PRÁCTICA 2

Limpieza y validación de los datos

Abstracto

Se realizo un análisis de datos exploratorios, utilizando funciones básicas y avanzadas para el preprocesamiento de datos

Mercy Pinargote
mpinargote@uoc.edu

Contenido

Introducción

El objetivo de esta actividad es realizar el tratamiento de un conjunto de datos, se ha escogido el conjunto de datos: Predict Future Sales (https://www.kaggle.com/c/competitive-data-sciencepredict-future-sales/

Se va a realizar un análisis de datos exploratorios, la utilización de funciones básicas y avanzadas y el preprocesamiento, diversas técnicas de validación de modelos.

1. Descripción del conjunto de datos.

El conjunto de datos contiene información histórica de ventas diarias de la empresa de software 1C que es una de las más grandes de Rusia. La tarea es predecir la cantidad total de productos vendidos en cada tienda para el próximo mes. El conjunto de datos está constituido por 6 columnas y contiene 2.935.849 registros.

Es interesante para las personas que quieran mejorar sus habilidades en ciencia de datos y participar en competencias ya que las competiciones se convierten en una oportunidad única para aprender y competir con otros.

sales_train.csv - el conjunto de entrenamiento. Datos históricos diarios de enero de 2013 a octubre de 2015

Descripción del conjunto de datos

- date - fecha en formato dd/mm/aaaa
- date_block_num - un número de mes consecutivo, utilizado por conveniencia. Enero de 2013 es 0, febrero de 2013 es 1, ..., octubre de 2015 es 33
- shop_id - identificador único de una tienda
- item_id - identificador único de un producto
- item_price - precio actual de un artículo
- item_cnt_day - número de productos vendidos.

La competencia requiere generar un conjunto de datos con la siguiente estructura

ID: un Id que representa una tupla (tienda, Artículo) dentro del conjunto de prueba
item_cnt_month: predicción del número de productos vendidos mensuales

2.Integración y selección de los datos de interés a analizar.

La competencia plantea seleccionar cuales son las variables que pueden ayudarnos a pronosticar las ventas del próximo mes. Además, se podrá proceder a crear modelos de reglas de asociación que permitan pronosticar las ventas del próximo mes por tienda y producto en función del histórico de datos.

3. Limpieza de los datos

Antes de comenzar con la limpieza de los datos, procedemos a realizar la lectura del fichero en formato CSV en el que se encuentran. El resultado devuelto por la llamada a la función read.csv () será un objeto data frame:

```
# Lectura de datos
sales_train_v2 <- read.csv ("C:/MERCY UOC/Tipología y Ciclo de los Datos/Practica/Data/sales_train_v2.csv")
str(sales_train_v2)
```

```
## 'data.frame':    2935849 obs. of  6 variables:
##  $ date          : Factor w/ 1034 levels "01.01.2013","01.01.2014",..: 35
69 137 171 477 307 35 103 341 69 ...
##  $ date_block_num: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ shop_id       : int  59 25 25 25 25 25 25 25 25 25 ...
##  $ item_id       : int  22154 2552 2552 2554 2555 2564 2565 2572 2572 2573
...
##  $ item_price    : num  999 899 899 1709 1099 ...
##  $ item_cnt_day  : num  1 1 -1 1 1 1 1 1 1 3 ...
```

De estas variables nos interesa utilizar: date_block_num shop_id item_id item_cnt_day Las otras variables no aportan para el estudio que se va a realizar

Se va a realizar estadística descriptiva utilizando la función summary () que nos muestra los valores de la media, mediana, 25 y 75 cuartiles, mín. y máx de todas las variables numéricas en el conjunto de datos.

```
summary(sales_train_v2$item_cnt_day)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  -22.000    1.000    1.000    1.243    1.000 2169.000
```

Se identifica que existen valores negativos en la variable item_cnt_day estos valores vamos a utilizarlos para el estudio, en este caso los valores negativos se van a considerar como devoluciones de productos por lo que no se va a realizar modificación a estos valores ni descartar.

