09.1.- Data Cleansing-Basic_04_19_vacunacion_completo_v_01

June 8, 2023

#

CU04_Optimización de vacunas

Citizenlab Data Science Methodology > II - Data Processing Domain *** > # 09.1.- Data Cleansing - Basic

Data Cleaning refers to identifying and correcting (or removing) errors in the dataset that may negatively impact a predictive model, replacing, modifying, or deleting the dirty or coarse data.

0.1 Tasks

Basic operations	Text data analysis
	Delete Needless/Irrelevant/Private Columns
	Inconsistent Data.
	Expected values
	Zeroes
	Columns with a Single Value
	Columns with Very Few Values
	Columns with Low Variance
	Duplicates (rows/samples) & (columns/features) Data
Missing Values	Missing Values Identification
	Missing Values Per Sample
	Missing Values Per Feature
	Zero Missing Values
	Other Missing Values
	Null/NaN Missing Values
	Delete Missing Values
	Deleting Rows with Missing Values in Target Column
	Deleting Rows with Missing Values
	Deleting Features with some Missing Values
	Deleting Features using Rate Missing Values
	Basic Imputation
	Imputation by Previous Row Value
	Imputation by Next Row Value
	Statistical Imputation
	Selection of Imputation Strategy
	Constant Imputation
	Mean Imputation
	Median Imputation
	Most Frequent Imputation
	Interpolation Imputation
	Prediction Imputation (KNN Imputation)
	Evaluating k-hyperparmeter in KNN Imputation
	Applying KNN Imputation
	Iterative Imputation
	Evaluating Different Imputation Order
	Applying Iterative Imputation
Outliers	Outliers - Univariate
	Visualizing Outliers
	Distribution
	Box Plots
	Isolation Forest
	Outliers Identification
	Grubbs' Test
	Z-Score
	Standard Deviation Method
	Interquartile Range Method
	Tukey's method
	Internally studentized residuals AKA z-score method
	Median Absolute Deviation method
	Outliers - MultiVariate
	Visualizing Outliers
	ScatterPlots
	Outliers Identification
	Mahalanobis Distance
	Robust Mahalanobis Distance
	DBSCAN Clustering
	PyOD Library
	Automatic Detection and Removal of Outliers
	Compare Algorithms
	LocalOutlierFactor
	IsolationForest
	Minimum Covariance Determinant

0.2 Consideraciones casos CitizenLab programados en R

- La mayoría de las tareas de este proceso se han realizado en los notebooks del proceso 05 Data Collection porque eran necesarias para las tareas ETL. En esos casos, en este notebook se referencia al notebook del proceso 05 correspondiente
- Por tanto en los notebooks de este proceso de manera general se incluyen las comprobaciones necesarias, y comentarios si procede
- Las tareas del proceso se van a aplicar solo a los archivos que forman parte del despliegue, ya que hay muchos archivos intermedios que no procede pasar por este proceso
- El nombre de archivo del notebook hace referencia al nombre de archivo del proceso 05 al que se aplica este proceso, por eso pueden no ser correlativa la numeración
- Las comprobaciones se van a realizar teniendo en cuenta que el lenguaje utilizado en el despliegue de este caso es R

0.3 File

- Input File: CU_04_08_20_vacunacion_gripe_train_and_test.csv
- Output File: CU_04_09.1_20_vacunacion_gripe_train_and_test.csv

0.4 Settings

0.4.1 Encoding

Con la siguiente expresión se evitan problemas con el encoding al ejecutar el notebook. Es posible que deba ser eliminada o adaptada a la máquina en la que se ejecute el código.

[3]:
```
Sys.setlocale(category = "LC_ALL", locale = "es_ES.UTF-8")
```

```
Warning message in Sys.setlocale(category = "LC_ALL", locale = "es_ES.UTF-8"):
"OS reports request to set locale to "es_ES.UTF-8" cannot be honored"
```

”

0.4.2 Libraries to use

[4]:
```
library(readr)
library(dplyr)
library(tidyr)
library(stringr)
```

0.4.3 Paths

[5]:
```
iPath <- "Data/Input/"
oPath <- "Data/Output/"
```

0.5 Data Load

OPCION A: Seleccionar fichero en ventana para mayor comodidad

Data load using the {tcltk} package. Ucomment the line if using this option

[6]:
```
# file_data <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

[60]:
```
iFile <- "CU_04_08_20_vacunacion_gripe_train_and_test.csv"
file_data <- paste0(iPath, iFile)

if(file.exists(file_data)){
    cat("Se leerán datos del archivo: ", file_data)
} else{
    warning("Cuidado: el archivo no existe.")
}
```

