bimbo_clear.R

gquai

2020-01-15

```
# DATA SCIENCE ACADEMY
# Big Data Analytics com R e Microsoft Azure Machine Learning
#
# Model to accurately predict inventory demand based on data sales histories
#
# Gabriel Quaiotti
# Dez 2019
#
# In this competition, you will forecast the demand of a product for a given week, at a
# particular store.
# The dataset you are given consists of 9 weeks of sales transactions in Mexico. Every week,
# there are delivery
# trucks that deliver products to the vendors. Each transaction consists of sales and returns.
# Returns are the products that are unsold and expired. The demand for a product in a certain
# week is defined as the sales this week subtracted by the return next week.
#
#
# The train and test dataset are split based on time, as well as the public and private
# leaderboard dataset split.
#
#
# Things to note:
#
# There may be products in the test set that don't exist in the train set. This is the expected
# behavior of inventory data,
# since there are new products being sold all the time. Your model should be able to accommodate
# this.
#
# The adjusted demand (Demanda_uni_equil) is always >= 0 since demand should be either 0 or a
# positive value. The reason that Venta_uni_hoy - Dev_uni_proxima
# sometimes has negative values is that the returns records sometimes carry over a few weeks.

# File descriptions
# train.csv - the training set
# test.csv - the test set
# sample_submission.csv - a sample submission file in the correct format
# cliente_tabla.csv - client names (can be joined with train/test on Cliente_ID)
# producto_tabla.csv - product names (can be joined with train/test on Producto_ID)
# town_state.csv - town and state (can be joined with train/test on Agencia_ID)

# Data fields
# Semana - Week number (From Thursday to Wednesday)
# Agencia_ID - Sales Depot ID
# Canal_ID - Sales Channel ID
# Ruta_SAK - Route ID (Several routes = Sales Depot)
# Cliente_ID - Client ID
# NombreCliente - Client name
```

```r
# Producto_ID - Product ID
# NombreProducto - Product Name
# Venta_uni_hoy - Sales unit this week (integer)
# Venta_hoy - Sales this week (unit: pesos)
# Dev_uni_proxima - Returns unit next week (integer)
# Dev_proxima - Returns next week (unit: pesos)
# Demanda_uni_equil - Adjusted Demand (integer) (This is the target you will predict)

setwd('D:/Github/DSA_BIMBO_INVENTORY')

library(data.table)

v_c_file_train <- "dataset/train_transform.csv"
v_c_file_train_out <- "dataset/train_clean.csv"

###################################
# TRAIN DATASET
###################################
train <- data.table::fread(file = v_c_file_train)

str(train)
```

```
## Classes 'data.table' and 'data.frame':   74180464 obs. of  13 variables:
##  $ Semana           : int  3 7 8 3 7 8 9 7 3 4 ...
##  $ Agencia_ID       : int  2061 2061 2061 2061 2061 2061 2655 2655 2061 2061 ...
##  $ Canal_ID         : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ Ruta_SAK         : int  7212 7212 7212 7212 7212 7212 4189 4189 7212 7212 ...
##  $ Cliente_ID       : int  26 26 26 26 26 26 26 26 26 26 ...
##  $ Producto_ID      : int  1182 1182 1182 4767 4767 4767 30235 30314 31393 31393 ...
##  $ Venta_uni_hoy    : int  39 35 42 84 42 42 96 48 20 16 ...
##  $ Venta_hoy        : num  562 504 605 1277 638 ...
##  $ Dev_uni_proxima  : int  7 35 42 42 0 0 0 0 0 0 ...
##  $ Dev_proxima      : num  NA 504 605 638 NA ...
##  $ Demanda_uni_equil: int  32 0 0 42 42 42 96 48 20 16 ...
##  $ unit_price       : num  14.4 14.4 14.4 15.2 15.2 ...
##  $ dev_unit_price   : num  NA 14.4 14.4 15.2 NA NA NA NA NA NA ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```r
summary(train[,Venta_hoy])
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##      0.0     16.8     30.0     68.5     56.1 647360.0
```

```r
# Removing outliers (by Cliente_ID and Producto_ID)
train[, mean := mean(Venta_uni_hoy), by=list(Cliente_ID, Producto_ID)]
train[, sd := sd(Venta_uni_hoy), by=list(Cliente_ID, Producto_ID)]

repeat {
  train <- train[Venta_uni_hoy >= (mean - 3 * sd) & Venta_uni_hoy <= (mean + 3 * sd)]

  train[, mean := mean(Venta_uni_hoy), by=list(Cliente_ID, Producto_ID)]
  train[, sd := sd(Venta_uni_hoy), by=list(Cliente_ID, Producto_ID)]
```

```
  if (nrow(train[Venta_uni_hoy < (mean - 3 * sd) & Venta_uni_hoy > (mean + 3 * sd)]) == 0) {
    break
  }
}

train$mean <- NULL
train$sd <- NULL

str(train)
```

```
## Classes 'data.table' and 'data.frame':   65055721 obs. of  13 variables:
##  $ Semana           : int  3 7 8 3 7 8 3 4 5 6 ...
##  $ Agencia_ID       : int  2061 2061 2061 2061 2061 2061 2061 2061 2061 2061 ...
##  $ Canal_ID         : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ Ruta_SAK         : int  7212 7212 7212 7212 7212 7212 7212 7212 7212 7212 ...
##  $ Cliente_ID       : int  26 26 26 26 26 26 26 26 26 26 ...
##  $ Producto_ID      : int  1182 1182 1182 4767 4767 4767 31393 31393 31393 31393 ...
##  $ Venta_uni_hoy    : int  39 35 42 84 42 42 20 16 15 15 ...
##  $ Venta_hoy        : num  562 504 605 1277 638 ...
##  $ Dev_uni_proxima  : int  7 35 42 42 0 0 0 0 0 0 ...
##  $ Dev_proxima      : num  NA 504 605 638 NA ...
##  $ Demanda_uni_equil: int  32 0 0 42 42 42 20 16 15 15 ...
##  $ unit_price       : num  14.4 14.4 14.4 15.2 15.2 ...
##  $ dev_unit_price   : num  NA 14.4 14.4 15.2 NA NA NA NA NA NA ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
summary(train[,Venta_hoy])
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##      0.0     17.8     31.1     73.7     60.0 647360.0
```

```
data.table::fwrite(x = train, file = v_c_file_train_out)

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

```
##            used  (Mb) gc trigger    (Mb)   max used    (Mb)
## Ncells   601599  32.2    1238960    66.2    1238960    66.2
## Vcells 79455753 606.2 1982787007 15127.5 2161254712 16489.1
```