bimbo_evaluate.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(caTools)
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(lares)
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff
```

```
library(MASS)
library(randomForest)
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library(Metrics)
```

```
## Warning: package 'Metrics' was built under R version 3.6.2
```

```
##
## Attaching package: 'Metrics'
```

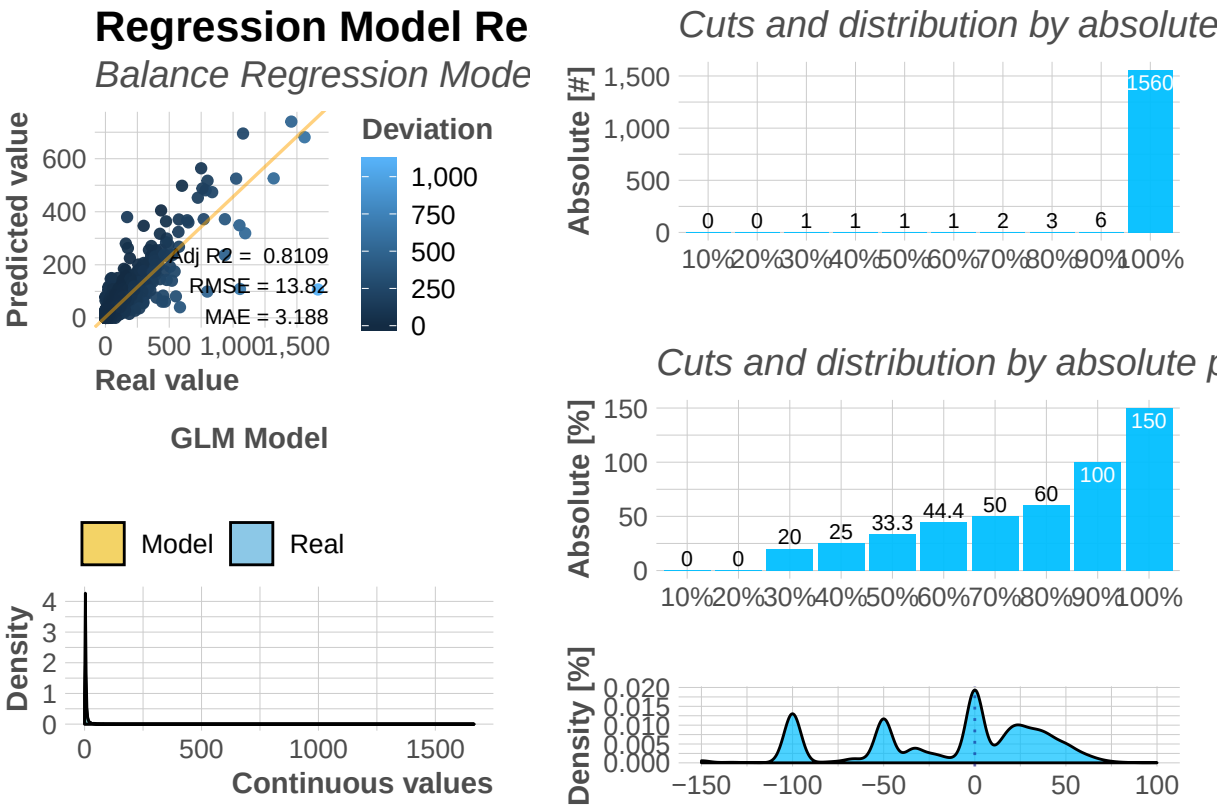
```
## The following objects are masked from 'package:lares':
##
##     mae, mape, mse, rmse
```

```
## The following objects are masked from 'package:caret':
##
##     precision, recall
```

```r
v_c_file_test <- "dataset/test_split.csv"

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

test_sample <- test[sample(nrow(test), 100000)]

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

test_sample$glm_predict <- round(predict(object = glm_model, newdata = test_sample), 0)

test_sample[glm_predict < 0, glm_predict := 0]