Adicional se observa que la media tiene un valor superior a la mediana. Por lo que se podría decir que existen valores extremos. Para comprobar esto más adelante se va a realizar el estudio de valores extremos.

3.1. Ceros y elementos vacíos

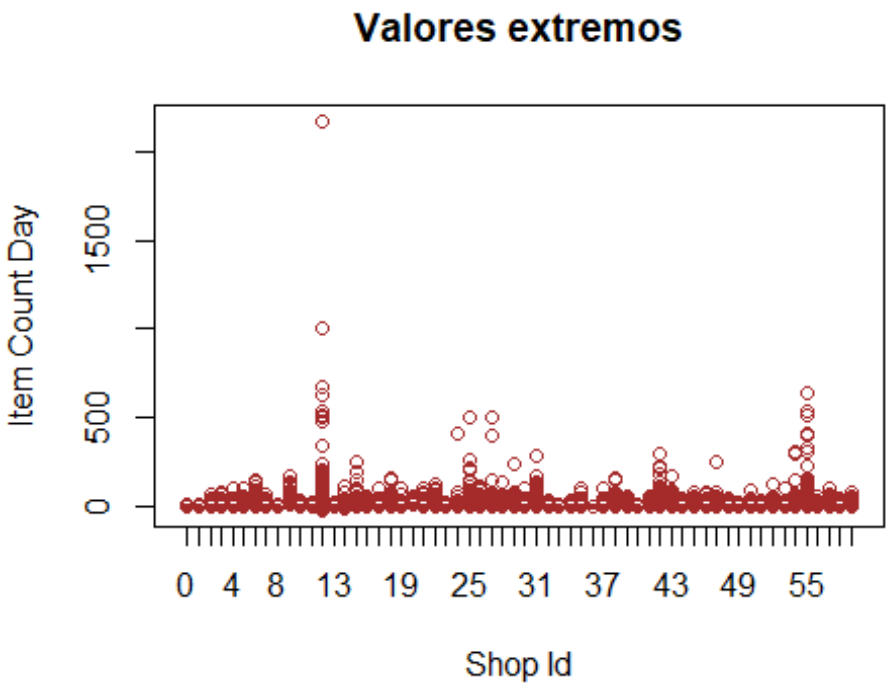
En R, los valores faltantes están representados por el símbolo NA (no disponible). Para verificar si existen elementos vacíos se va a utilizar la función is.na ().

```
# Números de valores desconocidos por campo
sapply (sales_train_v2, function(x) sum(is.na(x)))

##           date date_block_num        shop_id        item_id     item_price
##              0              0              0              0              0
##   item_cnt_day
##              0
```

El resultado muestra que no existen campos con valores vacíos. ### 3.2. Valores extremos

Un valor más extremo (outlier) es un valor en un conjunto de datos que es muy diferente de los otros valores. Para identificar los valores extremos se va a realizar un gráfico de cajas con la función boxplot.

```
boxplot (sales_train_v2$item_cnt_day ~ sales_train_v2$shop_id, main="Valores
extremos",
          xlab="Shop Id",
          ylab="Item Count Day",
          col="orange",
          border="brown")
```



