Hands-on R - Regressão Linear Múltipla

Rodrigo Heldt - rodrigoheldt@gmail.com

IMED - HOPEAD - Nov 13, 2017

Pré-processamento dos dados para análises posteriores

1. Limpando os objetos do ambiente

```
rm(list=ls())
```

2. Carregando (e instalando) pacotes de funções necessários

```
# install.packages('caTools')
# Esse comando deve ser rodado apenas na primeira vez que você usar o pacote para
# instalá-lo em seu computador. Após instalado, a cada vez que for utilizá-lo basta
# carregar o pacote usando a função library

library(caTools)
```

4. Importando a base de dados de um arquivo csv

```
# Usar a funcao read.csv

base <- read.csv("C:/Users/Rodrigo/Dropbox/Hands-on R Workshop/Parte 2 - Pre-processamento dos dados/Da
                 sep = ",")
```

4. Codificando variáveis categóricas

```
# Usar a função as.factor para que o R transforme as variáveis desejadas na classe tipo
# fator (categórico)

base$Mulher = as.factor(base$Mulher)
base$Educacao = as.factor(base$Educacao)
base$PartTime = as.factor(base$PartTime)
```

5. Definindo variáveis numéricas

```
# Usar a função as.numeric para que o R transforme as variáveis desejadas na classe tipo
# numeric (número)

base$SalarioPorHora = as.numeric(base$SalarioPorHora)
base$LogSalarioPorHora = as.numeric(base$LogSalarioPorHora)
base$Idade = as.numeric(base$Idade)
```

6. Divindindo a base em uma parte para treinamento dos modelos e outra para teste das previsões do modelo treinado

```
set.seed(123)
split = sample.split(base$SalarioPorHora, SplitRatio = 0.8)
baseTrain = subset(base, split == TRUE)
baseTest = subset(base, split == FALSE)
```

Regressão Linear Múltipla

0 Carregar pacotes

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

1. Ajustando um modelo de regressão linear múltipla usando a base de treinamento

1.1 Modelo: SalarioPorHora = alpha + Idade + Mulher + Educacao + PartTime

```
reg.multipla = lm(formula = SalarioPorHora ~ Idade + Mulher + Educacao + PartTime,
                  data = baseTrain)
summary(reg.multipla)
```

```
##
## Call:
## lm(formula = SalarioPorHora ~ Idade + Mulher + Educacao + PartTime,
##     data = baseTrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -77.008 -20.379  -3.589  15.497 150.258
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -43.9237     7.2898  -6.025 3.75e-09 ***
## Idade         3.5867     0.1758  20.401  < 2e-16 ***
## Mulher1      -0.1546     3.4266  -0.045    0.964
## Educacao2    19.6815     3.8394   5.126 4.56e-07 ***
## Educacao3    46.2056     4.1936  11.018  < 2e-16 ***
## Educacao4   103.1841     4.8220  21.399  < 2e-16 ***
## PartTime1   -42.6087     4.3262  -9.849  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.98 on 412 degrees of freedom
## Multiple R-squared:  0.7143, Adjusted R-squared:  0.7101
## F-statistic: 171.6 on 6 and 412 DF,  p-value: < 2.2e-16
```

```r
# Os resultados acima indicam que a variável Mulher não é significativa
# Assim, ajusta-se um novo modelo sem essa variável

reg.multipla = lm(formula = SalarioPorHora ~ Idade + Educacao + PartTime,
                  data = baseTrain)
summary(reg.multipla)
```

```
##
## Call:
## lm(formula = SalarioPorHora ~ Idade + Educacao + PartTime, data = baseTrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -77.134 -20.378  -3.665  15.462 150.101
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -44.0409     6.8036  -6.473 2.72e-10 ***
## Idade         3.5882     0.1722  20.834  < 2e-16 ***
## Educacao2    19.6847     3.8341   5.134 4.38e-07 ***
## Educacao3    46.2341     4.1407  11.166  < 2e-16 ***
## Educacao4   103.2385     4.6635  22.138  < 2e-16 ***
## PartTime1   -42.6694     4.1068 -10.390  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.95 on 413 degrees of freedom
## Multiple R-squared:  0.7143, Adjusted R-squared:  0.7108
## F-statistic: 206.5 on 5 and 413 DF,  p-value: < 2.2e-16
```

2. Prevendo os valores para a base de teste a partir dos modelos ajustes

```r
## Prevendo os valores do SalarioPorHora para a base de teste
y_pred_multipla = predict(reg.multipla, newdata = baseTest)
```

3. Verificando a performance das previsões realizadas

```r
## Verificando medidas de precisão das previsões
# Mean Absolut Error
mae <- error(true = baseTest$SalarioPorHora, forecast = y_pred_multipla, method = "mae")
mae
```

```
## [1] 18.25697
```

```r
# Mean Absolut Percentage Error
mape <- error(true = baseTest$SalarioPorHora, forecast = y_pred_multipla, method = "mape")
mape
```

```
## [1] 19.22616
```

```r
# Root Mean Square Error
rmse <- error(true = baseTest$SalarioPorHora, forecast = y_pred_multipla, method = "rmse")
rmse
```

```
## [1] 23.17781
```

4. Exercício

E1 - Ajuste uma regressão multipla com a variável dependente LogSalarioPorHora e as variáveis independentes Idade, Mulher, Educacao e PartTime na base de treinamento

E2 - Faça a previsão do LogSalarioPorHora a partir das variaveis presentes no modelo final para a base de teste

E3 - Cheque o mean absolut percentage error (mape) entre os valores reais e os valores previstos de LogSalarioPorHora