(1)
1162 modeloAnt <- train(glyhb~stab.glu + ratio + age, data = treino, method = "lm", trControl = train.control)
1163 print(modeloAnt)
1164 # Adjusted R-squared: 0.624422
1165 1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-3-1)
1166
1167 ## VS
1168
1169 set.seed(1)
1170 melhorModelo <- train(glyhb~stab.glu + ratio + age + time.ppn + stab.glu:time.ppn, data = treino, method = "
      lm", trControl = train.control)
1171 print(melhorModelo)
1172 # Adjusted R-squared: 0.637304
1173 1-(1-mean(melhorModelo$resample$Rsquared))*(nrow(treino)-1)/(nrow(treino)-5-1)
1174
1175 ## Nós achamos que não compensa aumentar essas duas variáveis para ter um ganho tão
1176 # pouco significativo pelo que determinados o melhor modelo como sendo:
1177 # glyhb ~ stab.glu + ratio + age

```

Regressão Logística

```
1
2
3
4 ### CROSS VALIDATION GLM
5
6 ## Retirar os nulls
7 dados = dplyr::select(diabetes, diabetesB,bp.1s,stab.glu,hdl,chol,age,ratio,stab.glu,weight,height,waist,hip
8   ,gender,location,frame)
9 dados2 = dados[complete.cases(dados),]
10 attach(dados2)
11
12 ## Utilizando a nossa analise de variaveis as mais importantes são: STAB.GLU age bp.1s ratio, waist
13 dados2 = nossoDiabetes
14 ## Melhor erro ponderaro: 3.46 , th:0.4
15 set.seed(1)
16 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu,dados2)
17 glm.fit=glm(diabetesB~stab.glu,data=dados,family=binomial)
18 plot(glm.fit)
19
20 melhor = melhorIndice(thr)
21
22 ## Melhor erro ponderaro: 3.43 , th:0.4
23 set.seed(1)
24 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+age,dados2)
25 melhor = melhorIndice(thr)
26
27 ## 1p - Melhor erro ponderaro: 3.54 , th:0.4
28 set.seed(1)
29 thr = graficoThreshold(0,1,0.01,glmCV,diabetesB~stab.glu+age+stab.glu:age,dados2)
30 melhor = melhorIndice(thr)
31 thr[40,]
32
33 ## Melhor erro ponderaro: 3.52 , th:0.2
34 set.seed(1)
35 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+age+bp.1s,dados2)
36 melhor = melhorIndice(thr)
37
38 ## Melhor erro ponderaro: 3.44 , th:0.4
39 set.seed(1)
40 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+age+bp.1s+ratio,dados2)
41 melhor = melhorIndice(thr)
42
43 ## Melhor erro ponderaro: 3.47 , th:0.4
44 set.seed(1)
45 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+age+bp.1s+ratio+waist,dados2)
46 melhor = melhorIndice(thr)
47
48 ## Melhor erro ponderaro: 3.47 , th:0.4
49 set.seed(1)
50 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+age+bp.1s+ratio+waist+weight,dados2)
51 melhor = melhorIndice(thr)
52
53 #####
54 #Utilizar o regsubsets #####
55 regfit.best=regsubsets(diabetesB~.,data=dados2,nvmax=19)
56 summary(regfit.best)
```

```

57
58
59
60 ## 1p - Melhor erro ponderaro: 3.46 , th:0.4
61 set.seed(1)
62 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~poly(stab.glu,3),dados2)
63 melhor = melhorIndice(thr)
64
65 ## 1p - Melhor erro ponderaro: 3.57 , th:0.4
66 set.seed(1)
67 thr = graficoThreshold(0,1,0.01,glmCV,diabetesB~poly(stab.glu,2),dados2)
68 melhor = melhorIndice(thr)
69 thr[40,]
70
71
72
73
74
75 ## 1p - Melhor erro ponderaro: 3.48 , th:0.4
76 set.seed(1)
77 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+ratio+stab.glu:ratio,dados2)
78 melhor = melhorIndice(thr)
79
80
81 ## 1p - Melhor erro ponderaro: 3.55 , th:0.4
82 set.seed(1)
83 thr = graficoThreshold(0,1,0.01,glmCV,diabetesB~stab.glu+ratio+stab.glu:ratio,dados2)
84 melhor = melhorIndice(thr)
85
86 ## 1p - Melhor erro ponderaro: 3.55 , th:0.4 anterior com 0.0.1 th
87 set.seed(1)
88 thr = graficoThreshold(0,1,0.01,glmCV,diabetesB~stab.glu+ratio+stab.glu:ratio,dados2)
89 melhor = melhorIndice(thr)
90
91 ## 1p - Melhor erro ponderaro: 3.57 , th:0.4 Stab.glu + ratio
92 set.seed(1)
93 thr = graficoThreshold(0,1,0.01,glmCV,diabetesB~poly(stab.glu,2)+ratio,dados2)
94 melhor = melhorIndice(thr)
95 thr[40,]
96
97 ## 1p - Melhor erro ponderaro: 3.54 , th:0.4 Stab.glu + ratio
98 set.seed(1)
99 thr = graficoThreshold(0,1,0.01,glmCV,diabetesB~poly(stab.glu,2)+ratio+age,nossoDiabetes)
100 melhor = melhorIndice(thr)
101 thr[40,]
102 thr[21,]
103
104
105 ## 2p -Melhor erro ponderaro: 3.43 , th:0.2
106 set.seed(1)
107 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+ age +stab.glu:age ,dados2)
108 melhor = melhorIndice(thr)
109
110
111 ## 2p -Melhor erro ponderaro: 3.48 , th:0.2
112 set.seed(1)
113 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+ratio,dados2)
114 melhor = melhorIndice(thr)
115

```

```

116 ## 2p -Melhor erro ponderaro: 3.55 , th:0.2
117 set.seed(1)
118 thr = graficoThreshold(0,1,0.01,glmCV,diabetesB~stab.glu+ratio,dados2)
119 melhor = melhorIndice(thr)
120
121 set.seed(1)
122 thr = graficoThreshold(0,1,0.01,glmCV,diabetesB~stab.glu+ratio,dados2)
123 melhor = melhorIndice(thr)
124
125 ## 3p - Melhor erro ponderaro: 3.46 , th:0.4
126 set.seed(1)
127 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+ratio+age,dados2)
128 melhor = melhorIndice(thr)
129
130 ## 4p - Melhor erro ponderaro: 3.46 , th:0.4
131 set.seed(1)
132 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+ratio+age+gender,dados2)
133 melhor = melhorIndice(thr)
134
135 ## 5p - Melhor erro ponderaro: 3.47 , th:0.4
136 set.seed(1)
137 thr = graficoThreshold(0,1,0.1,glmCV,diabetesB~stab.glu+ratio+age+gender+waist,dados2)
138 melhor = melhorIndice(thr)

```

KNN