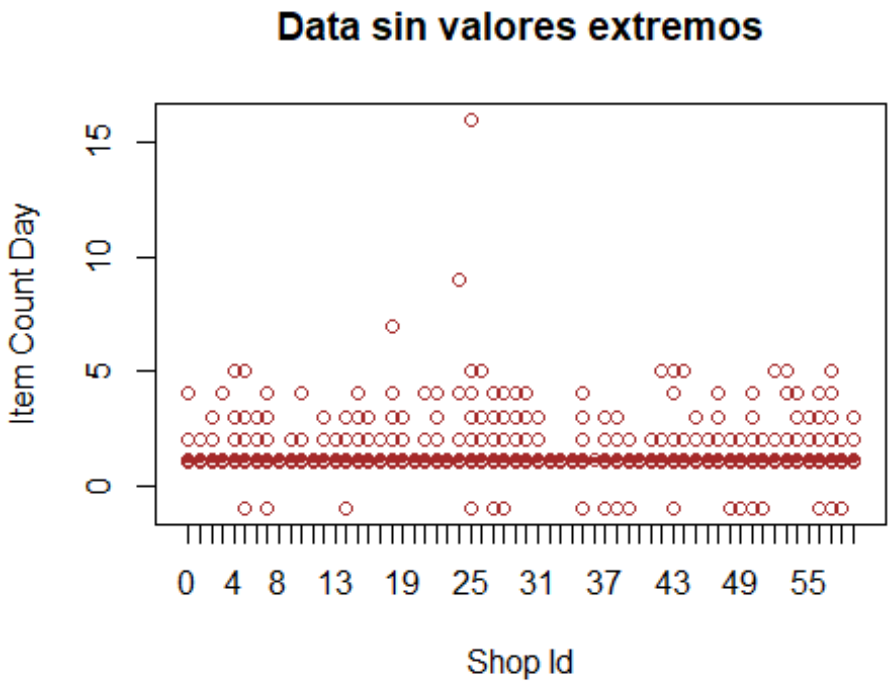
Mediante el grafico se puede observar la presencia de valores extremos. Se va a modificar los valores extremos para cada tienda.

```
ids <-unique(sales_train_v2$shop_id)
for (i in ids) {
  outlier_values <- boxplot.stats(sales_train_v2[which(sales_train_v2$shop_id
==i & sales_train_v2$item_cnt_day>-1 ),]$item_cnt_day)$out  # outliers
```

```
  # Reemplazar outlider con NA
  sales_train_v2[which(sales_train_v2$shop_id==i),]$item_cnt_day <- ifelse(sa
les_train_v2[which(sales_train_v2$shop_id==i),]$item_cnt_day %in% outlier_val
ues, NA, sales_train_v2$item_cnt_day)
  # Imputar valores NA con la media
  sales_train_v2$item_cnt_day[is.na(sales_train_v2[which(sales_train_v2$shop_
id==i),]$item_cnt_day)] <-mean(sales_train_v2[which(sales_train_v2$shop_id==i
),] $item_cnt_day,na.rm=T)
}
```

Grafica sin valores extremos

```
boxplot(item_cnt_day ~ shop_id, main="Data sin valores extremos", data=sales_
train_v2,
          xlab="Shop Id",
          ylab="Item Count Day",
          col="orange",
          border="brown")
```



Finalmente se van a agregar los datos para continuar con el análisis de los datos

3.3. Agregación de datos

```
library(sqldf)
```

```
sales_train_v2_sum <- sqldf ('SELECT date_block_num, shop_id, item_id, SUM(it
em_cnt_day) AS item_cnt_month FROM sales_train_v2 GROUP BY date_block_num, sh
op_id, item_id')
```

```
tail(sales_train_v2_sum)
```

```
##         date_block_num shop_id item_id item_cnt_month
## 1609119             33      59   21812       1.134464
## 1609120             33      59   22087       3.403271
## 1609121             33      59   22088       2.268993
## 1609122             33      59   22091       1.134700
## 1609123             33      59   22100       1.134496
## 1609124             33      59   22102       1.134428
```

3.4. Cambio de datos tipo número a factor

Se utiliza la función str () para verificar el tipo de datos de las variables del conjunto de datos que se va a utilizar para el análisis de datos

```
str(sales_train_v2_sum)
```

```
## 'data.frame':    1609124 obs. of  4 variables:
##  $ date_block_num: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ shop_id       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ item_id       : int  32 33 35 43 51 61 75 88 95 96 ...
##  $ item_cnt_month: num  4.54 3.4 1.13 1.13 2.27 ...
```

Se puede observar que la variable date_block_num es de tipo número y lo vamos a convertir a tipo factor

```
sales_train_v2_sum$date_block_num <- factor (sales_train_v2_sum$date_block_nu
m)
```

