Centro de Investigación en Computación

Instituto Politécnico Nacional

Julio 2023

Profesor: Alan Badillo Salas

Alumno: Daniel Miranda Badillo

dmirandacet@gmail.com

Práctica: Unir dos DataFrames de productos y ventas relacionados mediante un JOIN

INTRODUCCIÓN

La práctica está centrada en el manejo de DataFrames para hacer relación de datos.
El reto principal es conocer las operaciones de relación (“join”) para utilizarlas en pandas y saber que es lo que se espera recibir.

JUSTIFICACIÓN

Las funciones join son de las más utilizadas en la relación de tablas, siendo implementadas en bases de datos, y en funciones de SQL. La relación de tablas es importante en el análisis de datos ya que nos permite ampliar la variedad de datos para hacer un análisis o explotar la información.
Las tablas por separado aportan información valiosa para el análisis de datos, pero el establecer relaciones entre datos incrementa algunas caracteristicas de datos como Volumen, Variedad y Valor de los datos.

Pasos para resolución de práctica

Importar librería pandas para manejo de datos y dataframes

```
#importar pandas
import pandas as pd
```

Se genera el dataFrame de Productos

```
#crear dataFrame de productos
df_productos = pd.DataFrame({"PRODUCTO_ID":[1,2,3,4,5],
"NOMBRE":["Coca Cola","Galletas Marías","Gansito","Pepsi","Cehetos"],
"PRECIO":[17.5,18.9,21.3,16.0,11.5]})
```

VENTA_ID	PRODUCTO_ID	FECHA
101	1	2023-07-05 11:55:00
101	3	2023-07-05 11:55:00
101	1	2023-07-05 11:55:00
102	2	2023-07-05 11:57:00
102	2	2023-07-05 11:57:00
102	3	2023-07-05 11:57:00
103	5	2023-07-05 12:31:00
103	4	2023-07-05 12:31:00

Se genera el dataFrame de Ventas

```
df_ventas = pd.DataFrame({"VENTA_ID": [101,101,101,102,102,102,103,103 ],
                          "PRODUCTO_ID": [1,3,1,2,2,3,5,4],
                               "FECHA": ["2023-07-05 11:55:00 ","2023-07-05 11:55:00 ", "2023-07-05
11:55:00 ","2023-07-05 11:57:00 ","2023-07-05 11:57:00 ","2023-07-05 11:57:00 ","2023-07-05 12:31:00
","2023-07-05 12:31:00 "]
                          })
```

PRODUCTO_ID	NOMBRE	PRECIO
1	Coca Cola	17.500000
2	Galletas Marías	18.900000
3	Gansito	21.300000
4	Pepsi	16.000000
5	Chetos	11.500000

```
# Crea un DataFrame que extienda el DataFrame de Ventas con los productos usando el operador JOIN
df_extendido = pd.merge(df_productos,df_ventas, how='left', on='PRODUCTO_ID')
display(df_extendido)
```

Se utilizó la función merge de pandas para realizar el join de las tablas.
En este caso como ambas tablas tienen registros con información completa, es decir que no cuentan con campos nulos se utilizó una función de unión tipo “left”, teniendo los mismos resultados al usar una tipo “right”.

Resultado:

	PRODUCTO_ID	NOMBRE	PRECIO	VENTA_ID	FECHA
0	1	Coca Cola	17.5	101	2023-07-05 11:55:00
1	1	Coca Cola	17.5	101	2023-07-05 11:55:00
2	2	Galletas Marías	18.9	102	2023-07-05 11:57:00
3	2	Galletas Marías	18.9	102	2023-07-05 11:57:00
4	3	Gansito	21.3	101	2023-07-05 11:55:00
5	3	Gansito	21.3	102	2023-07-05 11:57:00
6	4	Pepsi	16.0	103	2023-07-05 12:31:00
7	5	Chetos	11.5	103	2023-07-05 12:31:00

Posterior a tener nuestra tabla extendida (unión de los dos DataFrames), se realiza un agrupamiento con la función gropuBy de pandas, a esta función se le requiere pasar como argumento el campo llave (“el campo que tienen en común ambas tablas”) y agregar la función .sum() para que al objeto agrupado se le aplique una función de suma. Se podrian realizar otras operaciones de agrupación como .count() o .mean().

```
# Agrupa el DataFrame extendido para calcular el monto total por venta (suma la columna precio)
df_group_venta = df_extendido.groupby(by="VENTA_ID").sum()
display(df_group_venta)
```

Como resultado se muestra una tabla agrupada por el numero de venta y los datos agrupados regresan la suma de todos los productos incluidos en esa venta. La columna PRODUCTO_ID no se toma en cuenta ya que no tiene sentido sumar una valor categórico.