```
Se leerán datos del archivo:
Data/Input/CU_04_08_20_vacunacion_gripe_train_and_test.csv
```

Data file to dataframe Usar la función adecuada según el formato de entrada (xlsx, csv, json, ...)

[61]:
```
data <- read_csv(file_data)
```

```
Rows: 21736 Columns: 48
  Column specification


Delimiter: ","
chr  (3): GEOCODIGO, DESBDT, nombre_zona
dbl (44): ano, semana, n_vacunas, n_citas, tmed, prec, velmedia,
presMax, be…
lgl  (1): is_train

 Use `spec()` to retrieve the full column specification for this
data.
 Specify the column types or set `show_col_types = FALSE` to quiet
this message.
```

Visualizo los datos.

Estructura de los datos:

[62]:
```
data |> glimpse()
```

```
Rows: 21,736
Columns: 48
$ GEOCODIGO         <chr> "097", "128", "155", "085", "049",
"254", "264", "27…
$ DESBDT            <chr> "Galapagar", "La Ribota",
"Majadahonda", "Ensanche V…
$ ano               <dbl> 2022, 2021, 2022, 2021, 2022, 2022,
2022, 2023, 2022…
```

```
$ semana            <dbl> 33, 47, 39, 46, 24, 5, 38, 1, 26,
2, 47, 18, 23, 5, …
$ n_vacunas         <dbl> 0, 451, 0, 813, 0, 250, 0, 144, 0,
282, 166, 0, 0, 1…
$ n_citas           <dbl> 0, 437, 0, 789, 0, 235, 0, 137, 0,
271, 159, 0, 0, 1…
$ tmed              <dbl> 21.768536, 6.039860, 15.436997,
9.887983, 21.108264,…
$ prec              <dbl> 0.0550769418, 1.2404689012,
0.6913641020, 0.07183897…
$ velmedia          <dbl> 2.4482484, 2.7974515, 2.7535661,
2.5478336, 3.956291…
$ presMax           <dbl> 901.1438, 936.6692, 926.6612,
952.3018, 833.8937, 89…
$ benzene           <dbl> 0.1795784, 0.3697754, 0.2254214,
0.4194085, 0.195865…
$ co                <dbl> 0.4692918, 0.3468722, 0.4797698,
0.2673996, 0.331213…
$ no                <dbl> 2.005147, 9.513899, 6.130449,
10.993518, 2.451963, 7…
$ no2               <dbl> 10.213564, 24.689603, 22.593902,
36.187953, 10.93601…
$ nox               <dbl> 13.02255, 38.42422, 31.55546,
53.19129, 13.60685, 25…
$ o3                <dbl> 88.27507, 36.57543, 58.67398,
32.54918, 77.88477, 55…
$ pm10              <dbl> 13.887308, 9.361394, 10.401526,
12.783278, 44.451891…
$ pm2.5             <dbl> 8.707578, 6.051115, 5.266344,
6.459633, 17.136398, 1…
$ so2               <dbl> 2.086115, 1.552412, 2.758390,
2.444614, 2.854909, 3.…
$ campana           <dbl> NA, 2021, 2022, 2021, NA, 2021,
2022, 2022, NA, 2021…
$ scampana          <dbl> NA, 12, 4, 11, NA, 22, 3, 18, NA,
19, 12, NA, NA, 22…
$ capacidad_zona    <dbl> 11051, 8524, 12733, 15717, 3792,
6640, 10796, 3364, …
$ prop_riesgo       <dbl> 0.14603798, 0.16062611, 0.21143809,
0.06622598, 0.20…
$ tasa_riesgo       <dbl> 0.003617039, 0.009632178,
0.005353189, 0.012969731, …
$ tasa_mayores      <dbl> 0.018360890, 0.034418204,
0.018018046, 0.026783402, …
$ poblacion_mayores <dbl> 0.13306650, 0.14633197, 0.19219091,
0.06053132, 0.18…
$ nombre_zona       <chr> "Galapagar", "La Ribota",
"Majadahonda", "Ensanche V…
```