4. Análisis de los datos

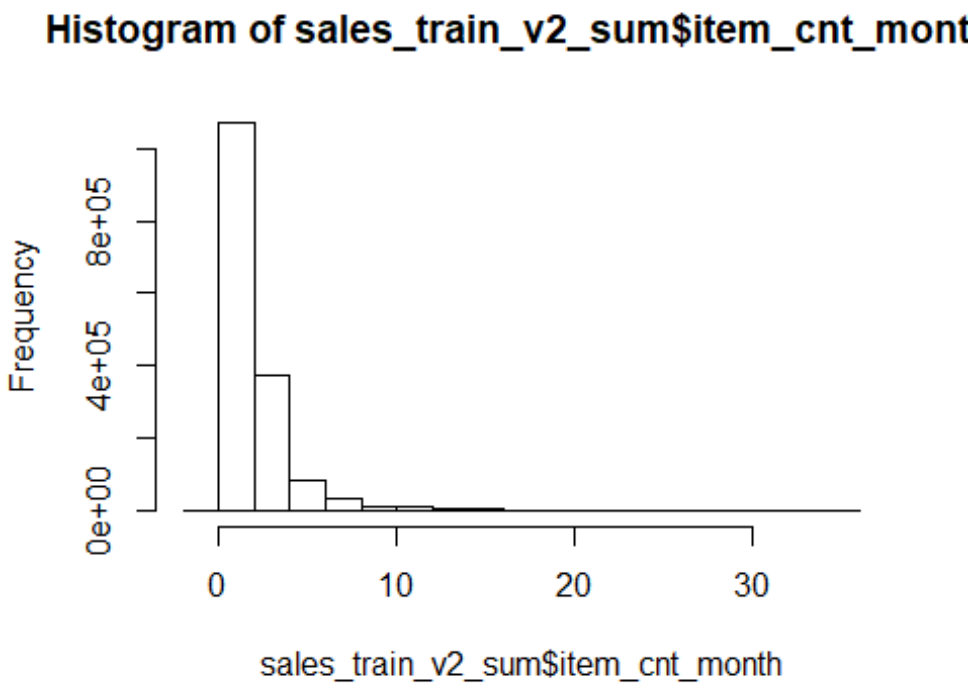
4.1. Selección de los grupos de datos que se quieren analizar/comparar

Se van a utilizar las variables: shop_id, item_id y item_cnt_month

4.2. Comprobación de la normalidad y homogeneidad de la varianza

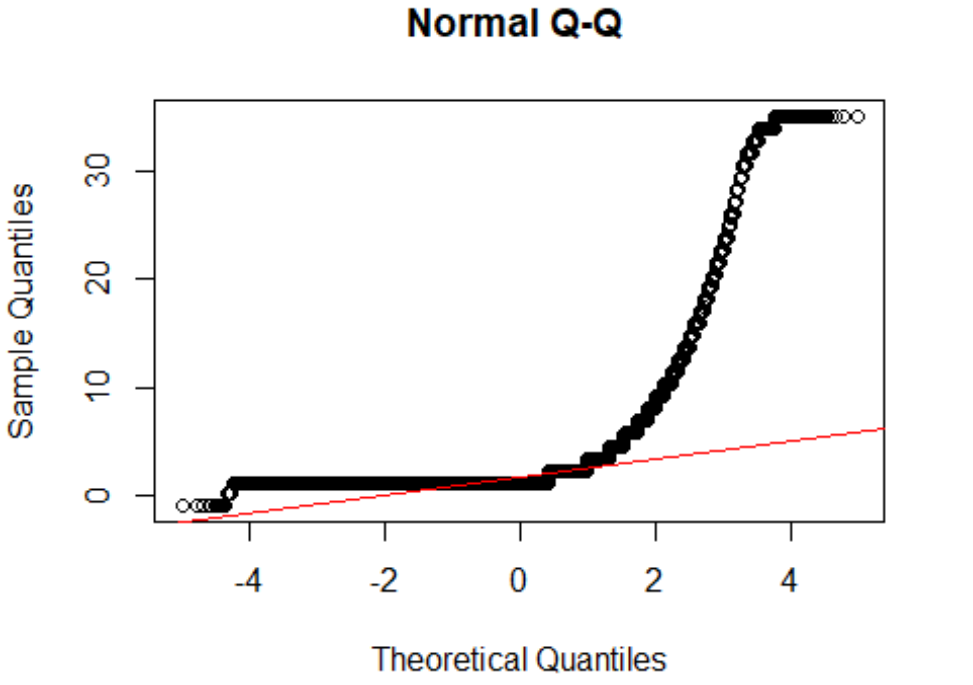
Para la comprobación de la normalidad se va a utilizar la prueba de Anderson Darling para este caso porque es un conjunto grande de datos.

```
library(nortest)
ad.test(sales_train_v2_sum$item_cnt_month) $p.value

## [1] 3.7e-24

hist(sales_train_v2_sum$item_cnt_month)
```