```

1 ## NOSSA ANALISE ##
2 ## Teste com 1 var ##
3 treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu)
4 treino = treino[complete.cases(treino),]
5
6 resultadoKnnCV1 = geraKNNCV(treino,treino$diabetesB,1,20)
7
8 melhorK1 = melhorIndice(resultadoKnnCV1)
9
10 ## Teste com 2 vars ##
11 treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu, age)
12 treino = treino[complete.cases(treino),]
13
14 resultadoKnnCV2 = geraKNNCV(treino,treino$diabetesB,1,20)
15
16 melhorK2 = melhorIndice(resultadoKnnCV2)
17
18 ## Teste com 3 vars ##
19 treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu, age, bp.1s)
20 treino = treino[complete.cases(treino),]
21
22 resultadoKnnCV3 = geraKNNCV(treino,treino$diabetesB,1,20)
23
24 melhorK3 = melhorIndice(resultadoKnnCV3)
25
26 ## Teste com 4 vars ##
27 treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu, age, bp.1s, ratio)
28 treino = treino[complete.cases(treino),]
29
30 resultadoKnnCV4 = geraKNNCV(treino,treino$diabetesB,1,20)
31
32 melhorK4 = melhorIndice(resultadoKnnCV4)
33

```

```

34 ## Teste com 5 vars ##
35 treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu, age, bp.1s, ratio, waist)
36 treino = treino[complete.cases(treino),]
37
38 resultadoKnnCV5 = geraKNNCV(treino,treino$diabetesB,1,20)
39
40 melhorK5 = melhorIndice(resultadoKnnCV5)
41
42 ## o melhor é com 3 e k=10 ##
43
44 ## ordena todos para selecionar os 5 melhores ##
45 valores = rep(1,dim(resultadoKnnCV1)[1])
46 res1 = data.frame(resultadoKnnCV1,valores)
47 valores = rep(2,dim(resultadoKnnCV1)[1])
48 res2 = data.frame(resultadoKnnCV2,valores)
49 valores = rep(3,dim(resultadoKnnCV1)[1])
50 res3 = data.frame(resultadoKnnCV3,valores)
51 valores = rep(4,dim(resultadoKnnCV1)[1])
52 res4 = data.frame(resultadoKnnCV4,valores)
53 valores = rep(5,dim(resultadoKnnCV1)[1])
54 res5 = data.frame(resultadoKnnCV5,valores)
55
56 todos = rbind(res1,res2,res3,res4,res5)
57 todosC = consomeValores(todos)
58 todosOrd = todosC[order(-todosC$valor),]
59
60
61
62 ## REGSUBSETS ##
63
64 ## Teste com 1 var do regsubsets ##
65 treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu)
66 treino = treino[complete.cases(treino),]
67
68 resultadoKnnCV1 = geraKNNCV(treino,treino$diabetesB,1,20)
69
70 melhorK1 = melhorIndice(resultadoKnnCV1)
71
72 ## Teste com 2 vars do regsubsets ##
73 treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu, chol)
74 treino = treino[complete.cases(treino),]
75
76 resultadoKnnCV2 = geraKNNCV(treino,treino$diabetesB,1,20)
77
78 melhorK2 = melhorIndice(resultadoKnnCV2)
79
80 ## Teste com 3 vars do regsubsets ##
81 treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu, ratio, age)
82 treino = treino[complete.cases(treino),]
83
84 resultadoKnnCV3 = geraKNNCV(treino,treino$diabetesB,1,20)
85
86 melhorK3 = melhorIndice(resultadoKnnCV3)
87
88 ## Teste com 4 vars do regsubsets ##
89 treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu, ratio, age, time.ppn)
90 treino = treino[complete.cases(treino),]
91
92 resultadoKnnCV4 = geraKNNCV(treino,treino$diabetesB,1,20)

```



```

93
94 melhorK4 = melhorIndice(resultadoKnnCV4)
95
96 ## com o metodo forward a 3 e 4 dá diferente ##
97 ## com 3 ##
98 treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu, chol, age)
99 treino = treino[complete.cases(treino),]
100
101 resultadoKnnCV3F = geraKNNCV(treino,treino$diabetesB,1,20)
102
103 melhorK3F = melhorIndice(resultadoKnnCV3F)
104
105 ## com 4 ##
106 treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu, chol, age, time.ppn)
107 treino = treino[complete.cases(treino),]
108
109 resultadoKnnCV4F = geraKNNCV(treino,treino$diabetesB,1,20)
110
111 melhorK4F = melhorIndice(resultadoKnnCV4F)
112
113 ## com o metodo backward a segunda dá diferente ##
114 treino = dplyr::select(nossoDiabetes, diabetesB,stab.glu, ratio)
115 treino = treino[complete.cases(treino),]
116
117 resultadoKnnCV2B = geraKNNCV(treino,treino$diabetesB,1,20)
118
119 melhorK2B = melhorIndice(resultadoKnnCV2B)
120
121 ## ordena todos para selecionar os 5 melhores ##
122 valores = rep(1,dim(resultadoKnnCV1)[1])
123 res1 = data.frame(resultadoKnnCV1,valores)
124 valores = rep(2,dim(resultadoKnnCV1)[1])
125 res2 = data.frame(resultadoKnnCV2,valores)
126 valores = rep(3,dim(resultadoKnnCV1)[1])
127 res3 = data.frame(resultadoKnnCV3,valores)
128 valores = rep(4,dim(resultadoKnnCV1)[1])
129 res4 = data.frame(resultadoKnnCV4,valores)
130 valores = rep(5,dim(resultadoKnnCV1)[1])
131 res3F = data.frame(resultadoKnnCV3F,valores)
132 valores = rep(6,dim(resultadoKnnCV1)[1])
133 res4F = data.frame(resultadoKnnCV4F,valores)
134 valores = rep(7,dim(resultadoKnnCV1)[1])
135 res2B = data.frame(resultadoKnnCV2B,valores)
136
137 todos = rbind(res1,res2,res3,res4,res3F,res4F,res2B)
138 todosC = consomeValores(todos)
139 todosOrd = todosC[order(-todosC$valor),]
140 ## o melhor é com k=9 e 2 vars, no entanto não são melhores ##
141 ## que o nossa pelo que não será usado ##

```

QDA e LDA

```

1
2 ## QDA
3 treino = dplyr::select(nossoDiabetes, diabetesB, age)
4 treino = treino[complete.cases(treino),]
5
6 qda1 = graficoThreshold(0,1,0.02,qdaCV,diabetesB~age,treino)
7 melhorQ1 = melhorIndice(qda1)

```