	PRODUCTO_ID	PRECIO
VENTA_ID		
101	5	56.3
102	7	59.1
103	9	27.5

Se realizó un segundo agrupamiento tomando como llave el ID de producto vendido,

```
# Agrupa el DataFrame extendido para calcular el monto total por producto
(suma la columna precio)
df_group_producto =
df_extendido.groupby(by="PRODUCTO_ID").sum().drop(["VENTA_ID"], axis=1)
display(df_group_producto)
```

Como resultado se obtiene un DataFrame donde por se ha agrupado por producto y se regresa la suma de los precios del producto vendido n veces en las distintas ventas.

	PRECIO
PRODUCTO_ID	
1	35.0
2	37.8
3	42.6
4	16.0
5	11.5

Se ha realizado la unión entre dos DataFrames para representar el id de producto, la suma de precio todos los productos vendidos (“PRECIO_x”), el nombre del producto asociado a al id y el precio por unidad.
Nota: cuando se hace el join de dos tablas que contienen registros distintos con mismo nombre de columna, pandas agrega el sufijo _x o _y para corregir un probable error. En la definición de tablas y dataFrame no puede haber dos columnas con el mismo nombre.

```
# Reporta los valores de los montos obtenidos en la impresión estándar
reporte = pd.merge(df_group_producto ,df_productos, how='left',
on='PRODUCTO_ID')
display(reporte)
```

	PRODUCTO_ID	PRECIO_x	NOMBRE	PRECIO_y
0	1	35.0	Coca Cola	17.5
1	2	37.8	Galletas Marías	18.9
2	3	42.6	Gansito	21.3
3	4	16.0	Pepsi	16.0
4	5	11.5	Chetos	11.5

Adicionalmente se exploró otra operación de agrupamiento con la función .count(). Con este agrupamiento se pretende obtener el número de items vendidos por cada producto.

```
df_group_count =
df_extendido.groupby(by="PRODUCTO_ID").count().drop(["PRECIO","NOMBRE",
"PRECIO","FECHA"], 1)
df_group_count.rename(columns={'VENTA_ID':"COUNT"}, inplace=True)
display(df_group_count)
```

Como resultado se obtiene el numero de elementos vendidos, se ha aplicado una funcion para borrar las columnas que no hace sentido agruparlas y renombrar a un nombre de columna mas descriptivo.

PRODUCTO_ID	COUNT
1	2
2	2
3	2
4	1
5	1

Para terminar se hizo un join de los datos obtenidos.

```
pd.merge(reporte,df_group_count, how='left', on="PRODUCTO_ID" )
```

Como resultado se obtiene un DataFrame com las columna “PRODUCTO_ID” que indica el identificador único de producto, “PRECIO_X” que contiene la suma de precio de todos los productos vendidos de una misma categoria, en NOMBRE se describe el tipo de producto asociado al id, en “PRECIO_y” encontramos el costo unitario de cada producto y en “COUNT” se muestra el número de productos vendidos por cada tipo de producto.

	PRODUCTO_ID	PRECIO_x	NOMBRE	PRECIO_y	COUNT
0	1	35.0	Coca Cola	17.5	2
1	2	37.8	Galletas Marías	18.9	2
2	3	42.6	Gansito	21.3	2
3	4	16.0	Pepsi	16.0	1
4	5	11.5	Chetos	11.5	1

CONCLUSIONES

A partir de dos tablas distintas, de productos y de ventas se hizo un cruce de información mediante funciones de pandas, esto nos permitió tener una tabla más completa que dio lugar a funciones de agrupamiento para poder presentar un resumen con datos más amplios. Este tipo de cruces se muestra útil cuando tenemos una lista de precios de productos, y varias sucursales que tienen distintos números de ventas, siendo útil entonces para presentar informes de ventas en una sola tabla.La parte más compleja de la práctica

es tener conocimientos previos sobre el cruce de información, entender el tipo de joins y tener muy claros los pasos que debemos seguir para obtener las columnas deseadas, asi mismo tener presente las funciones de agrupamiento y agregación para pode representar estadísticos más interesantes.