10.- Imbalanced Analysis_25_01_listas_espera_v_01

June 10, 2023

#

CU25_Modelo de gestión de Lista de Espera Quirúrgica

Citizenlab Data Science Methodology > II - Data Processing Domain *** > # 10.- Imbalanced Analysis

Data Balancing is the process to obtain an adequate data balance if is required, in order to have the adequate amount of data that reflects the intrinsic structure of the problem to be solved.

0.1 Tasks

Imbalanced Analysis

Evaluate Imbalanced Classification Models

Select appropiate metrics

Data Balancing

- Undersampling the Majority Class
- Oversampling the Minority Class
- Mix under-oversampling
- Evaluate a model with random oversampling and undersampling

Cost-Sensitive Algorithms

0.2 File

- Input File: CU_25_09.2_01_lista_espera_completo_clean_v_01
- Output File: No aplica

0.2.1 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.

[20]:
```
Sys.setlocale(category = "LC_ALL", locale = "es_ES.UTF-8")
```

'LC_COLLATE=es_ES.UTF-8;LC_CTYPE=es_ES.UTF-8;LC_MONETARY=es_ES.UTF-8;LC_NUMERIC=C;LC_TIME=es_ES.UTF-8'

0.3 Settings

0.3.1 Libraries to use

[21]:
```
library(readr)
library(dplyr)
library(sf)
library(tidyr)
library(stringr)
```

0.3.2 Paths

[22]:
```
iPath <- "Data/Input/"
oPath <- "Data/Output/"
```

0.4 Data Load

OPCION A: Seleccionar fichero en ventana para mayor comodidad

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

[23]:
```
# file_data <- tcltk::tk_choose.files(multi = FALSE)
```

OPCION B: Especificar el nombre de archivo

[24]:
```
iFile <- "CU_25_09.2_01_lista_espera_completo_clean_v_01.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_25_09.2_01_lista_espera_completo_clean_v_01.csv
```

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

[31]:
```
data <- read.csv(file_data)
```

Visualizo los datos.

Estructura de los datos:

[26]:
```
data |> glimpse()
```

```
Rows: 55,216
Columns: 46
$ Hospital          <chr> "HOSPITAL REY JUAN CARLOS",
"HOSPITAL CENTRAL DE LA …
```

```
$ Especialidad      <chr> "UROLOGÍA", "ODONTOESTOMATOLOGÍA",
"GINECOLOGÍA", "D…
$ total_pacientes   <dbl> 344, 0, 52, 37, 0, 4, 0, 718, 0,
271, 108, 0, 34, 86…
$ ano               <dbl> 2021, 2020, 2021, 2021, 2021, 2020,
2021, 2020, 2021…
$ semana            <dbl> 30, 36, 49, 23, 3, 5, 50, 7, 35, 1,
42, 10, 21, 33, …
$ CODCNH            <dbl> 281348, 280724, 281292, 281292,
281236, 280724, 2807…
$ id_area           <dbl> 8, 7, 11, 11, 11, 7, 3, 6, 1, 2, 2,
8, 11, 11, 1, 3,…
$ nombre_area       <chr> "SUR-OESTE I", "CENTRO-OESTE", "SUR
II", "SUR II", "…
$ cmunicipio        <dbl> 280920, 280796, 280133, 280133,
281610, 280796, 2800…
$ Municipio         <chr> "MÓSTOLES", "MADRID", "ARANJUEZ",
"ARANJUEZ", "VALDE…
$ CAMAS             <dbl> 382, 475, 98, 98, 182, 475, 507,
613, 269, 1143, 156…
$ Clase             <chr> "HOSPITALES GENERALES", "HOSPITALES
GENERALES", "HOS…
$ Dependencia       <chr> "SERVICIOS E INSTITUTOS DE SALUD DE
LAS COMUNIDADES …
$ TAC               <dbl> 2, 2, 1, 1, 1, 2, 3, 3, 0, 0, 1, 2,
6, 6, 1, 3, 4, 1…
$ RM                <dbl> 3, 2, 1, 1, 2, 2, 2, 3, 0, 0, 0, 2,
5, 5, 1, 2, 4, 1…
$ GAM               <dbl> 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
2, 2, 0, 0, 2, 0…
$ HEM               <dbl> 1, 2, 0, 0, 1, 2, 1, 2, 0, 0, 0, 1,
3, 3, 0, 1, 1, 0…
$ ASD               <dbl> 2, 1, 1, 1, 1, 1, 1, 3, 0, 0, 0, 1,
2, 2, 0, 1, 2, 1…
$ ALI               <dbl> 1, 2, 0, 0, 0, 2, 0, 4, 0, 0, 0, 0,
3, 3, 0, 2, 2, 0…
$ SPECT             <dbl> 1, 1, 0, 0, 0, 1, 0, 4, 0, 0, 0, 0,
3, 3, 0, 0, 0, 0…
$ MAMOS             <dbl> 2, 1, 1, 1, 1, 1, 2, 2, 0, 0, 1, 2,
3, 3, 1, 1, 3, 1…
$ DO                <dbl> 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1,
2, 2, 0, 1, 2, 0…
$ DIAL              <dbl> 20, 24, 13, 13, 17, 24, 28, 31, 0,
0, 0, 28, 43, 43,…
$ X                 <dbl> -3.870412, -3.745529, -3.610795,
-3.610795, -3.69744…
$ Y                 <dbl> 40.33920, 40.38791, 40.05726,
40.05726, 40.19884, 40…
```

```
$ t3_1              <dbl> 42.34715, 45.37878, 42.06149,
42.06149, 42.06149, 45…
$ t1_1              <dbl> 532487, 511605, 899702, 899702,
899702, 511605, 3830…
$ t2_1              <dbl> 0.5122493, 0.5296804, 0.5240445,
0.5240445, 0.524044…
$ t2_2              <dbl> 0.4877507, 0.4703198, 0.4759555,
0.4759555, 0.475955…
$ t4_1              <dbl> 0.1659665, 0.1054260, 0.1540793,
0.1540793, 0.154079…
$ t4_2              <dbl> 0.6371549, 0.6742432, 0.6753787,
0.6753787, 0.675378…
$ t4_3              <dbl> 0.1968769, 0.2203341, 0.1705449,
0.1705449, 0.170544…
$ t5_1              <dbl> 0.1137647, 0.1744493, 0.1747059,
0.1747059, 0.174705…
$ t6_1              <dbl> 0.1604646, 0.2629599, 0.2641879,
0.2641879, 0.264187…
$ t7_1              <dbl> 0.05422176, 0.05481008, 0.04898547,
0.04898547, 0.04…
$ t8_1              <dbl> 0.04120012, 0.04653221, 0.03679912,
0.03679912, 0.03…
$ t9_1              <dbl> 0.3348780, 0.4914365, 0.3346063,
0.3346063, 0.334606…
$ t10_1             <dbl> 0.13692541, 0.12170996, 0.15173209,
0.15173209, 0.15…
$ t11_1             <dbl> 0.5072726, 0.4915713, 0.5024130,
0.5024130, 0.502413…
$ t12_1             <dbl> 0.5849309, 0.5597213, 0.5900028,
0.5900028, 0.590002…
$ capacidad         <dbl> 17, 0, 8, 5, 0, 5, 1, 24, 6, 6, 30,
4, 2, 15, 20, 6,…
$ pacientes         <dbl> 1447, 1211, 1293, 1501, 1240, 1504,
1502, 1533, 1463…
$ consultas         <dbl> 573, 45, 108, 103, 44, 42, 36,
1119, 34, 466, 220, 6…
$ hospitalizaciones <dbl> 12, 0, 2, 2, 0, 1, 0, 4, 0, 12, 3,
0, 2, 4, 1, 2, 15…
$ Target            <dbl> 54.45, 0.00, 37.96, 23.14, 0.00,
6.25, 0.00, 78.20, …
$ is_train          <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE,
TRUE, TRUE, TRUE…
```

Muestra de los primeros datos:

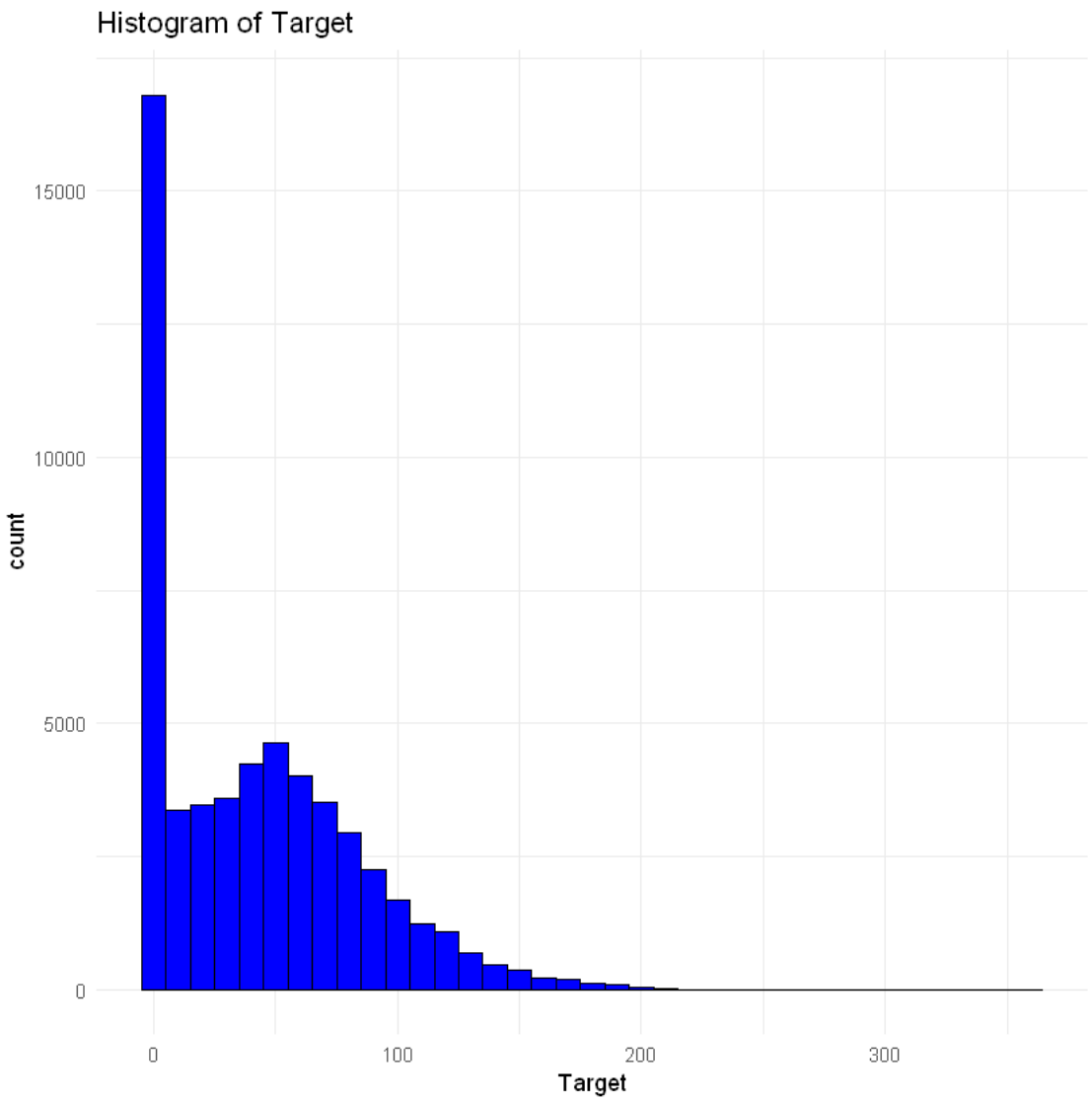
[27]:
```
data |> slice_head(n = 5)
```

	Hospital <chr>	Especialidad <chr>
	HOSPITAL REY JUAN CARLOS	UROLOGÍA
A spec_tbl_df: 5 × 46	HOSPITAL CENTRAL DE LA DEFENSA GOMEZ ULLA	ODONTOESTOMATOLO
	HOSPITAL UNIVERSITARIO DEL TAJO	GINECOLOGÍA
	HOSPITAL UNIVERSITARIO DEL TAJO	DERMATOLOGÍA
	HOSPITAL UNIVERSITARIO INFANTA ELENA	ODONTOESTOMATOLO

0.5 Imbalanced Analysis

[28]:
```r
# Visualizando los datos para comprobar como de balanceados están
# If not already installed, install the ggplot2 package
if(!require(ggplot2)) install.packages('ggplot2')

# Load the ggplot2 package
library(ggplot2)

# Select the column name
column_name <- "Target"  # replace with your column name

# Create a histogram of the numeric column
ggplot(data, aes_string(x = column_name)) +
  geom_histogram(binwidth = 10, fill = "blue", color = "black") +
  theme_minimal() +
  ggtitle(paste("Histogram of", column_name))

# Calculate basic statistical measures
summary(data[[column_name]])
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   0.00    0.00   35.90   42.25   68.49  356.00 
```



[29]:
```
# summarize class distribution
```

[30]:
```
# Generate and plot imbalanced classification dataset
```

0.6 Evaluate Imbalanced Classification Models

No aplica

0.7 Undersampling the Majority Class

No aplica

0.8 Oversampling the Minority Class

No aplica

0.9 Combine Data Undersampling and Oversampling with SMOTEENN

No aplica

0.10 Evaluating a model with random oversampling and undersampling

No aplica

0.11 Cost-Sensitive Algorithms

No aplica

0.12 Data Save

- 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

0.12.1 Proceso 10

[1]:
```
# caso <- "CU_XX"
# proceso <- '_10'
# tarea <- "_XX"
# archivo <- ""
# proper <- "_xxxxx"
# 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/.

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

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

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

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

0.13 REPORT

A continuación se realizará un informe de las acciones realizadas

0.14 Main Actions Carried Out

- No aplica el proceso al caso

0.15 Main Conclusions

- Al no haber clasificación no procede analizar el desbalanceo.

0.16 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

[ ]:
```

```