```

8  ## acerto muito baixo em torno dos 71%
9
10 treino = dplyr::select(nossoDiabetes, diabetesB, age, bp.1s)
11 treino = treino[complete.cases(treino),]
12
13 qda2 = graficoThreshold(0,1,0.02,qdaCV,diabetesB~age+bp.1s,treino)
14 melhorQ2 = melhorIndice(qda2)
15 qda2[12,]
16 ## acerto muito baixo em torno dos 71%
17
18 treino = dplyr::select(nossoDiabetes, diabetesB, age, bp.1s, waist)
19 treino = treino[complete.cases(treino),]
20
21 qda3 = graficoThreshold(0,1,0.02,qdaCV,diabetesB~age+bp.1s+waist,treino)
22 melhorQ3 = melhorIndice(qda3)
23 ## acerto muito baixo em torno dos 72,25%
24
25
26 ##### LDA
27 treino = dplyr::select(nossoDiabetes, diabetesB, age)
28 treino = treino[complete.cases(treino),]
29
30 lda1 = graficoThreshold(0,1,0.02,ldaCV,diabetesB~age,treino)
31 melhorL1 = melhorIndice(lda1)
32 ## acerto na ordem dos 71,75%
33
34 treino = dplyr::select(nossoDiabetes, diabetesB, age, bp.1s)
35 treino = treino[complete.cases(treino),]
36
37 lda2 = graficoThreshold(0,1,0.02,ldaCV,diabetesB~age+bp.1s,treino)
38 melhorL2 = melhorIndice(lda2)
39 ## acerto muito baixo em torno dos 71,75%
40
41 treino = dplyr::select(nossoDiabetes, diabetesB, age, bp.1s, waist)
42 treino = treino[complete.cases(treino),]
43
44 lda3 = graficoThreshold(0,1,0.02,ldaCV,diabetesB~age+bp.1s+waist,treino)
45 melhorL3 = melhorIndice(lda3)
46 ## acerto muito baixo em torno dos 72,25%

```

Seleção do melhor modelo

```

1  ## Dividir em teste e treino ##
2
3  set.seed(1)
4
5  teste=sample(dim(nossoDiabetes)[1],dim(nossoDiabetes)[1]/4)
6  diabetesTeste = nossoDiabetes[teste,]
7  diabetesTreino = nossoDiabetes[-teste,]
8  summary(diabetesTeste)
9  summary(diabetesTreino)
10
11
12
13 ## Selecionar os melhores modelos ##
14 ## knn ##
15
16 treinoA = dplyr::select(diabetesTreino, diabetesB,stab.glu, age, bp.1s)
17 summary(treinoA)

```

```

18 treinoA = treinoA[complete.cases(treinoA),]
19 summary(treinoA)
20 testeA = dplyr::select(diabetesTeste, diabetesB, stab.glu, age, bp.1s)
21 summary(testeA)
22 testeA = testeA[complete.cases(testeA),]
23 summary(testeA)
24
25 set.seed(1)
26 knnK10 = knn(treinoA[,-1], testeA[,-1], treinoA$diabetesB, k=10)
27 table(knnK10, testeA$diabetesB)
28 ## 3.5991 ##
29 set.seed(1)
30 knnK12 = knn(treinoA[,-1], testeA[,-1], treinoA$diabetesB, k=12)
31 table(knnK12, testeA$diabetesB)
32 ## 3.4744 ##
33
34 ## glm ##
35 glm.fit1 = glm(diabetesB~poly(stab.glu,2)+ratio, data=diabetesTreino, family=binomial)
36 summary(glm.fit1)
37 plot(glm.fit1)
38
39 testeA = dplyr::select(diabetesTeste, diabetesB, stab.glu, ratio)
40 testeA = testeA[complete.cases(testeA),]
41 summary(testeA)
42
43 glm.pred1 = predict(glm.fit1, testeA, type="response")
44 glm.pred1 = ifelse(glm.pred1 > 0.39, 1, 0)
45
46 table(glm.pred1, testeA$diabetesB)
47 ## 3.5991 ##
48
49 ## Ratio n é importante, p-valor muito baixo ##
50 diabetesTreino[149,] ## é o 195
51 auxTreino = diabetesTreino#[-149,]
52 auxTreino[149,]
53 glm.fit1 = glm(diabetesB~poly(stab.glu,2), data=diabetesTreino, family=binomial)
54 summary(glm.fit1)
55 plot(glm.fit1)
56
57 testeA = dplyr::select(diabetesTeste, diabetesB, stab.glu)
58 testeA = testeA[complete.cases(testeA),]
59 summary(testeA)
60
61 glm.pred1 = predict(glm.fit1, testeA, type="response")
62 glm.pred1 = ifelse(glm.pred1 > 0.39, 1, 0)
63
64 table(glm.pred1, testeA$diabetesB)
65 ## 3.5991 ##
66
67 ## lm ##
68 diabetesTreino[149,] ## é o 195
69 auxTreino = diabetesTreino#[-149,]
70 auxTreino[149,]
71
72 lm.fitAnt <- lm(glyhb~stab.glu + age, data = auxTreino)
73 summary(lm.fitAnt)
74 plot(lm.fitAnt)
75
76

```

```

77 testeA = dplyr::select(diabetesTeste, diabetesB, stab.glu, age)
78 testeA = testeA[complete.cases(testeA),]
79 summary(testeA)
80
81 lm.pred1 = predict(lm.fitAnt, testeA, interval="prediction")
82
83 valoresLM1 = transformaBinaria(lm.pred1[,1])
84
85 table(valoresLM1, testeA$diabetesB)
86
87 ## 3.6318 ##
88
89 lm.fit2 <- lm(glyhb~stab.glu + ratio + age, data = diabetesTreino)
90 summary(lm.fit2)
91 plot(lm.fit2)
92
93 testeA = dplyr::select(diabetesTeste, diabetesB, stab.glu, ratio, age)
94 testeA = testeA[complete.cases(testeA),]
95 summary(testeA)
96
97 lm.pred2 = predict(lm.fit2, testeA, interval="prediction")
98
99 valoresLM2 = transformaBinaria(lm.pred2[,1])
100
101 table(valoresLM2, testeA$diabetesB)
102
103 ## 3.6318 ##

```

Resultados e respostas

```

1  lm.fitAnt <- lm(glyhb~stab.glu + age, data = auxTreino)
2  summary(lm.fitAnt)
3  plot(lm.fitAnt)
4
5
6  testeA = dplyr::select(diabetesTeste, diabetesB, stab.glu, age)
7  testeA = testeA[complete.cases(testeA),]
8  summary(testeA)
9
10 lm.pred1 = predict(lm.fitAnt, testeA, interval="prediction")
11
12 valoresLM1 = transformaBinaria(lm.pred1[,1])
13
14 table(valoresLM1, testeA$diabetesB)
15
16 ## Resposta ##
17 summary(nossoDiabetes$stab.glu)
18
19
20 ##meus dados o ano passado JOSE PEDRO
21 stab.glu = 91
22 age = 20
23 testeResposta = data.frame(stab.glu, age)
24 lm.predR = predict(lm.fitAnt, testeResposta, interval="prediction", level = .89999)
25 ## com ~90% de confiança posso afirmar que não tenho diabetes
26
27 ##
28 stab.glu = 91
29 age = 15

```