#RMSLE
rmsle(test_sample[, Demanda_uni_equil], test_sample[, glm_predict])
```

```
## [1] 0.4345751
```

```r
lares::mplot_full(tag = test_sample[, Demanda_uni_equil],
                  score = test_sample[, glm_predict],
                  splits = 10,
                  subtitle = "Balance Regression Model",
                  model_name = "GLM Model",
                  save = T)
```



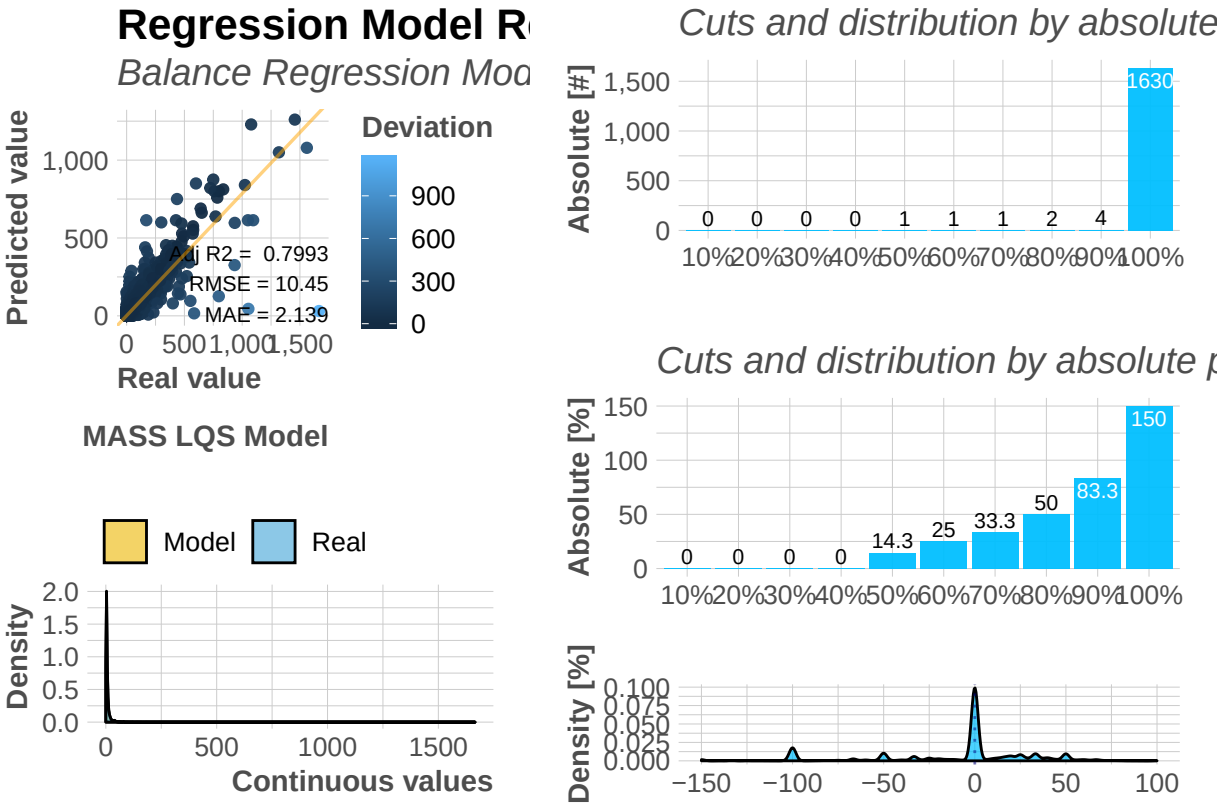

```
remove(glm_model)
gc()
```

```
##             used   (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells   3010931  160.9    5762974  307.8   4100965  219.1
## Vcells 212045568 1617.8  394645614 3011.0 394569688 3010.4
```

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

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

test_sample[mass_lqs_predict < 0, mass_lqs_predict := 0]

#RMSLE
rmsle(test_sample[, Demanda_uni_equil], test_sample[, mass_lqs_predict])
```

```
## [1] 0.3772318
```

```
lares::mplot_full(tag = test_sample[, Demanda_uni_equil],
                  score = test_sample[, mass_lqs_predict],
                  splits = 10,
                  subtitle = "Balance Regression Model",
                  model_name = "MASS LQS Model",
                  save = T)
```



