18.- Feature Construction_04_19_vacunacion_completo_v_01

June 8, 2023

#

CU04_Optimización de vacunas

Citizenlab Data Science Methodology > III - Feature Engineering Domain *** > # 18.- Feature Construction

Feature Construction is the process related to create new features from your existing ones to improve model performance.

0.1 Tasks

Feature Construction - Create Interaction Features - Create derived variables - Combine Sparse Classes - Explore Binning for Feature Construction

0.2 Consideraciones casos CitizenLab programados en R

- Algunas 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
- Otras tareas típicas de este proceso se realizan en los notebooks del dominio IV al ser más eficiente realizarlas en el propio pipeline de modelización.
- 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: No aplica

0.3.1 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.

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

0.4 Settings

0.4.1 Libraries to use

[ ]:
```
library(readr)
library(dplyr)
library(tidyr)
library(forcats)
library(lubridate)
```

0.4.2 Paths

[ ]:
```
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

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

OPCION B: Especificar el nombre de archivo

[1]:
```
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.")
}
```

```
  Cell In[1], line 4
    if(file.exists(file_data)){
                               ^

SyntaxError: invalid syntax
```

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

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

Estructura de los datos:

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

Muestra de los primeros datos:

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

0.6 Creating Interaction Features

Ver notebooks del proceso 05 Data Collectio

0.7 Creating derived variables

Ver notebooks del proceso 05 Data Collectio

0.8 Combining Sparse Classes

Ver notebooks del proceso 05 Data Collectio

0.9 Binning for Feature Construction

Ver notebooks del proceso 05 Data Collectio