```

30 testeResposta = data.frame(stab.glu,age)
31 lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .911)
32 ## com 91.1% de confiança posso afirmar que não tenho diabetes
33
34 ##
35 stab.glu = 91
36 age = 50
37 testeResposta = data.frame(stab.glu,age)
38 lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .785)
39 ## com 50 anos a confiança desce para ~78.5% com a mesma stab.glu
40
41 ##
42 stab.glu = 91
43 age = 70
44 testeResposta = data.frame(stab.glu,age)
45 lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .666)
46 ## com 70 anos a confiança desce para ~66.6% com a mesma stab.glu
47
48 ##
49 stab.glu = 61
50 age = 20
51 testeResposta = data.frame(stab.glu,age)
52 lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .971)
53 ## com a mesma idade e menos stab.glu a confiança aumentou para 97.1
54
55 ##
56 stab.glu = 121
57 age = 20
58 testeResposta = data.frame(stab.glu,age)
59 lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .721)
60 ## com a mesma idade e mais stab.glu a confiança desceu para 72.1
61
62 ##
63 stab.glu = 151
64 age = 20
65 testeResposta = data.frame(stab.glu,age)
66 lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .40)
67 ## com a mesma idade e mais stab.glu a confiança desceu para 40
68
69 ## ter diabetes ##
70 ##
71 stab.glu = 191
72 age = 60
73 testeResposta = data.frame(stab.glu,age)
74 lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .547)
75 ## com estes valores temos 54.7% de confiança que tem diabetes
76
77 ##
78 stab.glu = 191
79 age = 40
80 testeResposta = data.frame(stab.glu,age)
81 lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .37)
82 ## com estes valores temos 37% de confiança que tem diabetes
83
84 ##
85 stab.glu = 191
86 age = 20
87 testeResposta = data.frame(stab.glu,age)
88 lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .167)

```

```

89  ## com 20 anos a confianca baixa para 16.7%
90
91  ##
92  stab.glu = 191
93  age = 75
94  testeResposta = data.frame(stab.glu,age)
95  lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .66)
96  ## aumentando a idade para 75 a confiança sobe para 66%
97
98
99  ##
100 stab.glu = 151
101 age = 60
102 testeResposta = data.frame(stab.glu,age)
103 lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .005)
104 ## diminuindo a stab.glu para 151 a confiança passa para 0.5%
105
106
107 ##
108 stab.glu = 171
109 age = 60
110 testeResposta = data.frame(stab.glu,age)
111 lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .299)
112 ## diminuindo a stab.glu para 171 a confiança passa para 29.9%
113
114 ##
115 stab.glu = 211
116 age = 60
117 testeResposta = data.frame(stab.glu,age)
118 lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .738)
119 ## aumentando a stab.glu para 211 a confiança passa para 73.8%
120
121 ##
122 stab.glu = 231
123 age = 60
124 testeResposta = data.frame(stab.glu,age)
125 lm.predR = predict(lm.fitAnt,testeResposta,interval="prediction",level = .863)
126 ## aumentando a stab.glu para 231 a confiança passa para 86.3%
127
128
129 ## Incidencia pessoas com mais e menos 50 anos ##
130 menos50 = nossoDiabetes[nossoDiabetes$age<50,]
131 mais50 = nossoDiabetes[nossoDiabetes$age>=50,]
132
133 quantosMenos0 = length(menos50$diabetesB[!is.na(menos50$diabetesB) & menos50$diabetesB==0])
134 quantosMenos1 = length(menos50$diabetesB[!is.na(menos50$diabetesB) & menos50$diabetesB==1])
135
136 quantosMenos0/(quantosMenos0+quantosMenos1) ##94,22% dos q tem menos de 50 n tem diabetes
137
138 quantosMais0 = length(menos50$diabetesB[!is.na(mais50$diabetesB) & mais50$diabetesB==0])
139 quantosMais1 = length(menos50$diabetesB[!is.na(mais50$diabetesB) & mais50$diabetesB==1])
140
141 quantosMais0/(quantosMais0+quantosMais1) ##70,40% dos q tem mais de 50 n tem diabetes
142
143 quantosMais0/(quantosMais0+quantosMenos0) ## 42.55% dos q n tem diabetes tem mais de 50 anos
144
145 quantosMais1/(quantosMais1+quantosMenos1) ## 83.54% dos q tem diabetes tem mais de 50 anos
146
147 ## Incidencia location ##

```

```

148 cidadeTeste = dplyr::select(nossoDiabetes,diabetesB,location)
149 cidadeTeste = cidadeTeste[complete.cases(cidadeTeste),]
150
151 barplot(table(cidadeTeste$diabetesB,cidadeTeste$location),col=c( "cornflowerblue" , "cornsilk3" ),beside=T,
      legend.text=c("Nao_ter","Ter"))
152 table(cidadeTeste$diabetesB,cidadeTeste$location)
153 31/190 ## em buckingham 16,32% tem diabetes
154 29/200 ## em louisa 14.5% tem diabetes
155
156 31/60 ##dos q tem diabetes 51.67% pertencem a buckingham
157 159/(159+171) ##dos q n tem diabetes 48.18%pertencem a buckingham
158 ## as incidencias sao bastante parecidas ##
159
160
161 ## Resultado melhor com lm ##

```

Definição de funções auxiliares

```

1 graficoThreshold <- function(inferior,superior,delta, funcao, formula, dados){
2   num = ((superior - inferior) / delta) + 2
3   aux = 0
4   auxT = 0
5   comeco = inferior
6   v = rep(0,num)
7   acerto = rep(0,num)
8   acertoP = rep(0,num)
9   acertoN = rep(0,num)
10
11   for(a in 1: num){
12     comeco = inferior + ((a-1)*delta)
13     resultados = funcao(dados,formula,comeco)
14     v[a] = comeco
15     acerto[a] = resultados[1,1]
16     acertoP[a] = resultados[1,2]
17     acertoN[a] = resultados[1,3]
18   }
19   estrutura <- data.frame(v,acerto,acertoP,acertoN)
20 }
21
22 melhorThreshold <- function(inferior,superior,delta, fit, teste){
23   num = ((superior - inferior) / delta) + 2
24   acerto = 0
25   valor = -1
26   comeco = inferior
27
28   for(a in 1: num){
29     comeco = inferior + ((a-1)*delta)
30     resultados = rep(NA,183)
31     resultados[fit<=comeco] = 0
32     resultados[fit>comeco] = 1
33     condicaoT <- resultados == teste
34
35     mediaAuxT = mean(teste==resultados,na.rm=TRUE) ##aqui pode-se utilizar depois
36     ##algo diferente tendo em conta aquilo q disse na analise
37     if(mediaAuxT >= 0.65){
38       mediaAuxP = length(resultados[!is.na(condicaoT) & condicaoT & resultados==0])/length(resultados[!is.na
          (teste) & teste == 0])
39
40       if(mediaAuxP >= 0.65){

```

