



Data Science

Parte 1: Fundamentos de programación con Python

Nombre

Espinosa Ávila Ana Yessica

Proyecto 1

INTRODUCCIÓN A PYTHON

Viernes 11 de febrero de 2022

Contenido

1. INTRODUCCIÓN	3
2. DEFINICIÓN DE CÓDIGO	3
2.1. PRODUCTOS MÁS VENDIDOS Y PRODUCTOS REZAGADOS	5
Generar un listado de los 5 productos con mayores ventas y uno con los 10 productos con mayor número de búsquedas	5
Por categoría, generar un listado con los 5 productos con menores ventas y uno con los 10 productos con menores búsquedas	6
2.2. PRODUCTOS POR RESEÑA EN EL SERVICIO	7
Mostrar dos listados de 5 productos cada uno, un listado para productos con las mejores reseñas y otro para las peores, considerando los productos con devolución (no considerar productos sin reseñas)	8
2.3. TOTAL DE INGRESOS Y VENTAS PROMEDIO MENSUALES, TOTAL ANUAL Y MESES CON MÁS VENTAS AL AÑO	8
2.4. ENLACE A REPOSITORIO GITHUB	11
3. SOLUCIÓN AL PROBLEMA	12
4. CONCLUSIONES	12

PROYECTO 1. INTRODUCCIÓN A PYTHON

1. INTRODUCCIÓN

LifeStore es una tienda virtual que maneja una amplia gama de artículos, recientemente, la Gerencia de ventas, se percató que la empresa tiene una importante acumulación de inventario. Asimismo, se ha identificado una reducción en las búsquedas de un grupo importante de productos, lo que ha redundado en una disminución sustancial de sus ventas del último trimestre.

Para realizar el análisis se utilizará el archivo `lifestore-file.py`, el cual contiene los registros de las compras, búsquedas y productos manejados por la tienda. El análisis considerará el desarrollo de un sistema de análisis, en el que mediante autenticación de usuario-administrador se muestra un reporte mensual que especifica los puntos señalados en la consigna.

2. DEFINICIÓN DE CÓDIGO

A continuación, se describe el funcionamiento del código

Primeramente, se importan las librerías de pandas y de numpy, además del archivo de datos

```
✓ [4] # Importamos librerías
import pandas as pd
import numpy as np
# Importamos archivo de datos
from lifestore_file import lifestore_products, lifestore_sales, lifestore_searches
```

Para la preparación del conjunto de datos de ventas, convertimos los datos a matriz mediante la función `array` de numpy, se nombran las columnas, y las columnas de

id_sale, id_product, score, refund se convierten a formato entero y date a formato de fecha, como se muestra a continuación:

```
5 # Dataframe de ventas
matriz_sales = np.array(lifestore_sales)
df_sales = pd.DataFrame(matriz_sales, columns=['id_sale','id_product','score','date','refund'])
# lifestore_sales = [id_sale, id_product, score (from 1 to 5), date, refund (1 for true or 0 to false)] Refund = reembolso
df_sales["id_sale"] = pd.to_numeric(df_sales["id_sale"])
df_sales["id_product"] = pd.to_numeric(df_sales["id_product"])
df_sales["score"] = pd.to_numeric(df_sales["score"])
df_sales["refund"] = pd.to_numeric(df_sales["refund"])
df_sales["date"] = pd.to_datetime(df_sales["date"])
df_sales.head()
```

Obtenemos la siguiente salida:

```
> id_sale id_product score date refund
0      1          1      5 2020-07-24      0
1      2          1      5 2020-07-27      0
2      3          2      5 2020-02-24      0
3      4          2      5 2020-05-22      0
4      5          2      5 2020-01-01      0
```

Se realiza un proceso similar para el conjunto de datos de productos, se convierte a matriz, después a conjunto de datos de pandas y posterior a ello se convierten las columnas correspondientes a formato número:

```
[ ] # Dataframe de productos
matriz_products = np.array(lifestore_products)
df_products = pd.DataFrame(matriz_products, columns=['id_product','name','price','category','stock'])
df_products["id_product"] = pd.to_numeric(df_products["id_product"])
df_products["price"] = pd.to_numeric(df_products["price"])
df_products["stock"] = pd.to_numeric(df_products["stock"])
df_products.head()
```