El resultado de la prueba y el histograma indican que los datos no siguen una distribución normal ya que valor p es inferior al coeficiente 0.05. Y el grafico del histograma muestra que no es una distribución normal.

```
qqnorm (sales_train_v2_sum$item_cnt_month, main= "Normal Q-Q ")
qqline (sales_train_v2_sum$item_cnt_month, col="red”)
```



Para estudiar la homogeneidad de varianzas se va a utilizar la prueba de Fligner-Killeen porque en este caso los datos se desvían de la normal. En este caso, estudiaremos esta homogeneidad en cuanto a los grupos conformados por las tiendas de la cadena.

```
fligner.test (item_cnt_month ~ shop_id, data = sales_train_v2_sum)
```

```
##
## Fligner-Killeen test of homogeneity of variances
##
## data:  item_cnt_month by shop_id
## Fligner-Killeen: med chi-squared = 65876, df = 59, p-value <
## 2.2e-16
```

Como el p-valor es menor a 0,05, rechazamos la hipótesis de que las varianzas de las muestras son homogéneas.

4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos.

En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc.

4.3.1 Contraste entre meses

Se puede contrastar si las ventas de la cadena han aumentado durante los últimos dos meses. Se puede verificar si se puede afirmar que con un nivel de confianza del 90% las ventas han aumentado los últimos dos meses. Para hacer esto se puede utilizar un contraste de hipótesis de una muestra de datos apareados, tal como se describe en Rovira (2009) (p.21). Se trata de un contraste unilateral. Hipótesis nula y alternativa

H0:μ32=μ33 H1:μ33>μ32

o de forma equivalente:

H0:dif=0 H1:dif>0 donde "dif" es la muestra de las diferencias entre los meses 32 y 33

```
sales_train_v2_test <- subset (sales_train_v2_sum, date_block_num == 33 | dat
e_block_num ==32)
var.test (item_cnt_month ~ date_block_num, data = sales_train_v2_test)
```

```
##
## F test to compare two variances
##
## data:  item_cnt_month by date_block_num
## F = 1.116, num df = 29454, denom df = 31287, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 1.091155 1.141388
```

```
## sample estimates:
## ratio of variances
##           1.115984
```

El valor p del contraste unilateral es 2.2e-16. Es un valor inferior al 0.10 establecido con un 90% de nivel de confianza. Por tanto, se puede rechazar la hipótesis nula de que las ventas entre los meses 32 y 33 son las mismas.

4.3.2 Contrastes entre categoría de productos

Se puede contrastar si los productos que pertenecen a las categorías consolas ("15 Xbox 360" y "16 Xbox one") tienen un nivel de ventas superior al resto de categorías de la muestra, con un nivel de confianza del 97%.