```

41     mediaAuxN = length(resultados[!is.na(condicaoT) & condicaoT & resultados==1])/length(resultados[!is.
      na(teste) & teste == 1])
42
43     if(mediaAuxN >= 0.65){
44         mediaAux = (2*mediaAuxT) + mediaAuxP + mediaAuxN ##aqui pode-se
45         ##truncar os valores para x casas decimais
46
47         if(mediaAux > acerto){
48             acerto = mediaAux
49             valor = comeco
50         }
51     }
52 }
53
54 }
55 }
56
57 }
58 resultados = rep(NA,183)
59 resultados[fit<=valor] = 0
60 resultados[fit>valor] = 1
61 condicaoT <- resultados == teste
62
63 acertoT = mean(teste==resultados,na.rm=TRUE) ##aqui pode-se utilizar depois
64
65 acertoN = length(resultados[!is.na(condicaoT) & condicaoT & resultados==0])/length(resultados[!is.na(teste
      ) & teste == 0])
66
67 acertoP = length(resultados[!is.na(condicaoT) & condicaoT & resultados==1])/length(resultados[!is.na(teste
      ) & teste == 1])
68
69 acerto = acerto / 4
70 estrutura <- data.frame(valor, acertoT, acertoN, acertoP,acerto)
71 }
72
73 geraKNNCV <- function(treinoX, treinoY, inicio, fim){
74     num = fim - inicio + 1
75     v = rep(0,num)
76     t = rep(0,num)
77     fp = rep(0,num)
78     fn = rep(0,num)
79     for(a in 1:num){
80         i = inicio + a - 1
81         set.seed(1)
82         resultados = knn.cv(treinoX[,i],treinoY,k=a)
83
84         v[a] = i
85         v[a] = i
86         mediaAux = mean(treinoY==resultados,na.rm=TRUE) ##aqui pode-se utilizar depois
87         ##algo diferente tendo em conta aquilo q disse na analise
88         t[a] = mediaAux
89
90         condicaoT <- resultados == treinoY
91         divide = length(resultados[!is.na(treinoY) & treinoY == 0])
92         if(divide != 0){
93             mediaAux =length(resultados[!is.na(condicaoT) & condicaoT & resultados==0])/divide
94             fn[a] = mediaAux
95         }
96         divide = length(resultados[!is.na(treinoY) & treinoY == 1])

```



```

97
98   if(divide!=0){
99     mediaAux = length(resultados[!is.na(condicaoT) & condicaoT & resultados==1])/divide
100    fp[a] = mediaAux
101  }
102
103  }
104  estrutura <- data.frame(v,t,fp,fn)
105 }
106
107 melhorIndice = function(dados){
108   data = dados
109   indice = -1
110   valor = 0
111   for(i in 1:dim(data)[1]){
112     valorAux = (2*data[i,2]) + data[i,3] + data[i,4]
113
114     if(valorAux > valor){
115       valor = valorAux
116       indice = i
117     }
118   }
119
120   resultado = data.frame(indice,valor)
121
122 }
123
124 glmCV = function(dados,formula,thr){
125   treino = dados[complete.cases(dados),]
126
127   quantos = dim(treino)[1]/10
128   acerto = rep(NA,quantos)
129   acertoP = rep(NA,quantos)
130   acertoN = rep(NA,quantos)
131
132   for(a in 1:quantos){
133     teste = sample(dim(treino)[1],10)
134     dadosTeste = treino[teste, ]
135     treino = treino[-teste, ]
136
137     if(a == 1){
138
139       glm.fit = glm(formula, data=treino, family=binomial)
140       glm.probs=predict(glm.fit, newdata=dadosTeste, type="response")
141       resultados = ifelse(glm.probs > thr, 1, 0)
142       treinoY = dadosTeste[,1]
143
144       mediaAux = mean(treinoY==resultados,na.rm=TRUE) ##aqui pode-se utilizar depois
145       ##algo diferente tendo em conta aquilo q disse na analise
146       acerto[a] = mediaAux
147
148       condicaoT <- resultados == treinoY
149       divide = length(resultados[!is.na(treinoY) & treinoY == 0])
150       if(divide != 0){
151         mediaAux=length(resultados[!is.na(condicaoT) & condicaoT & resultados==0])/divide
152         acertoN[a] = mediaAux
153       }
154       divide = length(resultados[!is.na(treinoY) & treinoY == 1])
155

```

```

156     if(divide!=0){
157         mediaAux = length(resultados[!is.na(condicaoT) & condicaoT & resultados==1])/divide
158         acertoP[a] = mediaAux
159     }
160     jaUsados = dadosTeste
161 }
162 else{
163     paraTreinar = rbind(treino,jaUsados)
164     glm.fit = glm(formula, data=paraTreinar, family=binomial)
165     glm.probs=predict(glm.fit, newdata=dadosTeste, type="response")
166     resultados = ifelse(glm.probs > thr, 1, 0)
167     treinoY = dadosTeste[,1]
168
169     mediaAux = mean(treinoY==resultados,na.rm=TRUE) ##aqui pode-se utilizar depois
170     ##algo diferente tendo em conta aquilo q disse na analise
171     acerto[a] = mediaAux
172
173     condicaoT <- resultados == treinoY
174     divide = length(resultados[!is.na(treinoY) & treinoY == 0])
175     if(divide != 0){
176         mediaAux =length(resultados[!is.na(condicaoT) & condicaoT & resultados==0])/divide
177
178         acertoN[a] = mediaAux
179     }
180     divide = length(resultados[!is.na(treinoY) & treinoY == 1])
181     if(divide != 0){
182         mediaAux = length(resultados[!is.na(condicaoT) & condicaoT & resultados==1])/divide
183
184         acertoP[a] = mediaAux
185     }
186     jaUsados = rbind(jaUsados,dadosTeste)
187 }
188 }
189 acertoMedio = mean(acerto, na.rm = TRUE)
190 acertoMedioP = mean(acertoP, na.rm = TRUE)
191 acertoMedioN = mean(acertoN, na.rm = TRUE)
192
193 data.frame(acertoMedio,acertoMedioP,acertoMedioN)
194 }
195
196 qdaCV = function(dados,fitFrom,thr){
197     treino = dados[complete.cases(dados),]
198
199     quantos = dim(treino)[1]/10
200     acerto = rep(NA,quantos)
201     acertoP = rep(NA,quantos)
202     acertoN = rep(NA,quantos)
203
204     for(a in 1:quantos){
205         set.seed(1)
206         teste = sample(dim(treino)[1],10)
207         dadosTeste = treino[teste, ]
208         treino = treino[-teste, ]
209
210         if(a == 1){
211             set.seed(1)
212             qda.fit=qda(fitFrom,data=treino)
213             qda.pred=predict(qda.fit,dadosTeste)
214             resultadosAux = qda.pred$posterior[,2]

```