```
$ nsec              <dbl> 17, 19, 34, 28, 6, 12, 22, 11, 20,
21, 10, 12, 15, 1…
$ t3_1              <dbl> 40.03807, 39.60720, 42.19556,
34.34724, 43.62860, 41…
$ t1_1              <dbl> 44067, 34068, 51144, 62530, 15146,
26552, 43267, 134…
$ t2_1              <dbl> 0.5121733, 0.5109523, 0.5298013,
0.5077573, 0.501588…
$ t2_2              <dbl> 0.4878267, 0.4890477, 0.4701987,
0.4922427, 0.498411…
$ t4_1              <dbl> 0.17622140, 0.19623219, 0.16029496,
0.23756034, 0.14…
$ t4_2              <dbl> 0.6906908, 0.6574383, 0.6475255,
0.7018912, 0.676094…
$ t4_3              <dbl> 0.13306650, 0.14633197, 0.19219091,
0.06053132, 0.18…
$ t5_1              <dbl> 0.15387677, 0.07211496, 0.12445661,
0.12744893, 0.12…
$ t6_1              <dbl> 0.22398769, 0.11679614, 0.21183967,
0.19323644, 0.15…
$ t7_1              <dbl> 0.07342751, 0.05250060, 0.07595339,
0.04601377, 0.05…
$ t8_1              <dbl> 0.05728152, 0.03935768, 0.06703038,
0.03454148, 0.04…
$ t9_1              <dbl> 0.4408272, 0.4406703, 0.5570257,
0.4603761, 0.387025…
$ t10_1             <dbl> 0.12371972, 0.11272335, 0.08802468,
0.13945576, 0.11…
$ t11_1             <dbl> 0.5291455, 0.6094153, 0.5018791,
0.6560315, 0.515400…
$ t12_1             <dbl> 0.6040733, 0.6814646, 0.5505073,
0.7524379, 0.585228…
$ area              <dbl> 96647460.4, 1364369.5, 30837796.0,
48678625.6, 87516…
$ densidad_hab_km   <dbl> 455.95611, 24969.77491, 1658.48428,
1284.54736, 173.…
$ tuits_gripe       <dbl> 34, 280, 126, 206, 46, 144, 98, 24,
70, 508, 280, 12…
$ interes_gripe     <dbl> 11, 64, 42, 64, 21, 20, 32, 64, 20,
69, 64, 36, 26, …
$ is_train          <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE,
TRUE, TRUE, TRUE…
```

Muestra de los primeros datos:

[63]:
```
data |> slice_head(n = 5)
```

	GEOCODIGO <chr>	DESBDT <chr>	ano <dbl>	semana <dbl>	n_vacunas <dbl>	n_citas <dbl>	tmed <dbl>
	097	Galapagar	2022	33	0	0	21.76853
A spec_tbl_df: 5 × 48	128	La Ribota	2021	47	451	437	6.039860
	155	Majadahonda	2022	39	0	0	15.43699
	085	Ensanche Vallecas	2021	46	813	789	9.887983
	049	Cercedilla	2022	24	0	0	21.10826

0.6 Text data analysis

Select columns

[64]:
```
# Select column
text_columns <- sapply(data, is.character)
```

Operation

[65]:
```
# Analizar datos de texto y verificar su corrección
# e.g. faltas ortografía, etc
```

[66]:
```
# pasar a mayúsculas todas las columnas de texto
data[, text_columns] <- lapply(data[, text_columns], function(x) toupper(x))
```

0.7 Delete Columns Needless/Irrelevant/Private

Select columns

[67]:
```
# Select columns
```

Operation

[68]:
```
# Eliminamos columnas que consideramos irrelevantes o innecesarias
```

Todas las columnas son relevantes, por lo que no aplica.

0.8 Inconsistent Data

[ ]:
```

```

Select columns and value

[69]:
```
# Select column and value
# e.g. age > 100
numeric_columns <- sapply(data, is.numeric)
```

Operation

[70]:
```
# Inconsistent data is unique to each data set and
# must be searched manually
data[, numeric_columns] <- lapply(data[, numeric_columns], function(x) {
```