Lo primero que se va a realizar es la preparación de datos, se va a cargar el conjunto de datos de categorías y se va a hacer un merge con los datos de ventas

```
# Lectura de datos
item_categories <- read.csv ("C:/MERCY UOC/Tipología y Ciclo de los Datos/Practica/Data/category.txt")
# Merge de datos
sales_train_v2_sum <- merge (sales_train_v2_sum, item_categories)
```

Ahora se van a crear dos data frames que contengan por separado las ventas de productos que pertenecen a las categorías de consolas mencionadas y, por otra parte, el resto de productos de la muestra, con un nivel de confianza del 97%.

```
Consolas <- sales_train_v2_sum [which (sales_train_v2_sum$category==15 | sales_train_v2_sum$category==16),]
noConsolas <- sales_train_v2_sum [which (sales_train_v2_sum$category! =15 & sales_train_v2_sum$category! =16),]
```

Hipótesis nula H0:μConsolas=μnonConsolas H1:μConsolas>μnonConsolas

Se va aplicar un contraste de dos muestras sobre la diferencia de medias. Se aplica el caso de muestras grandes no normales, según Gibergans Baguena (2009) (p.9). Es un contraste unilateral.

```
t.test(x = Consolas$item_cnt_month,
       y = noConsolas$item_cnt_month,
       alternative = "two.sided", mu = 0, var.equal = TRUE, conf.level = 0.97
)
```

```
##
## Two Sample t-test
##
## data:  Consolas$item_cnt_month and noConsolas$item_cnt_month
```

```
## t = -2.2345, df = 1600200, p-value = 0.02545
## alternative hypothesis: true difference in means is not equal to 0
## 97 percent confidence interval:
## -0.150464509 -0.002200959
## sample estimates:
## mean of x mean of y
## 1.988592 2.064925

rm(Consolas)
rm(noConsolas)
```

Como el valor p=0.02545 es inferior a α=0.03, notablemente inferior a α=0.03, podemos rechazar la hipótesis nula de que las ventas mensuales entre estas dos categorías son iguales a favor de la hipótesis alternativa.

4.3.3 Modelo de regresión lineal

El problema plantea que se debe pronosticar las ventas por tienda y producto, por lo que primero se va a crear una tupla utilizando las variables shop_id y item_id.

```
#2. Asignar ID unico para tienda y producto
sales_train_v2_sum <- sales_train_v2_sum[order(sales_train_v2_sum$shop_id, sa
les_train_v2_sum$item_id),]
sales_train_v2_sum$ID <- cumsum(!duplicated(sales_train_v2_sum[3:4]))
#sales_train_v2_clean <- sales_train_v2_sum[!duplicated(sales_train_v2_sum$ID
), ]

summary(sales_train_v2_sum)
```

```
##     item_id      date_block_num       shop_id      item_cnt_month
## Min.:    0   11:  66276   Min.: 0.00   Min.   :-1.000
##  1st Qu.: 5045   2      :  63977   1st Qu.:21.00   1st Qu.: 1.134
##  Median :10497   0      :  63224   Median :31.00   Median : 1.134
##  Mean   :10681   1      :  59935   Mean   :32.81   Mean   : 2.065
##  3rd Qu.:16060   23     :  59275   3rd Qu.:47.00   3rd Qu.: 2.269
##  Max.   :22169   6      :  58035   Max.   :59.00   Max.   :35.181
##                  (Other):1238402                   NA's   :8886
##     category           ID
##  Min.   : 0.00   Min.   :     1
##  1st Qu.:30.00   1st Qu.: 58487
##  Median :40.00   Median :127395
##  Mean   :41.54   Mean   :122408
##  3rd Qu.:55.00   3rd Qu.:184040
##  Max.   :83.00   Max.   :242256
##
```

Se va aplicar un modelo de regresión lineal para calcular las ventas futuras. Primero se va a evaluar

Primero se va a crear dos conjuntos de datos uno para entrenamiento y otro para pruebas

```
rowstrain <- nrow(sales_train_v2_sum)*0.8
set.seed(100000)
index <-sample(1:nrow(sales_train_v2_sum),size=rowstrain)

train <- sales_train_v2_sum[index,]
test <- sales_train_v2_sum[-index,]

modelo1 <- lm(item_cnt_month ~  ID + category, data = train)
modelo2 <- lm(item_cnt_month ~  ID, data = train)

tabla.coeficientes <- matrix(c(1, summary(modelo1)$r.squared,
2, summary(modelo2)$r.squared),
ncol = 2, byrow = TRUE)
colnames(tabla.coeficientes) <- c("Modelo", "R^2")
tabla.coeficientes

##      Modelo          R^2
## [1,]      1 0.0077870236
## [2,]      2 0.0001014877
```

