bimbo_final_train.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 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)
library(MASS)

v_c_file_train <- "dataset/train_features.csv"

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

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

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

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

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

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

x <- quantile(train[, Demanda_uni_equil], probs = 0.9)

train <- train[Demanda_uni_equil <= x]

# The training step was made with 10.000.000 rows
# train <- train[sample(nrow(train), 10000000)]
# To Knit the doc will use only 10.000 rows sample
train <- train[sample(nrow(train), 10000)]

lqs_model <- MASS::lqs(formula = Demanda_uni_equil ~ .,
                       data = train[, list(Demanda_uni_equil,
                                           prod_cust_balance,
                                           prod_rout_balance,
                                           cust_rout_balance,
                                           last_week_balance,
                                           prod_cust_bal_mean,
                                           prod_rout_prop,
                                           cust_rout_prop)])

# Do not save on knit doc
```

```r
# saveRDS(lqs_model, file = 'model/lqs_model_final.rds')
remove(lqs_model)
gc()
```

```
##             used  (Mb) gc trigger   (Mb)   max used    (Mb)
## Ncells    602949  32.3    1226820   65.6     926347    49.5
## Vcells 127432218 972.3 1077858369 8223.5 1336871703 10199.6
```

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

```
##             used  (Mb) gc trigger   (Mb)   max used    (Mb)
## Ncells    603142  32.3    1226820   65.6     926347    49.5
## Vcells 127329353 971.5  862286696 6578.8 1336871703 10199.6
```