```
  ifelse(is.na(x), ifelse(is.integer(x), as.integer(mean(x, na.rm = TRUE)),␣
 ↪mean(x, na.rm = TRUE)), x)
})

data <- data[!(data$ano < 2020 | data$ano > 2024), ]
data <- data[!(data$semana < 0 | data$semana > 53), ]
```

0.9 Expected values

[71]:
```
# Check for expected value
nan_counts <- colSums(is.na(data))
nan_counts
```

GEOCODIGO 0 **DESBDT** 868 **ano** 0 **semana** 0 **n_vacunas** 0 **n_citas** 0 **tmed** 0 **prec** 0 **velmedia** 0 **presMax** 0 **benzene** 0 **co** 0 **no** 0 **no2** 0 **nox** 0 **o3** 0 **pm10** 0 **pm2.5** 0 **so2** 0 **campana** 0 **scampana** 0 **capacidad_zona** 0 **prop_riesgo** 0 **tasa_riesgo** 0 **tasa_mayores** 0 **poblacion_mayores** 0 **nombre_zona** 564 **nsec** 0 **t3_1** 0 **t1_1** 0 **t2_1** 0 **t2_2** 0 **t4_1** 0 **t4_2** 0 **t4_3** 0 **t5_1** 0 **t6_1** 0 **t7_1** 0 **t8_1** 0 **t9_1** 0 **t10_1** 0 **t11_1** 0 **t12_1** 0 **area** 0 **densidad_hab_km** 0 **tuits_gripe** 0 **interes_gripe** 0 **is_train** 0

0.10 Zeros

No aplica. ETL satisface los requisitos de calidad de los datos para valores cero.

0.11 Single Value

[72]:
```
# We obtain the number of different values of each column
distinct_counts <- sapply(data, function(x) n_distinct(x, na.rm = TRUE))
distinct_counts
```

GEOCODIGO 286 **DESBDT** 282 **ano** 3 **semana** 52 **n_vacunas** 699 **n_citas** 671 **tmed** 21736 **prec** 19422 **velmedia** 20888 **presMax** 21451 **benzene** 17215 **co** 18303 **no** 17020 **no2** 20023 **nox** 19397 **o3** 20831 **pm10** 16586 **pm2.5** 19704 **so2** 9453 **campana** 4 **scampana** 23 **capacidad_zona** 281 **prop_riesgo** 283 **tasa_riesgo** 283 **tasa_mayores** 283 **poblacion_mayores** 283 **nombre_zona** 286 **nsec** 37 **t3_1** 283 **t1_1** 281 **t2_1** 283 **t2_2** 283 **t4_1** 283 **t4_2** 283 **t4_3** 283 **t5_1** 283 **t6_1** 283 **t7_1** 283 **t8_1** 283 **t9_1** 283 **t10_1** 283 **t11_1** 283 **t12_1** 283 **area** 287 **densidad_hab_km** 283 **tuits_gripe** 63 **interes_gripe** 48 **is_train** 2

[73]:
```
# Columns with a single unique value
#
# Identify columns with a single unique value
cols_to_remove <- sapply(data, function(x) length(unique(x)) == 1)
data <- data[, !cols_to_remove]
```

0.12 Very Few Values

Select rate

[74]:
```
# Select rate
threshold <- 0.8
```

Operation

[75]:
```
# Show features with over rate rows being the same value
cols_to_keep <- sapply(data, function(x) {
  freqs <- table(x) / length(x)
  max(freqs) <= threshold
})
print(cols_to_keep)
```

```
       GEOCODIGO            DESBDT               ano            semana
            TRUE              TRUE              TRUE              TRUE
       n_vacunas          n_citas              tmed              prec
            TRUE              TRUE              TRUE              TRUE
        velmedia          presMax           benzene                co
            TRUE              TRUE              TRUE              TRUE
              no               no2               nox                o3
            TRUE              TRUE              TRUE              TRUE
            pm10             pm2.5               so2          campana
            TRUE              TRUE              TRUE              TRUE
        scampana    capacidad_zona       prop_riesgo       tasa_riesgo
            TRUE              TRUE              TRUE              TRUE
    tasa_mayores poblacion_mayores       nombre_zona              nsec
            TRUE              TRUE              TRUE              TRUE
            t3_1              t1_1              t2_1              t2_2
            TRUE              TRUE              TRUE              TRUE
            t4_1              t4_2              t4_3              t5_1
            TRUE              TRUE              TRUE              TRUE
            t6_1              t7_1              t8_1              t9_1
            TRUE              TRUE              TRUE              TRUE
           t10_1             t11_1             t12_1              area
            TRUE              TRUE              TRUE              TRUE
 densidad_hab_km       tuits_gripe     interes_gripe          is_train
            TRUE              TRUE              TRUE              TRUE
```

[ ]:
```

```

[76]:
```
# Summarize the number of unique values in each column
# followed by the percentage of unique values for each
# variable as a percentage of the total number of rows
# in the dataset.