Se va a utilizar el segundo modelo porque tiene un mayor coeficiente de determinación.

```
predict_mo <- predict(modelo2, test, type="response")
mc_sl<-data.frame(real=test$item_cnt_month, predicted=predict_mo, dif=ifelse(
test$item_cnt_month>predict_mo, -predict_mo*100/test$item_cnt_month, predict_
mo*100/test$predict_mo))
colnames(mc_sl)<- c("Real", "Predecido", "Dif%")
tail(mc_sl)
```

```
##             Real Predecido      Dif%
## 1607575 1.134507  2.102171        NA
## 1607675 3.403297  2.102171 -61.76865
## 1607923 1.134496  2.102171        NA
## 1608299 1.134428  2.102171        NA
## 1608562 1.133946  2.102171        NA
## 1608586 2.268677  2.102171 -92.66068
```

Finalmente se va a utilizar el conjunto de datos de Kaggle para realizar la predicción solicitada en la competencia y generar el conjunto de datos para Kaggle

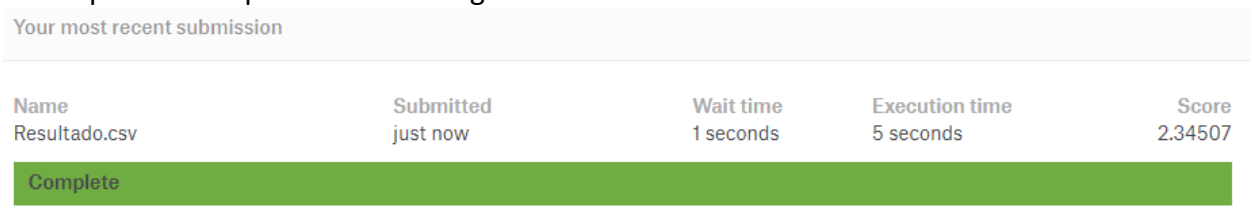
```
library(dplyr)

train <- sales_train_v2_sum[,c("shop_id","item_id","item_cnt_month")]
test <- read.csv("C:/MERCY UOC/Tipologia y Ciclo de los Datos/Practica/Data/t
est.csv")
train <-inner_join(train,test)
```

```
## Joining, by = c("shop_id", "item_id")

modelo <- lm(item_cnt_month ~  ID , data = train)
predict_mo <- predict(modelo, test, type="response")

resultado <-data.frame(ID=test$ID, item_cnt_month=predict_mo)
resultado <- unique(resultado)
write.csv(resultado, file = "C:/MERCY UOC/Tipologia y Ciclo de los Datos/Practica/Data/Resultado.csv",row.names=FALSE)
```

Conclusiones

La limpieza de datos es un proceso largo pero necesario para conseguir buenos modelos estadísticos. Se han aplicado varias técnicas para encontrar valores NA, valores extremos y se realizó imputación de valores utilizando la media para tener una mejor muestra para nuestro estudio.

El conjunto de datos seleccionado tiene 5 variables y solo se utilizaron 3 para la generación del archivo final para realizar el archivo que se subió a la competencia de kaggle. Los resultados en la competencia se pueden ver en el grafico a continuación:



Es muy interesante el utilizar el conjunto de datos de una competencia porque me ha permitido mejorar el modelo y el código para mejora la puntuación.

Referencias

- Squire, Megan (2015). Clean Data. Packt Publishing Ltd.
- Jiawei Han, Micheine Kamber, Jian Pei (2012). Data mining: concepts and techniques. Morgan Kaufmann.
- Gibergans Baguena (2009). Regresión lineal múltiple.