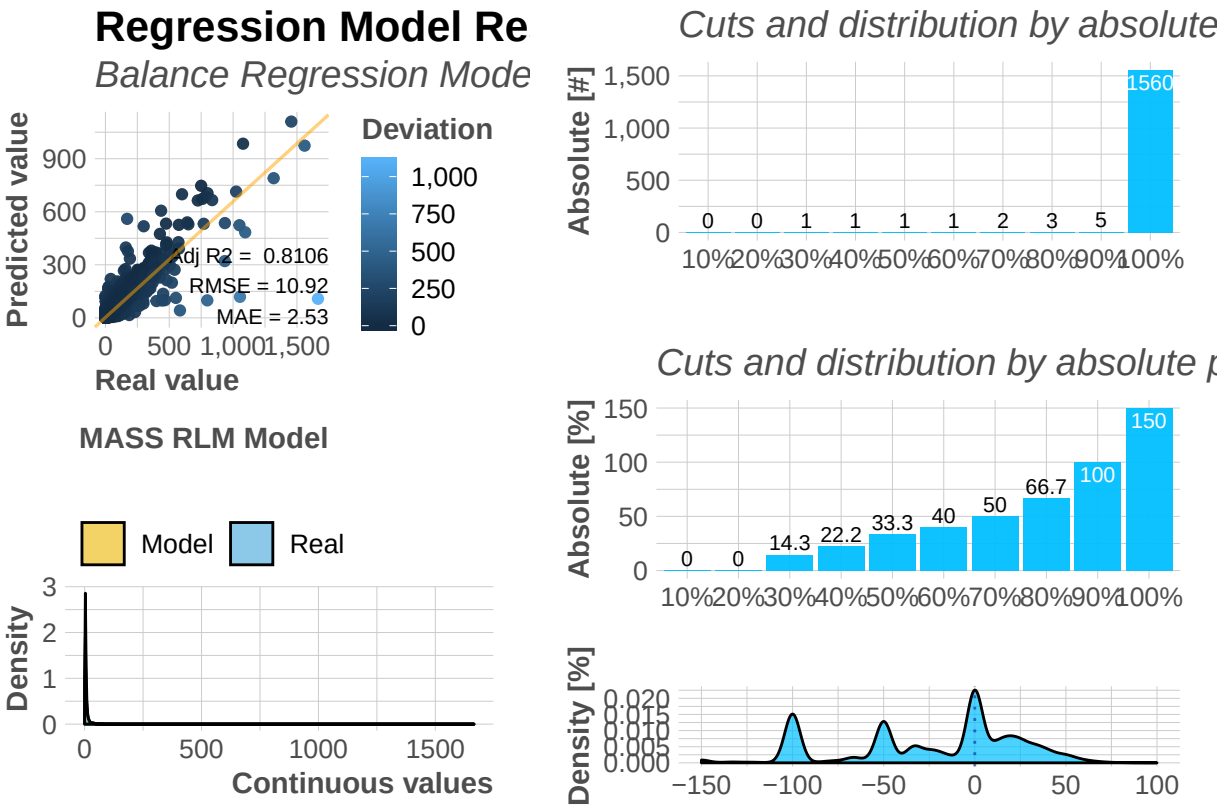

```
remove(lqs_model)
gc()
```

```
##             used   (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells   3152885  168.4    7178258  383.4   7178258  383.4
## Vcells 214858853 1639.3  394645614 3011.0 394638871 3010.9
```

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

test_sample$mass_rlm_predict <- ceiling(predict(object = rlm_model, newdata = test_sample))

test_sample[mass_rlm_predict < 0, mass_rlm_predict := 0]

#RMSLE
rmsle(test_sample[, Demanda_uni_equil], test_sample[, mass_rlm_predict])
```