# First, find the number of unique values in each column
```

```r
num_unique_values <- sapply(data, function(x) length(unique(x)))

# Then, calculate the percentage of unique values as a proportion of total rows
percentage_unique_values <- num_unique_values / nrow(data) * 100

# Finally, create a data frame to summarize the results
summary_df <- data.frame(
  Column = names(data),
  UniqueValues = num_unique_values,
  PercentageOfUniqueValues = percentage_unique_values
)

# Print the summary
print(summary_df)
```

	Column	UniqueValues	PercentageOfUniqueValues
GEOCODIGO	GEOCODIGO	286	1.315789e+00
DESBDT	DESBDT	283	1.301987e+00
ano	ano	3	1.380199e-02
semana	semana	52	2.392344e-01
n_vacunas	n_vacunas	699	3.215863e+00
n_citas	n_citas	671	3.087045e+00
tmed	tmed	21736	1.000000e+02
prec	prec	19422	8.935407e+01
velmedia	velmedia	20888	9.609864e+01
presMax	presMax	21451	9.868881e+01
benzene	benzene	17215	7.920040e+01
co	co	18303	8.420593e+01
no	no	17020	7.830328e+01
no2	no2	20023	9.211907e+01
nox	nox	19397	8.923905e+01
o3	o3	20831	9.583640e+01
pm10	pm10	16586	7.630659e+01
pm2.5	pm2.5	19704	9.065145e+01
so2	so2	9453	4.349006e+01
campana	campana	4	1.840265e-02
scampana	scampana	23	1.058152e-01
capacidad_zona	capacidad_zona	281	1.292786e+00
prop_riesgo	prop_riesgo	283	1.301987e+00
tasa_riesgo	tasa_riesgo	283	1.301987e+00
tasa_mayores	tasa_mayores	283	1.301987e+00
poblacion_mayores	poblacion_mayores	283	1.301987e+00
nombre_zona	nombre_zona	287	1.320390e+00
nsec	nsec	37	1.702245e-01
t3_1	t3_1	283	1.301987e+00
t1_1	t1_1	281	1.292786e+00
t2_1	t2_1	283	1.301987e+00

t2_2	t2_2	283	1.301987e+00
t4_1	t4_1	283	1.301987e+00
t4_2	t4_2	283	1.301987e+00
t4_3	t4_3	283	1.301987e+00
t5_1	t5_1	283	1.301987e+00
t6_1	t6_1	283	1.301987e+00
t7_1	t7_1	283	1.301987e+00
t8_1	t8_1	283	1.301987e+00
t9_1	t9_1	283	1.301987e+00
t10_1	t10_1	283	1.301987e+00
t11_1	t11_1	283	1.301987e+00
t12_1	t12_1	283	1.301987e+00
area	area	287	1.320390e+00
densidad_hab_km	densidad_hab_km	283	1.301987e+00
tuits_gripe	tuits_gripe	63	2.898417e-01
interes_gripe	interes_gripe	48	2.208318e-01
is_train	is_train	2	9.201325e-03

[ ]:

0.13 Low Variance

A) Calculating variances

[77]:
```
# calculate variance for all columns
variances <- sapply(data, var, na.rm = TRUE)

# print the variances
print(variances)
```

```
Warning message in FUN(X[[i]], …):
"NAs introduced by coercion"
Warning message in FUN(X[[i]], …):
"NAs introduced by coercion"
```

GEOCODIGO	DESBDT	ano	semana
6.816564e+03	NA	2.908721e-01	2.660441e+02
n_vacunas	n_citas	tmed	prec
2.445457e+04	2.212373e+04	5.817079e+01	5.758088e+00
velmedia	presMax	benzene	co
8.066028e-01	3.752815e+02	1.062871e-01	1.467896e-02
no	no2	nox	o3
1.053120e+02	1.251476e+02	6.814005e+02	3.936654e+02
pm10	pm2.5	so2	campana
1.422510e+02	2.479786e+01	5.606923e-01	1.736922e-01
scampana	capacidad_zona	prop_riesgo	tasa_riesgo
2.369290e+01	6.816768e+06	4.380610e-03	2.251756e-05
tasa_mayores	poblacion_mayores	nombre_zona	nsec

1.030401e-04	3.620811e-03	NA	5.354712e+01
t3_1	t1_1	t2_1	t2_2
1.215538e+01	1.090875e+08	3.583480e-04	3.283615e-04
t4_1	t4_2	t4_3	t5_1
1.722349e-03	1.507711e-03	3.620811e-03	3.444767e-03
t6_1	t7_1	t8_1	t9_1
6.470828e-03	2.987716e-04	3.151205e-04	2.353691e-02
t10_1	t11_1	t12_1	area
1.403804e-03	3.810876e-03	3.519760e-03	4.019781e+15
densidad_hab_km	tuits_gripe	interes_gripe	is_train
1.760075e+08	1.932409e+04	6.425895e+02	1.600294e-01

B) Automatic calculation and representation of variances

Define thresholds to check

[ ]:
```
# define thresholds to check
thresholds = 0.8
```

Operation

[ ]:
```
```

C) Delete variables with low variance

Select column

[80]:
```
# Identify numeric columns
numeric_cols <- sapply(data, is.numeric)

