bimbo_test.R

gquai

2020-01-16

```
# 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
# Jan 2020
#
# 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 test 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 test 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
# test.csv - the testing 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 test/test on Cliente_ID)
# producto_tabla.csv - product names (can be joined with test/test on Producto_ID)
# town_state.csv - town and state (can be joined with test/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
# 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)
library(MASS)

v_c_file_test <- "dataset/test.csv"
v_c_file_features <- "dataset/train_features.csv"

###################################
# test DATASET
###################################
test <- data.table::fread(file = v_c_file_test)
features <- data.table::fread(file = v_c_file_features)

# Get the feature values previous calculated at train dataset
test <- merge(x = test,
              y = features[, list(prod_cust_balance = max(prod_cust_balance), prod_cust_bal_mean = max(p
              all.x = TRUE,
              by.x = c("Cliente_ID", "Producto_ID"),
              by.y = c("Cliente_ID", "Producto_ID"))

# If cannot find, set as 0
test[is.na(prod_cust_balance), prod_cust_balance := 0]
test[is.na(prod_cust_bal_mean), prod_cust_bal_mean := 0]

# Get the feature values previous calculated at train dataset
test <- merge(x = test,
              y = features[, list(prod_rout_balance = max(prod_rout_balance), prod_rout_prop = max(prod_
              all.x = TRUE,
              by.x = c("Ruta_SAK", "Producto_ID"),
              by.y = c("Ruta_SAK", "Producto_ID"))

# If cannot find, set as 0
test[is.na(prod_rout_balance), prod_rout_balance := 0]
test[is.na(prod_rout_prop), prod_rout_prop := 0]


# Get the feature values previous calculated at train dataset
test <- merge(x = test,
              y = features[, list(cust_rout_balance = max(cust_rout_balance), cust_rout_prop = max(cust_
              all.x = TRUE,
              by.x = c("Ruta_SAK", "Cliente_ID"),
              by.y = c("Ruta_SAK", "Cliente_ID"))

# If cannot find, set as 0
```

```
test[is.na(cust_rout_balance), cust_rout_balance := 0]
test[is.na(cust_rout_prop), cust_rout_prop := 0]

# Get the feature values previous calculated at train dataset
test <- merge(x = test,
              y = features[Semana == 9, list(last_week_balance = max(last_week_balance)), by = list(Ruta
              all.x = TRUE,
              by.x = c("Ruta_SAK", "Cliente_ID", "Producto_ID"),
              by.y = c("Ruta_SAK", "Cliente_ID", "Producto_ID"))

# If cannot find, set as 0
test[is.na(last_week_balance), last_week_balance := 0]

# Normalize
test[, prod_cust_balance := (prod_cust_balance - min(prod_cust_balance)) / (max(prod_cust_balance) - mi
test[, prod_cust_bal_mean := (prod_cust_bal_mean - min(prod_cust_bal_mean)) / (max(prod_cust_bal_mean)

test[, prod_rout_balance := (prod_rout_balance - min(prod_rout_balance)) / (max(prod_rout_balance) - mi

test[, cust_rout_balance := (cust_rout_balance - min(cust_rout_balance)) / (max(cust_rout_balance) - mi

test[, last_week_balance := (last_week_balance - min(last_week_balance)) / (max(last_week_balance) - mi

test[, prod_rout_prop := (prod_rout_prop - min(prod_rout_prop)) / (max(prod_rout_prop) - min(prod_rout_
test[, cust_rout_prop := (cust_rout_prop - min(cust_rout_prop)) / (max(cust_rout_prop) - min(cust_rout_

lqs_model <- readRDS(file = 'model/lqs_model.rds')

test$mass_lqs_predict <- ceiling(predict(object = lqs_model, newdata = test))

test[mass_lqs_predict < 0, mass_lqs_predict := 0]

remove(lqs_model)
gc()
```

```
##             used   (Mb) gc trigger    (Mb)   max used    (Mb)
## Ncells    606305   32.4    6093622   325.5    7617027   406.8
## Vcells 832471535 6351.3 1362235652 10393.1 1339438788 10219.2
```

```
# Do not save on knit doc
# fwrite(x = test[, list(.I, mass_lqs_predict)], file = "dataset/bimbo_submit.csv")

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

```
##             used   (Mb) gc trigger   (Mb)   max used    (Mb)
## Ncells    606506   32.4    4874898  260.4    7617027   406.8
## Vcells 133949214 1022.0 1089788522 8314.5 1339438788 10219.2
```