## ETL: TIPOS DE DATOS

Vamos a tratar brevemente los tipos de datos o mejor tipo de valores que podemos encontrarnos en un conjunto de datos desde una perspectiva de Data Scientist. Es decir, no hablamos de tipos de datos en genérico como en píldoras anteriores, ni tampoco de tipos de datos de Python, numpy o pandas.

Para ello carguemos una vez más nuestro dataset de viajes aéreos (que pronto nos dejará por otros datos diferentes):

[1]:

**import** pandas **as** pd

**import** random

df\_datos **=** pd.read\_csv("./data/dataset\_ETL\_full\_aviones.csv")

## Tipos de datos

En el contexto de la ciencia de datos o Data Science se suele hablar de los siguientes tipos de datos (no es una lista exahustiva pero sí son todos los que están):

* Numéricos
* Categóricos (incluye booleanos)
* Tipo Texto
* Tipo Fecha
* Multimedia: Imágenes, vídeos, Audio

Hasta que tratemos el procesamiento de lenguaje natural (NLP) y las imágenes con redes, nosotros nos vamos a centrar en lo que denominamos datos estructurados o tabulares en las sesiones anteriores y dentro de este conjunto de datos en los tipos de valores no multimedia. Véamos que pinta tiene cada uno

### Numéricos

Es decir, los campos o características o columnas que contienen valores numéricos y por tanto susceptibles de ser operados matemáticamente. Si miramos nuestro dataset:

[2]:

df\_datos.info()

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

"Distancia","consumo\_kg","duracion", "ingresos" y el indicador "IC" son valores numéricos. Podemos calcular sus medias, varianzas, distribuciones, ver esas distribuciones gráficamente, etc.

[6]:

df\_datos["IC"].value\_counts()

[6]:

Tabla

Descripción generada automáticamente

df\_datos["IC"].hist()

Gráfico, Histograma

Descripción generada automáticamente

Muchas veces es interesante en la misma exploración para "transformar y limpiar" hacer pequeñas visualizaciones para entender cómo están los datos.

Para hacer un EDA no es necesario, pero cuando entremos en Machine y Deep Learning veremos que todos los valores, todas las columnas deben pasar a valores numéricos y ya veremos cómo hacerlo.

### Categóricos

Con categóricos nos referimos a los valores (columnas en nuestro caso) de datos cuyo rango de valores está limitado y que suelen representar categorías o etiquetas. Por ejemplo, la columna "Categoria\_IC":

[16]:

df\_datos["Categoria\_IC"].value\_counts()

Texto

Descripción generada automáticamente

df\_datos["Origen"].value\_counts()

[17]:

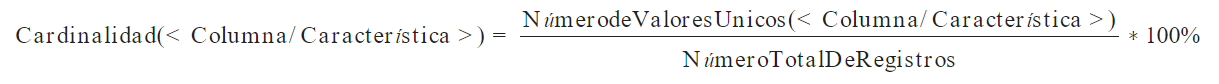
Tabla

Descripción generada automáticamente

len(df\_datos.Destino.unique())

11

El proceso para que pensemos en clasificar un dato, columna o característica como categórica (además de que nos lo digan exprofeso) es el de calcular la cardinalidad:



Por ejemplo, la cardinalidad para Categoria\_IC

[22]:

cardinalidad\_CatIC**=**len(df\_datos["Categoria\_IC"].unique())**/**(len(df\_datos))**\***100

print(round(cardinalidad\_CatIC,2),"%")

0.26 %

La de Destino (empleando el método de pandas, nunique):

[23]:

cardinalidad\_destino**=**df\_datos["Destino"].nunique()**/**len(df\_datos)**\***100

print(round(cardinalidad\_destino,2),"%")

0.94 %

Menos del 1%, compara ahora con la de los valores numéricos:

cardinalidad\_consumo=df\_datos["consumo\_kg"].nunique()/len(df\_datos)\*100

print(round(cardinalidad\_consumo,2),"%")

71.1 %

Prueba tu a calcular la cardinalidad de "ingresos", por ejemplo.

Con cardinalidades por debajo del 5-10% ya podríamos considerar el valor o columna como categórico. ¿Y eso por qué es importante? Por dos motivos principales:

1. Para el EDA: Suponen, en general, interesantes columnas o características para hacer agrupaciones y explorarlas
2. Para Machine y Deep learnnig: Existen métodos de transformación a números bastante potentes y utilizados, por lo que suelen ser buenas características para incluir

En general, los datos categóricos suelen ser etiquetas de texto, pero podemos hacer datos "categóricos" de datos numéricos y los datos booleanos (como "con\_escala") se suelen considerar datos categóricos con dos categorías (Verdadero o Falso, 1 o 0)

Ah y no pierdas de vista el concepto de Cardinalidad de un conjunto de datos porque lo vamos a usar muchas más veces.

[Nota: Estrictamente hablando cardinalidad es el número de valores diferentes, pero nosotros vamos a emplearlo como el porcentaje mostrado antes]

### Tipo Texto

Por valores de tipo texto entendemos aquellas columnas que son texto y no se pueden asimilar directamente a una categoría. Consideremos por ejemplo la columna "Incidencias":

[26]:

df\_datos["Incidencias"].value\_counts()

[26]:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

df\_datos["Incidencias"].nunique()

154

card=df\_datos["Incidencias"].nunique()/len(df\_datos)\*100

print(card)

13.20754716981132

Considerando la cardinalidad, podríamos intentar hacer de esta columna una categórica, pero observando el tipo de dato que tiene, con lenguaje natural que puede cambiar y en general es de por si desestructurado y donde las abreviaturas, sinónimos, etc, lo hacen difícil de considerar de primeras, lo normal es considerarlo como un campo de texto libre y tratarlo a partir de ahí.

En general, lo trataremos con técnicas de procesamiento de lenguaje natural y, otras veces, además, obtendremos otras columnas con datos resumidos a partir de estas columnas de texto.

### Tipo Fecha

Como su nombre indica hace referencia a los valores que son fechas. El hecho de que no sigan el sistema decimal, y tengan sus particularidades de calendario (vamos que 45 minutos más 55 minutos no sean 1 hora, sino 1 hora y 30 minutos, y que del 28 de febrero a veces pasemos al 29 y otras al 1 de marzo, por ejemplo) hace que haya que tratarlos de una forma un poco especial o de aprovechar las capacidades de Python y pandas para hacerlo.

En nuestro dataset, claramente "Hora\_Vuelo" es de este tipo.

[31]:

df\_datos["Hora\_Vuelo"].value\_counts()

[31]:

Texto

Descripción generada automáticamente

Pero si miramos el tipo que considera pandas:

[ ]:

df\_datos["Incidencias"].dtypes

Hace referencia a que es un "object" o sea un string o cualquier mezcla de tipos. Tendremos que hacer conversiones especiales y tratarlas con cariño para sacar el potencial a las fechas. Lo veremos en la píldora correspondiente.