# Calculate variance for numeric columns only
numeric_variances <- sapply(data[, numeric_cols], var, na.rm = TRUE)

# Set a threshold for variance
var_threshold = 0.1

# Find columns that have variance greater than the threshold
cols_to_keep <- c(!numeric_cols, numeric_cols_to_keep)
```

```
Error in eval(expr, envir, enclos): object 'numeric_cols_to_keep' not found
Traceback:
```

Operation

[82]:
```
# Keep only those columns
data <- data[, cols_to_keep]
```

[ ]:
```
```

0.14 Duplicates

Entendido como ERROR –> Eliminar duplicados

[86]:
```
data <- data[!duplicated(data), ]
```

0.15 Data Save

- Solo si se han hecho cambios
- No aplica

Identificamos los datos a guardar

[ ]:
```
data_to_save <- data
```

Estructura de nombre de archivos:

- Código del caso de uso, por ejemplo “CU_04”
- Número del proceso que lo genera, por ejemplo ”_06”.
- Resto del nombre del archivo de entrada
- Extensión del archivo

Ejemplo: ”CU_04_06_01_01_zonasgeo.json, primer fichero que se genera en la tarea 01 del proceso 05 (Data Collection) para el caso de uso 04 (vacunas) y que se ha transformado en el proceso 06

Importante mantener los guiones bajos antes de proceso, tarea, archivo y nombre

0.15.1 Proceso 09.1

[90]:
```
caso <- "CU_04"
proceso <- '_09.1'
tarea <- "_20"
archivo <- ""
proper <- "_vacunacion_gripe_train_and_test"
extension <- ".csv"
```

OPCION A: Uso del paquete “tcltk” para mayor comodidad

- Buscar carpeta, escribir nombre de archivo SIN extensión (se especifica en el código)
- Especificar sufijo2 si es necesario
- Cambiar datos por datos_xx si es necesario

[ ]:
```
# file_save <- paste0(caso, proceso, tarea, tcltk::tkgetSaveFile(), proper,␣
↪extension)
# path_out <- paste0(oPath, file_save)
# write_csv(data_to_save_xxxx, path_out)

# cat('File saved as: ')
# path_out
```

OPCION B: Especificar el nombre de archivo

- Los ficheros de salida del proceso van siempre a Data/Output/.

[91]:
```
file_save <- paste0(caso, proceso, tarea, archivo, proper, extension)
path_out <- paste0(oPath, file_save)
write_csv(data, path_out)

cat('File saved as: ')
path_out
```

```
File saved as:
```

'Data/Output/CU_04_09.1_20_vacunacion_gripe_train_and_test.csv'

Copia del fichero a Input Si el archivo se va a usar en otros notebooks, copiar a la carpeta Input

[92]:
```
path_in <- paste0(iPath, file_save)
file.copy(path_out, path_in, overwrite = TRUE)
```

TRUE

0.16 REPORT

A continuación se realizará un informe de las acciones realizadas

0.17 Main Actions Carried Out

- Si eran necesarias se han realizado en el proceso 05 por cuestiones de eficiencia

0.18 Main Conclusions

- Los datos están limpios para el despliegue

0.19 CODE TO DEPLOY (PILOT)

A continuación se incluirá el código que deba ser llevado a despliegue para producción, dado que se entiende efectúa operaciones necesarias sobre los datos en la ejecución del prototipo

Description

- No hay nada que desplegar en el piloto, ya que estos datos son estáticos o en todo caso cambian con muy poca frecuencia, altamente improbable durante el proyecto.

CODE

[ ]:
```
```