CUNEF ETL - Assessment

Master DS - Leonardo Hansa

Octubre, 2021

Tienes una base de datos con información sobre alojamientos de Airbnb. La tarea consiste en:

- extraer parte de esos datos,
- construir nuevas tablas transformándolos con ciertas directrices
- y subirlas de nuevo a la base de datos.

Observación. La metodología nos sirve para fines académicos, aunque en un proyecto real es posible que el origen de datos y el destino fueran diferentes.

Objetivo

Construye dos tablas.

- **Tabla 1.** Evolución mensual del número de críticas por distrito, con predicción para el mes siguiente.
- **Tabla 2.** Distribución del tipo de alojamiento `room_type` por distrito. Incluye:
  - Nota media ponderada (`review_scores_rating` ponderado con `number_of_reviews`).
  - Precio mediano (`price`).
  - Número de alojamientos (`id`).

Datos

La base de datos es el fichero `airbnb.sqlite`. Tiene tres tablas:

- **Listings.** Datos sobre los alojamientos de Madrid listados en julio de 2021.
- **Reviews.** Listado con comentarios hechos sobre cada alojamiento, con fecha e identificador del inquilino.
- **Hoods.** Relación entre barrio (`neighbourhood`) y distrito (`neighbourhood_group`).

Herramientas

- Realiza la extracción con SQL. Extrae únicamente los datos necesarios. Nada de descargar datos de más *por si acaso.*
- El resto del trabajo lo puedes hacer en R ó Python (mezclando lenguajes o usando uno solo, como prefieras).
- El formato de entrega te lo cuento en clase.

Tareas

De ahora en adelante tienes detalles de los pasos necesarios.

Extracción

1. **Extracción (listings)** Descarga la tabla *listings* en un data frame de `pandas` o de R. Con SQL, haz un *join* con la tabla *hoods* para añadir el dato de distrito (`neighbourhood_group`) y asegúrate de que extraes esta columna en el data frame en lugar de `neighbourhood`.

2. **Extracción (reviews)** Descarga la tabla *reviews* en un data frame de `pandas` o de R, con las siguientes consideraciones:

   - Con SQL, haz un *join* con la tabla *hoods* para añadir el dato de distrito (`neighbourhood_group`) y asegúrate de que extraes esta columna en el data frame en lugar de `neighbourhood`.
   - También en SQL, cuenta a nivel de distrito y mes el número de reviews. Para calcular el mes a partir de una fecha en una tabla SQL, usa `strftime('%Y-%m', date) as mes`.
   - Además, extrae los datos desde 2011 en adelante (también SQL). Te resultará de nuevo útil la función `strftime`. **Observación.** Esta función devuelve un texto.

Transformación

3. **Transformación (listings).** Antes de realizar la agregación que se pide, tienes tratar las columnas `price,`, `number_of_reviews` y `review_scores_rating`. Empieza con el precio. Necesitas pasarla a numérica. Ahora mismo es de tipo texto y lo primero que necesitamos es quitar símbolos raros. Tanto R como Python sabe convertir un texto como `"15.00"` a número, pero no saben convertir `"$1,400.00"`. Tienes que quitar tanto el símbolo del dólar como la coma. En expresiones regulares, el símbolo del dólar se usa para una cosa muy concreta, así que necesitarás usar algo como `"\\$"` (lo que se conoce como *escapar*).

4. **Transformación (listings).**

   - (*Opción A.*) Toca imputar los valores missing de `number_of_reviews` y `review_scores_rating`. Normalmente en estos casos se habla con la gente que más usa los datos y se llega con ellos a un acuerdo de cómo se imputaría esta información. En este caso, imputa los valores missing con valores reales dentro de la tabla escogidos de manera aleatoria. Es decir, si hay un valor missing en `number_of_reviews` lo reemplazarías con un valor aleatorio de esa misma columna. Tienes libertad para plantear esto como te resulte más cómodo.
   - (*Opción B.*) Toca imputar los valores missing de `number_of_reviews` y `review_scores_rating`. Normalmente en estos casos se habla con la gente que más usa los datos y se llega con ellos a un acuerdo de cómo se imputaría esta información. En este caso, imputa los valores missing con valores reales dentro de la tabla, a nivel de `room_type`, escogidos de manera aleatoria. Es decir, si hay un valor missing en `number_of_reviews` para un registro con `room_type == "Entire home/apt"`, lo reemplazarías con un valor aleatorio de esa misma columna para los que `room_type` sea `"Entire home/apt"`. Tienes libertad para plantear esto como te resulte más cómodo. *Pista.* Yo he hecho un bucle *for()* con R base (sí, lo nunca visto en mí :P)

5. **Transformación (listings).** Con los missing imputados y el precio en formato numérico ya puedes agregar los datos. A nivel de distrito y de tipo de alojamiento, hay que calcular:

   - Nota media ponderada (`review_scores_rating` ponderado con `number_of_reviews`).
   - Precio mediano (`price`).
   - Número de alojamientos (`id`).

La tabla resultante tendrá cuatro columnas: distrito (llamada habitualmente `neighbourhood_group`), tipo de alojamiento (`room_type`), nota media y precio mediano. Esta tabla puede ser útil para estudiar diferencias entre mismo un tipo de alojamiento en función del distrito en el que esté.

6. **Transformación (reviews).** La mayor parte de la transformación para *Reviews* la has hecho ya con SQL. Vamos a añadir ahora a simular que tenemos un modelo predictivo y lo vamos a aplicar sobre nuestros datos. Así, la tabla que subamos de nuevo a la base de datos tendrá la predicción añadida. El último mes disponible es julio, así que daremos la predicción para agosto. Esto no es una asignatura de predicción de series temporales, así que nos vamos a conformar con tomar el valor de julio como predicción para agosto (a nivel de distrito). Es decir, si el dato en `"Centro"` para julio es de 888 reviews, añadiremos una fila con los valores `"Centro"`, `"2021-08"` y `888`, así para cada distrito. Tienes libertad para plantearlo como veas adecuado. **Al final, deja el data frame ordenado a nivel de distrito y mes.** *Pista.* Yo he creado un data frame nuevo con todas estas predicciones y lo he apilado al data frame original. Esto se puede hacer con la función `bind_rows()` de dplyr o el método `append()` o función `concat()` de pandas.

7. (*Extra*) **Transformación (reviews).** Hay casos que no tienen dato, por ejemplo, febrero de 2011 en Arganzuela. Como no hay dato, asumiremos que es 0. Siguiendo esta idea, añade todos los registros necesarios a la tabla. Puedes hacerlo de la manera que te resulte más intuitiva. **Recuerda ordenar la tabla final por distrito y mes.** *Pista.* Yo he creado primero un vector con todas las fechas posibles y otro con los posibles distritos. Con esos vectores hago un data frame de dos columnas, con todas las combinaciones posibles entre meses y distritos. Hay muchas formas de hacer eso. Luego hago un *full join* con los datos originales. Si después del *join* la columna *reviews* tiene valor missing, es que no estaba en el caso original. Sustituyo esos missing por ceros y ya tengo la tabla final.

8. **Carga.** Sube a la base de datos las dos tablas que has creado. No sobreescibas las que hay: crea dos tablas nuevas. Haz una prueba de que todo está en orden, haciendo `SELECT * FROM nombre_tabla LIMIT 10` para cada tabla. Si la fecha tiene un formato raro, es posible que necesites definirla en el data frame como tipo texto.