Para el conjunto de datos de búsquedas, el código es el siguiente:

```
# Dataframe de búsquedas
matriz_searches = np.array(lifestore_searches)
df_searches = pd.DataFrame(matriz_searches, columns=['id_search','id_product'])
# ---- lifestore_searches = [id_search, id product]
df_searches["id_search"] = pd.to_numeric(df_searches["id_search"])
df_searches["id_product"] = pd.to_numeric(df_searches["id_product"])
df_searches.head()
```

Se obtendrán ahora dos conjuntos de datos en donde se agrupará el total de ventas mediante groupby usando la opción count, lo cual nos regresará un conteo de ventas por producto, se eliminan las columnas no usadas y se renombra la

correspondiente, además mediante merge unimos los conjuntos de datos de productos y de ventas para obtener las descripciones correspondientes:

```
[ ] # Ventas
df_ventas = df_sales.groupby(by="id_product").count()
df_ventas = df_ventas.drop(['score', 'date', 'refund'], axis=1)
df_ventas = df_ventas.set_axis(['ventas'], axis=1)
df_ventas = pd.merge(df_products, df_ventas, on='id_product')

[ ] # Búsquedas
df_búsquedas = df_searches.groupby(by="id_product").count()
df_búsquedas = df_búsquedas.set_axis(['búsquedas'], axis=1)
df_búsquedas = pd.merge(df_products, df_búsquedas, on='id_product')
```

2.1. PRODUCTOS MÁS VENDIDOS Y PRODUCTOS REZAGADOS

Se obtiene a partir del análisis de las categorías con menores ventas y categorías con menores búsquedas

Generar un listado de los 5 productos con mayores ventas y uno con los 10 productos con mayor número de búsquedas

Para el listado de los 5 productos más vendidos tomamos el conjunto de datos previamente creado df_ventas, y lo ordenamos descendientemente y mostramos únicamente los primeros 5

```
[ ] # Productos más vendidos
df_masvendidos = df_ventas.sort_values(by='ventas', ascending=False).head(5)
df_masvendidos
```

	id_product	name	price	category	stock	ventas
32	54	SSD Kingston A400, 120GB, SATA III, 2.5", 7mm	259	discos duros	300	50
2	3	Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, S...	3089	procesadores	987	42
4	5	Procesador Intel Core i3-9100F, S-1151, 3.60GH...	1779	procesadores	130	20
22	42	Tarjeta Madre ASRock Micro ATX B450M Steel Leg...	1779	tarjetas madre	0	18
33	57	SSD Adata Ultimate SU800, 256GB, SATA III, 2.5...	889	discos duros	15	15

Para los 10 productos con mayor número de búsquedas, ordenamos df_búsquedas en orden descendente y mostramos los primeros 10

```
[ ] # Productos con mayor número de búsquedas
df_masbuscados = df_busquedasc.sort_values(by='busquedas', ascending=False).head(10)
df_masbuscados
```

	id_product	name	price	category	stock	busquedas
37	54	SSD Kingston A400, 120GB, SATA III, 2.5", 7mm	259	discos duros	300	263
39	57	SSD Adata Ultimate SU800, 256GB, SATA III, 2.5...	889	discos duros	15	107
22	29	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GA...	2499	tarjetas madre	10	60
2	3	Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, S...	3089	procesadores	987	55
3	4	Procesador AMD Ryzen 3 3200G con Gráficos Rade...	2209	procesadores	295	41
50	85	Logitech Audífonos Gamer G635 7.1, Alámbrico, ...	2159	audífonos	39	35
43	67	TV Monitor LED 24TL520S-PU 24, HD, Widescreen,...	3229	pantallas	411	32
6	7	Procesador Intel Core i7-9700K, S-1151, 3.60GH...	8559	procesadores	114	31
4	5	Procesador Intel Core i3-9100F, S-1151, 3.60GH...	1779	procesadores	130	30
31	47	SSD XPG SX8200 Pro, 256GB, PCI Express, M.2	1209	discos duros	8	30