```
## [1] 0.405933
```

```
lares::mplot_full(tag = test_sample[, Demanda_uni_equil],
                  score = test_sample[, mass_rlm_predict],
                  splits = 10,
                  subtitle = "Balance Regression Model",
                  model_name = "MASS RLM Model",
                  save = T)
```



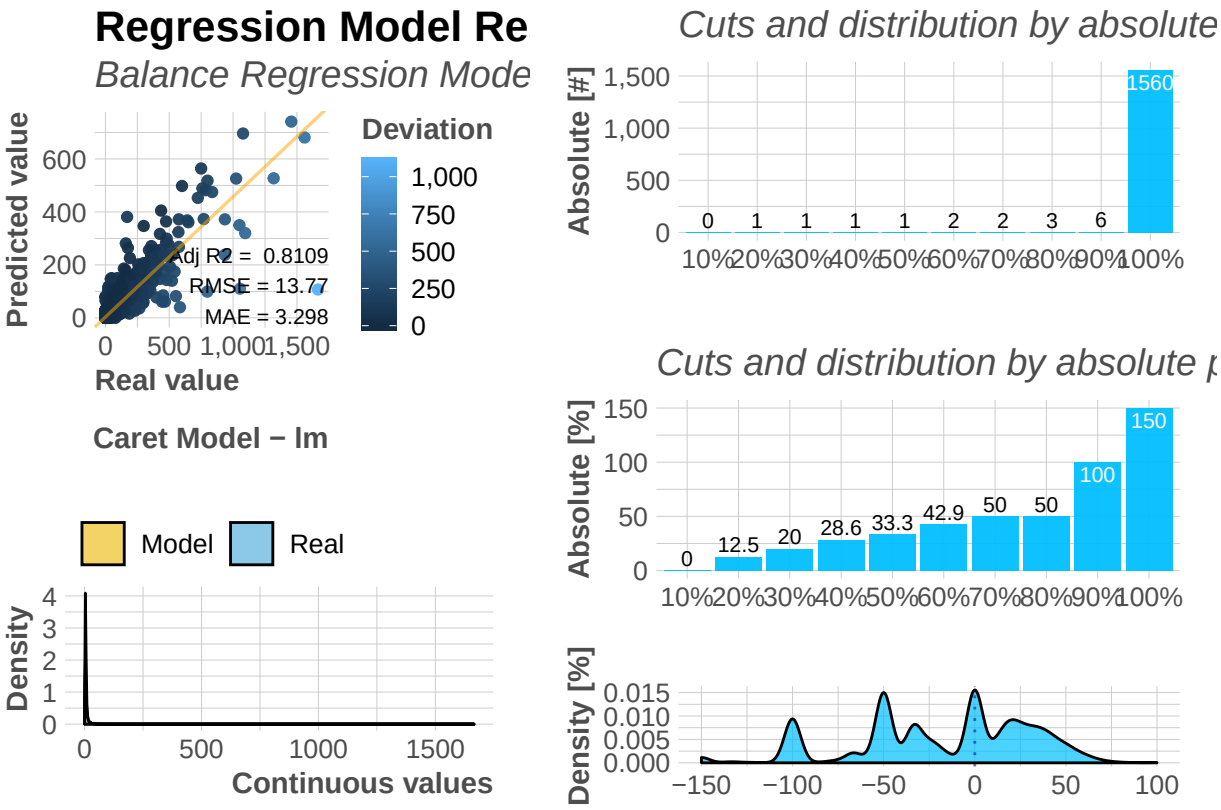

```
remove(rlm_model)
gc()
```

```
##             used   (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells   3259882  174.1    7178258  383.4   7178258  383.4
## Vcells 216478473 1651.6  394645614 3011.0 394638871 3010.9
```

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

test_sample$caret_predict <- ceiling(predict(object = caret_model, newdata = test_sample))

test_sample[caret_predict < 0, caret_predict := 0]

#RMSLE
rmsle(test_sample[, Demanda_uni_equil], test_sample[, caret_predict])
```

```
## [1] 0.4764126
```

```
lares::mplot_full(tag = test_sample[, Demanda_uni_equil],
                  score = test_sample[, caret_predict],
                  splits = 10,
                  subtitle = "Balance Regression Model",
                  model_name = "Caret Model - lm",
                  save = T)
```




```
remove(caret_model)
gc()
```

```
##             used   (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells   3367830  179.9    7178258  383.4   7178258  383.4
## Vcells 218069066 1663.8  685204701 5227.7 685156212 5227.4
```

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

```
##            used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells  3367807 179.9    7178258  383.4   7178258  383.4
## Vcells 31311179 238.9  548163761 4182.2 685156212 5227.4
```