```

215     resultados = ifelse(resultadosAux >= thr, 1, 0)
216     treinoY = dadosTeste[,1]
217
218     mediaAux = mean(treinoY==resultados,na.rm=TRUE) ##aqui pode-se utilizar depois
219     ##algo diferente tendo em conta aquilo q disse na analise
220     acerto[a] = mediaAux
221
222     condicaoT <- resultados == treinoY
223     divide = length(resultados[!is.na(treinoY) & treinoY == 0])
224     if(divide != 0){
225         mediaAux =length(resultados[!is.na(condicaoT) & condicaoT & resultados==0])/divide
226         acertoN[a] = mediaAux
227     }
228     divide = length(resultados[!is.na(treinoY) & treinoY == 1])
229
230     if(divide!=0){
231         mediaAux = length(resultados[!is.na(condicaoT) & condicaoT & resultados==1])/divide
232         acertoP[a] = mediaAux
233     }
234     jaUsados = dadosTeste
235 }
236 else{
237     paraTreinar = rbind(treino,jaUsados)
238     qda.fit=qda(fitFrom,data=paraTreinar)
239     qda.pred=predict(qda.fit,dadosTeste)
240     resultadosAux = qda.pred$posterior[,2]
241     resultados = ifelse(resultadosAux >= thr, 1, 0)
242     treinoY = dadosTeste[,1]
243
244     mediaAux = mean(treinoY==resultados,na.rm=TRUE) ##aqui pode-se utilizar depois
245     ##algo diferente tendo em conta aquilo q disse na analise
246     acerto[a] = mediaAux
247
248     condicaoT <- resultados == treinoY
249     divide = length(resultados[!is.na(treinoY) & treinoY == 0])
250     if(divide != 0){
251         mediaAux =length(resultados[!is.na(condicaoT) & condicaoT & resultados==0])/divide
252
253         acertoN[a] = mediaAux
254     }
255     divide = length(resultados[!is.na(treinoY) & treinoY == 1])
256     if(divide != 0){
257         mediaAux = length(resultados[!is.na(condicaoT) & condicaoT & resultados==1])/divide
258
259         acertoP[a] = mediaAux
260     }
261     jaUsados = rbind(jaUsados,dadosTeste)
262 }
263 }
264 acertoMedio = mean(acerto, na.rm = TRUE)
265 acertoMedioP = mean(acertoP, na.rm = TRUE)
266 acertoMedioN = mean(acertoN, na.rm = TRUE)
267
268 data.frame(acertoMedio,acertoMedioP,acertoMedioN)
269 }
270
271 ldaCV = function(dados,fitFrom,thr){
272     treino = dados[complete.cases(dados),]
273

```

```

274 quantos = dim(treino)[1]/10
275 acerto = rep(NA,quantos)
276 acertoP = rep(NA,quantos)
277 acertoN = rep(NA,quantos)
278
279 for(a in 1:quantos){
280   set.seed(1)
281   teste = sample(dim(treino)[1],10)
282   dadosTeste = treino[teste, ]
283   treino = treino[-teste, ]
284
285   if(a == 1){
286     set.seed(1)
287     qda.fit=lda(fitFrom,data=treino)
288     qda.pred=predict(qda.fit,dadosTeste)
289     resultadosAux = qda.pred$posterior[,2]
290     resultados = ifelse(resultadosAux >= thr, 1, 0)
291     treinoY = dadosTeste[,1]
292
293     mediaAux = mean(treinoY==resultados,na.rm=TRUE) ##aqui pode-se utilizar depois
294     ##algo diferente tendo em conta aquilo q disse na analise
295     acerto[a] = mediaAux
296
297     condicaoT <- resultados == treinoY
298     divide = length(resultados[!is.na(treinoY) & treinoY == 0])
299     if(divide != 0){
300       mediaAux =length(resultados[!is.na(condicaoT) & condicaoT & resultados==0])/divide
301       acertoN[a] = mediaAux
302     }
303     divide = length(resultados[!is.na(treinoY) & treinoY == 1])
304
305     if(divide!=0){
306       mediaAux = length(resultados[!is.na(condicaoT) & condicaoT & resultados==1])/divide
307       acertoP[a] = mediaAux
308     }
309     jaUsados = dadosTeste
310   }
311   else{
312     paraTreinar = rbind(treino,jaUsados)
313     qda.fit=lda(fitFrom,data=paraTreinar)
314     qda.pred=predict(qda.fit,dadosTeste)
315     resultadosAux = qda.pred$posterior[,2]
316     resultados = ifelse(resultadosAux >= thr, 1, 0)
317     treinoY = dadosTeste[,1]
318
319     mediaAux = mean(treinoY==resultados,na.rm=TRUE) ##aqui pode-se utilizar depois
320     ##algo diferente tendo em conta aquilo q disse na analise
321     acerto[a] = mediaAux
322
323     condicaoT <- resultados == treinoY
324     divide = length(resultados[!is.na(treinoY) & treinoY == 0])
325     if(divide != 0){
326       mediaAux =length(resultados[!is.na(condicaoT) & condicaoT & resultados==0])/divide
327
328       acertoN[a] = mediaAux
329     }
330     divide = length(resultados[!is.na(treinoY) & treinoY == 1])
331     if(divide != 0){
332       mediaAux = length(resultados[!is.na(condicaoT) & condicaoT & resultados==1])/divide

```

