INFORME SOBRE CALIDAD DE LOS DATOS

MANUEL RODRÍGUEZ VILLEGAS

En este caso, se nos presenta un Datset con la información relativa a la venta de pizzas de la misma pizzería que en el análisis anterior pero del año 2016. Sin embargo, en este caso los datos no vienen ordenados, sino que traen distintos formatos, datos ausentes, caracteres intercambiados, étc. Por este motivo, este informe se centrará en explorar los datos que se nos presentan con el objetivo de transformarlos para poder trabajar con ello de la misma manera que se hizo con los datos de 2015.

Los datos desordenados son los correspondientes con los Datasets *orders_id* y *order_details_id*. En particular, los datos ahora vienen separados por ; en vez de ,.

Una vez que cargamos los datos, podemos empezar a limpiarlos. Comenzaremos por *order-details_id*. En este Dataset las columnas de *pizza_id* y *quantity* contienen datos de distinto formato, por lo que vamos a ver cómo transformarlos. En primer lugar, podemos observar que los nombres de las pizzas por lo general están separados de su tamaño por un guion bajo. No obstante, en algunos casos aparecen separados por un espacio o por un guion normal. Además, observamos que algunas “e” están cambiadas por 3, al igual que algunas “a”, “o”, que están cambiadas por @ y 0, respectivamente. Por otro lado, en la columna de *quantity* tenemos algunos “One” o “one” en vez de “1”, así como “two” en vez de “2”. Todo esto se puede solucionar haciendo uso de expresiones regulares, más en concreto con el comando “sub”. Además, observamos que tenemos 5673 valores NaN en *pizza_id* y 4726 NaN en *quantity*. Para la columna de *pizza_id*, rellenaremos estos valores con pizzas aleatorias con el objetivo de mantener la misma estructura y no influir en exceso en el análisis final. Para la columna *quantity*, rellenaremos las NaN con 1, ya que es el valor correspondiente a la inmensa mayoría de los casos.

El segundo Dataset con valores desordenados es *orders_id*, donde los cambios de formato afectan a las columnas de *date* y *time*. Por esta razón, nos valdremos del módulo *datetime* para convertir las fechas y horas al formato que más nos convenga, en este caso 01/01/2016 para las fechas y 20:03:53 para las horas, auqnue esta columna no la convertiremos ya que no es relevate para nuestro análisis. Utilizaremos la función “strptime” para convertir a un formato estándar, así como la función *recognize_format*, creada manualmente, para reconocer el formato. En este

Dataset también encontramos valores NaN, pero en este caso no los rellenaremos de forma aleatoria sino fijándonos en los valores adyacentes. Concretamente, observamos 2353 NaNs en *date* y 2038 en *time*.