00-CargaTransformacao.R

rique

2020-12-21

```
# Coleta e Transformação de Dados

# Este código contém comandos para filtrar e transformar os dados de aluguel de bikes,
# dados que estão em nosso dataset.

# Este código foi criado para executar tanto no Azure, quanto no RStudio.
# Para executar no Azure, altere o valor da variavel Azure para TRUE.
# Se o valor for FALSE, o código sera executado no RStudio

# Obs: Caso tenha problemas com a acentuação, consulte este link:
# https://support.rstudio.com/hc/en-us/articles/200532197-Character-Encoding

# Configurando o diretório de trabalho
# Coloque entre aspas o diretório de trabalho que você está usando no seu computador
# Não use diretórios com espaço no nome
# setwd("C:/FCD/BigDataRAzure/Cap14/Projeto")
# getwd()


# Variável que controla a execução do script
Azure <- FALSE

# Execução de acordo com o valor da variável Azure
if(Azure){
  source("src/Tools.R")
  bikes <- maml.mapInputPort(1)
  bikes$dteday <- set.asPOSIXct(bikes)
}else{
  source("src/Tools.R")
  bikes <- read.csv("bikes.csv", sep = ",", header = TRUE, stringsAsFactors = FALSE )

  # Selecionar as variáveis que serão usadas
  cols <- c("dteday", "mnth", "hr", "holiday",
            "workingday", "weathersit", "temp",
            "hum", "windspeed", "cnt")

  # Criando um subset dos dados
  bikes <- bikes[, cols]

  # Transformar o objeto de data
  bikes$dteday <- char.toPOSIXct(bikes)
```

```
  # Esta linha acima gera dois valores NA
  # Esta linha abaixo corrige
  bikes <- na.omit(bikes)

  # Normalizar as variaveis preditoras numericas
  cols <- c("temp", "hum", "windspeed")
  bikes[, cols] <- scale(bikes[, cols])
}

#?scale
str(bikes)
```

```
## 'data.frame':    17377 obs. of  10 variables:
##  $ dteday    : POSIXct, format: "2011-01-01 00:00:00" "2011-01-01 01:00:00" ...
##  $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ hr        : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ workingday: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ weathersit: int  1 1 1 1 1 2 1 1 1 1 ...
##  $ temp      : num  -1.33 -1.44 -1.44 -1.33 -1.33 ...
##  $ hum       : num  0.947 0.895 0.895 0.636 0.636 ...
##  $ windspeed : num  -1.55 -1.55 -1.55 -1.55 -1.55 ...
##  $ cnt       : int  16 40 32 13 1 1 2 3 8 14 ...
##  - attr(*, "na.action")= 'omit' Named int  6803 15692
##   ..- attr(*, "names")= chr  "6803" "15692"
```

```
View(bikes)

# Criar uma nova variável para indicar dia da semana (workday)
bikes$isWorking <- ifelse(bikes$workingday & !bikes$holiday, 1, 0)

# Adicionar uma coluna com a quantidade de meses, o que vai ajudar a criar o modelo
bikes <- month.count(bikes)

# Criar um fator ordenado para o dia da semana, comecando por segunda-feira
# Neste fator eh convertido para ordenado numérico para ser compativel com os tipos de dados do Azure M
bikes$dayWeek <- as.factor(weekdays(bikes$dteday))


############ ATENÇÃO ############

# ==> Analise o dataframe bikes.
# Se os nomes dos dias da semana estiverem em português na coluna bikes$dayWeek,
# execute o Bloco1 abaixo, caso contrátio, execute o Bloco2 com os nomes em inglês.
# Execute um bloco ou o outro.
str(bikes$dayWeek)
```

```
##  Factor w/ 7 levels "domingo","quarta-feira",..: 4 4 4 4 4 4 4 4 4 4 ...
```

```
# Bloco1
# Se o seu sistema operacional estiver em português, execute o comando abaixo.
bikes$dayWeek <- as.numeric(ordered(bikes$dayWeek,
                                    levels = c("segunda-feira",
                                               "terça-feira",
                                               "quarta-feira",
```

```
                                            "quinta-feira",
                                            "sexta-feira",
                                            "sábado",
                                            "domingo")))

# Bloco2
# Se o seu sistema operacional estiver em inglês, execute o comando abaixo.
bikes$dayWeek <- as.numeric(ordered(bikes$dayWeek,
                                    levels = c("Monday",
                                            "Tuesday",
                                            "Wednesday",
                                            "Thursday",
                                            "Friday",
                                            "Saturday",
                                            "Sunday")))

# Agora os dias da semana devem estar como valores numéricos
# Se estiverem como valores NA, volte e verifique se você seguiu as instruções acima.
str(bikes$dayWeek)
```

```
##  num [1:17377] NA NA NA NA NA NA NA NA NA NA ...
```

```
str(bikes)
```

```
## 'data.frame':    17377 obs. of  13 variables:
##  $ dteday    : POSIXct, format: "2011-01-01 00:00:00" "2011-01-01 01:00:00" ...
##  $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ hr        : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ workingday: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ weathersit: int  1 1 1 1 1 2 1 1 1 1 ...
##  $ temp      : num  -1.33 -1.44 -1.44 -1.33 -1.33 ...
##  $ hum       : num  0.947 0.895 0.895 0.636 0.636 ...
##  $ windspeed : num  -1.55 -1.55 -1.55 -1.55 -1.55 ...
##  $ cnt       : int  16 40 32 13 1 1 2 3 8 14 ...
##  $ isWorking : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ monthCount: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ dayWeek   : num  NA NA NA NA NA NA NA NA NA NA ...
##  - attr(*, "na.action")= 'omit' Named int  6803 15692
##   ..- attr(*, "names")= chr  "6803" "15692"
```

```
# Adiciona uma variável com valores únicos para o horário do dia em dias de semana e dias de fim de sem
# Com isso diferenciamos as horas dos dias de semana, das horas em dias de fim de semana
bikes$workTime <- ifelse(bikes$isWorking, bikes$hr, bikes$hr + 24)

# Transforma os valores de hora na madrugada, quando a demanda por bibicletas é praticamente nula
bikes$xformHr <- ifelse(bikes$hr > 4, bikes$hr - 5, bikes$hr + 19)

# Adiciona uma variável com valores únicos para o horário do dia para dias de semana e dias de fim de s
# Considerando horas da madrugada
bikes$xformWorkHr <- ifelse(bikes$isWorking, bikes$xformHr, bikes$xformHr + 24)

# str(bikes)
# View(bikes)
# O trabalho que fizemos até aqui também é chamado de Feature Engineering ou
```

```r
# Engenharia de Atributos

# Gera saída no Azure ML
if(Azure) maml.mapOutputPort('bikes')
```