```

333
334     acertoP[a] = mediaAux
335   }
336   jaUsados = rbind(jaUsados,dadosTeste)
337 }
338 }
339 acertoMedio = mean(acerto, na.rm = TRUE)
340 acertoMedioP = mean(acertoP, na.rm = TRUE)
341 acertoMedioN = mean(acertoN, na.rm = TRUE)
342
343 data.frame(acertoMedio,acertoMedioP,acertoMedioN)
344 }
345
346 normaliza = function(dados){
347   data = dados
348   for(i in 1:dim(data)[2]){
349     if(class(data[,i])!="factor"){
350       data[,i] <- (data[,i] - min(data[,i], na.rm = TRUE)) / (max(data[,i],na.rm = TRUE)-min(data[,i],na.rm
        = TRUE))
351     }
352   }
353
354   resultado = data
355 }
356
357 consomeValores = function(dados){
358   indice = rep(NA,dim(dados)[1])
359   k = rep(NA,dim(dados)[1])
360   valor = rep(NA,dim(dados)[1])
361   for(i in 1:dim(dados)[1]){
362     indice[i] = dados[i,5]
363     k[i] = dados[i,1]
364     valor[i] = (2*dados[i,2]) + dados[i,3] + dados[i,4]
365   }
366
367   data.frame(indice,k,valor)
368 }
369
370 transformaBinaria = function(dados){
371   binaria <- rep(NA,length(dados))
372   binaria[dados > 7] = 1
373   binaria[dados <= 7] = 0
374   binaria
375 }

```



AA1

Apresentação do trabalho prático

14 de Dezembro de 2018

Dataset



Informações do Dataset:

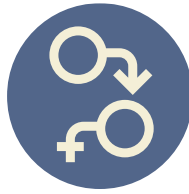
- ✓ Nome: Diabetes
- ✓ Origem: EUA
- ✓ Data: 1997
- ✓ Quantidade: 403 obs. e 19 var

Variáveis

- ✓ Stab.glu
- ✓ Chol
- ✓ Time.ppn
- ✓ Hip
- ✓ Weight
- ✓ Height

- ✓ Bp.1s
- ✓ Bp.1d
- ✓ Bp.2s
- ✓ Bp.2d
- ✓ Gly.hb

- ✓ Ratio
- ✓ Frame
- ✓ Location
- ✓ Hdl
- ✓ Gender
- ✓ Waist

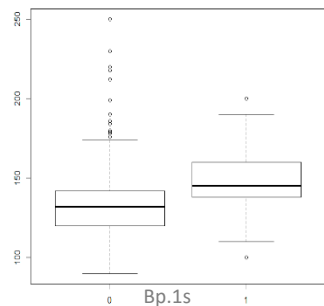
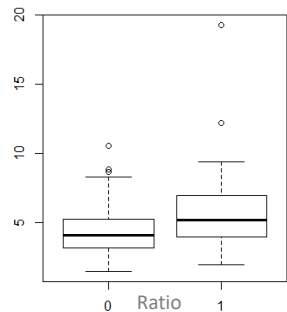
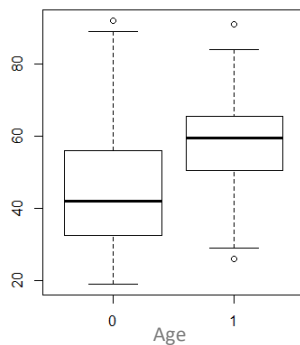
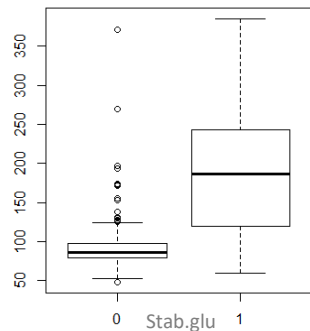




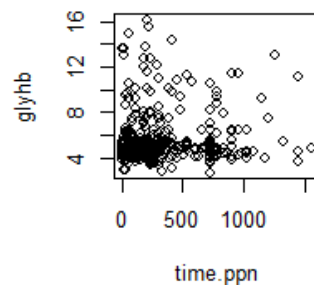
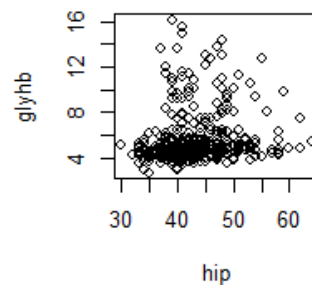
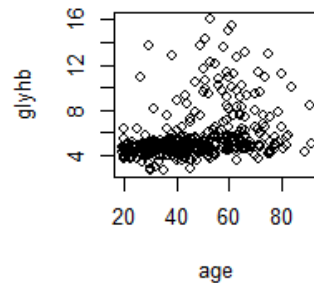
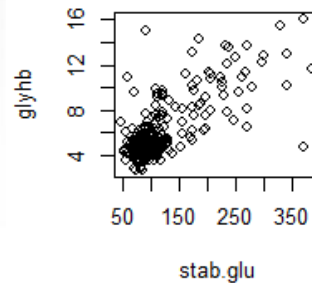
Análise exploratória

Análise de influencia

Var Binária



Gly.hb



Ranking variáveis

✓ Var Binária

1. Stab.glu
2. Age
3. Bp.1s
4. Racio
5. Waist
6. Weight
7. Hip
8. Hdl
9. Chol
10. Bp.1d

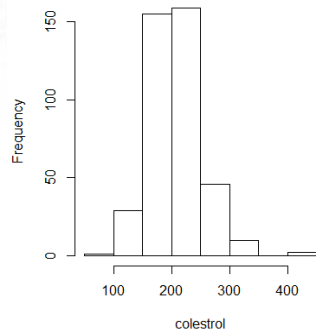
✓ Gly.hb

1. Stab.glu
2. Age
3. Time.ppn
4. Hip
5. Bp.1s
6. Waist
7. Bp.1d
8. Chol
9. Height
10. Weighth

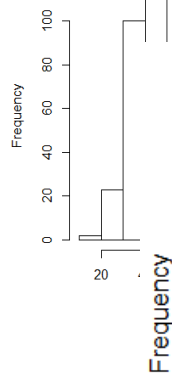


Análise de normalidade

Histogram of colesterol



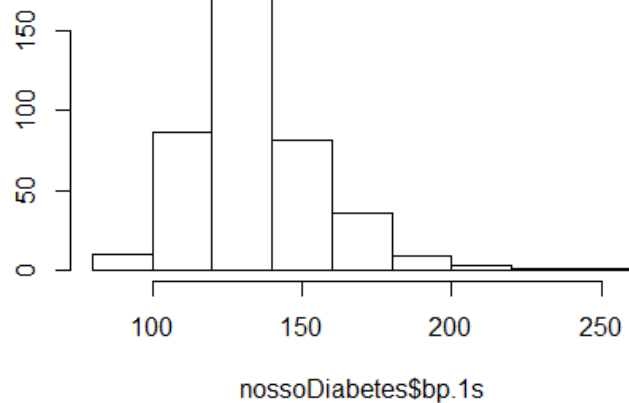
Histogram of hdl



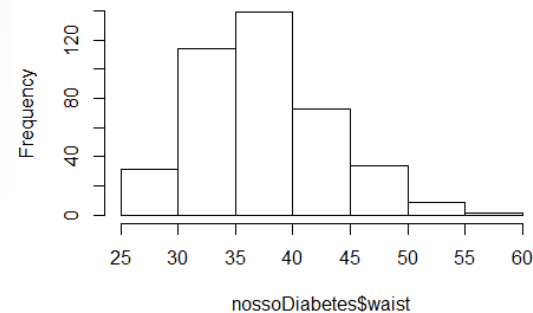
Histogram of nossoDiabetes\$bp.1d



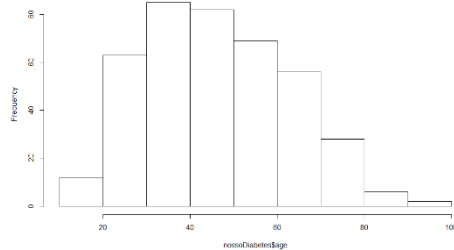
Histogram of nossoDiabetes\$bp.1s



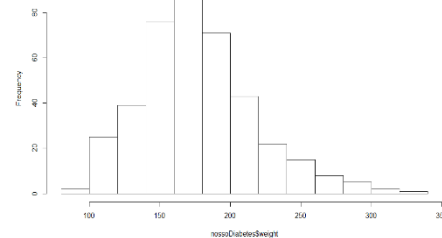
Histogram of nossoDiabetes\$waist



Histogram of nossoDiabetes\$age



Histogram of nossoDiabetes\$weight





Seleção do modelo

LM vs GLM vs QDA vs KNN vs LDA

Regressão linear (LM)

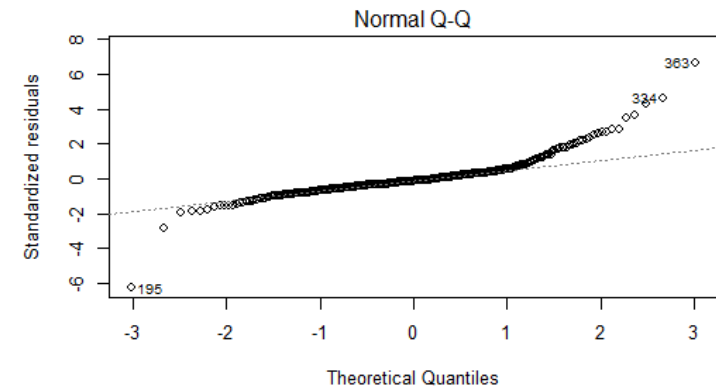
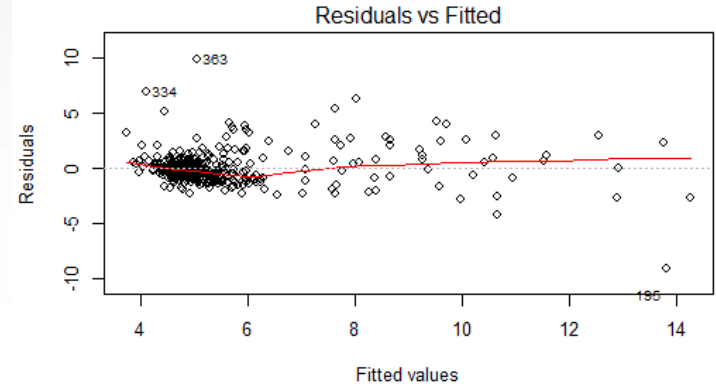
Modelos:

Dataset:

- `stab.glu`
Adj. R-squared: 0.5602
- `stab.glu^2`
Adj. R-squared: 0.5678
- `stab.glu^3`
Adj. R-squared: 0.5877

CV(K=10):

- `stab.glu`
Adj. R-squared: 0.5678
- `stab.glu^2`
Adj. R-squared: 0.5621
- `stab.glu^3`
Adj. R-squared: 0.5581

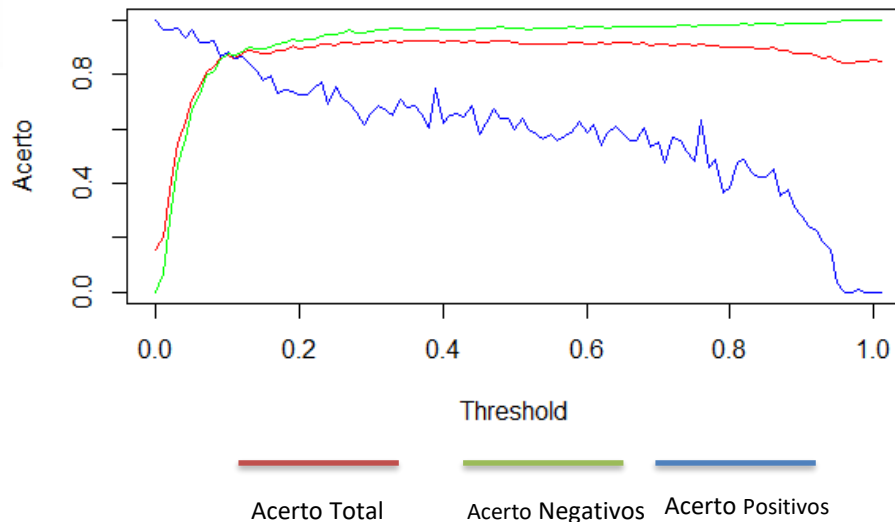


Regressão logística (GLM)

Modelos:

Fórmula	Thr	Acerto
Stab.glu^2	0.39	3.5714
Stab.glu^2+ratio	0.39	3.5714
Stab.glu+ratio + stab.glu x ratio	0.39	3.5520

Gráfico Threshold:



Nossa abordagem

Variáveis	K	Acerto ponderado
Stab.glu	14	3.4825
Stab.glu+age	9	3.4707
Stab.glu+age+bp.1s	10	3.5320
Stab.glu+age+bp.1s+ratio	12	3.5316
Stab.glu+age+bp.1s+ratio+waist	10	3.5307

Regsubsets

Variáveis	K	Acerto ponderado
Stab.glu+chol	9	3.4902
Stab.glu+ratio+age	16	3.4766
Stab.glu+ratio+age+time.ppn	3	3.4286
(F)stab.glu+chol+age	8	3.4684
(F)stab.glu+chol+age+time.ppn	4	3.4014
(B)stab.glu+ratio	15	3.4820



QDA E LDA

QDA

Variáveis	Threshold	Acerto Ponderado
Age	0.22	2.8265
Age+bp.1s	0.24	2.8260
Age+bp.1s+waist	0.18	2.9115

LDA

Variáveis	Threshold	Acerto Ponderado
Age	0.18	2.8578
Age+bp.1s	0.26	2.8039
Age+bp.1s+waist	0.16	2.9364



Seleção melhor modelo (CV)

Modelo	Regressão linear	Regressão logística	KNN
Fórmula	Stab.glu + age	stab.glu^2	stab.glu + age + bp.1s
Threshold	>7	0.39	K=10
Acerto ponderado	$adj.r^2 = 56.23\%$	3.5714	3.5320



Seleção melhor modelo

(Teste e Treino)

Modelo	Regressão linear	Regressão logística	KNN
Fórmula	Stab.glu + age	stab.glu^2	stab.glu + age + bp.1s
Threshold	>7	0.39	K=10
Acerto ponderado	3.6318	3.5991	3.5665

Modelo escolhido

Modelo: Regressão linear

- ✓ Fórmula: $\text{glyhb} = 1.693 + 0.027 * \text{stab.glu} + 0.020 * \text{age}$
- ✓ Adjusted R^2 : 56.23%
- ✓ Acerto ponderado: 3.63
- ✓ Acerto total: 95.88%
- ✓ Acerto positivos: 71.43%
- ✓ Acerto negativos: 100%





Discussão de resultados

- ? **Será que o colesterol/pressão arterial/tempo após refeição/fatores corporais afetam os diabetes?**
- ? **Qual fator corporal explica melhor o valor da diabete?**
- ? **Quais fatores influenciam mais o resultado final?**
- ? **De que forma os fatores selecionados para a explicação dos resultados o influenciam?
(crescentemente, decrescentemente, linearmente)**

Discussão de resultados

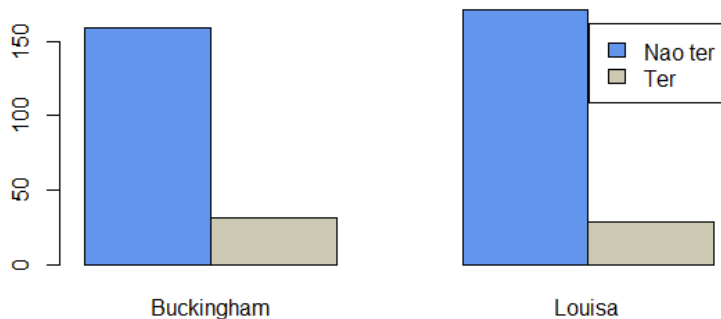
- ? Qual a probabilidade (ou confiança no resultado) de uma pessoa com as características X (por exemplo colesterol=180, altura=175, peso=67, etc.) ter diabetes?

Stab.glu	Age	Confiança
151	60	0.5%
171	60	29.9%
191	60	54.7%
211	60	73.8%
231	60	86.3%

Stab.glu	Age	Confiança
191	20	16.7%
191	40	37%
191	60	54.7%
191	75	66%

Discussão de resultados

- ? Qual a taxa de incidência em pessoas com menos e com mais de 50 anos?
- ? Qual a cidade apresenta maior incidência? (visto serem só dois podemos comparar)
- ? O resultado é mais exato utilizando um modelo de classificação ou de regressão (e de seguida classificando)?



Incidência em pessoas com mais/menos do que 50 anos

- ✓ 94.22% das pessoas que têm menos de 50 anos não têm diabetes
- ✓ 70,4% das pessoas que têm mais de 50 anos não têm diabetes
- ✓ 42.55% dos que não têm diabetes têm mais de 50 anos
- ✓ 83.54% dos que têm diabetes têm mais de 50 anos



Fim

Q & A