Por categoría, generar un listado con los 5 productos con menores ventas y uno con los 10 productos con menores búsquedas

Para obtener el listado con los 5 productos con menores ventas, tomamos el mismo conjunto de datos df_ventasc pero en esta ocasión el parámetro ascendente lo indicamos en verdadero y con head indicamos que muestre los primeros 5

```
# Productos con menores ventas
df_menosvendidos = df_ventasc.sort_values(by='ventas', ascending=True).head(5)
df_menosvendidos
```

	id_product	name	price	category	stock	ventas
41	94	HyperX Audífonos Gamer Cloud Flight para PC/PS...	2869	audífonos	12	1
15	22	Tarjeta de Video MSI NVIDIA GeForce GTX 1050 T...	3429	tarjetas de video	0	1
21	40	Tarjeta Madre Gigabyte XL-ATX TRX40 Designare,...	17439	tarjetas madre	1	1
24	45	Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151...	2869	tarjetas madre	25	1
12	17	Tarjeta de Video Gigabyte AMD Radeon R7 370 OC...	4199	tarjetas de video	1	1

Para los productos con menor número de búsquedas lo realizamos de manera similar tomando df_busquedasc, ordenándolo ascendentemente mostrando los primeros 10 resultados

```
[ ] # Productos con menor número de búsquedas
df_menosbuscados = df_busquedasc.sort_values(by='busquedas', ascending=True).head(10)
df_menosbuscados
```

	id_product	name	price	category	stock	busquedas
44	70	Samsung Smart TV LED 43, Full HD, Widescreen, ...	7679	pantallas	10	1
53	93	Ginga Audífonos con Micrófono GI18ADJ01BT-RO, ...	160	audífonos	139	1
24	35	Tarjeta Madre Gigabyte micro ATX Z390 M GAMING...	3419	tarjetas madre	30	1
48	80	Ghia Bocina Portátil BX800, Bluetooth, Inalámb...	1359	bocinas	15	1
8	9	Procesador Intel Core i3-8100, S-1151, 3.60GHz...	2549	procesadores	35	1
9	10	MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PC...	889	tarjetas de video	13	1
20	27	Tarjeta de Video VisionTek AMD Radeon HD5450, ...	2109	tarjetas de video	43	1
40	59	SSD Samsung 860 EVO, 1TB, SATA III, M.2	5539	discos duros	10	1
29	45	Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151...	2869	tarjetas madre	25	1
52	91	Genius GHP-400S Audífonos, Alámbrico, 1.5 Metr...	137	audífonos	16	2

2.2. PRODUCTOS POR RESEÑA EN EL SERVICIO

Se obtiene a partir del análisis de categorías con mayores ventas y categorías con mayores búsquedas.

Para este punto realizamos otro conjunto de datos para reseñas, el cual se muestra a continuación:

```
# Reseñas
df_resenas = df_sales.groupby(by=["id_product", "score"], as_index=False).count()
df_resenas = df_resenas.set_axis(['id_product', 'score', 'resenas', 'calificacion', 'promedio'], axis=1)
df_resenas["id_product"] = pd.to_numeric(df_resenas["id_product"])
df_resenas["score"] = pd.to_numeric(df_resenas["score"])
df_resenas["calificacion"] = df_resenas["resenas"]*df_resenas["score"]
df_resenas = df_resenas.groupby(by="id_product").sum()
df_resenas["promedio"] = df_resenas["calificacion"]/df_resenas["resenas"]
df_resenas = pd.merge(df_products, df_resenas, on='id_product')
df_resenas = df_resenas.filter(['id_product', 'name', 'promedio'])
```

Para ello lo que realizamos fue primero tomar el conjunto de datos y agruparlo por id_product y score, índice en False, después renombramos columnas, y cambiamos las columnas correspondientes a tipo numérico. A continuación en la columna calificación asignamos el resultado de la multiplicación de reseñas por score, agrupamos la suma por id_product y lo dividimos entre el total de número de reseñas. Y finalmente hacemos un join entre df_products y df_resenas tomando como pivote id_product. Para mostrar, seleccionamos las columnas id_product, name y promedio.

Mostrar dos listados de 5 productos cada uno, un listado para productos con las mejores reseñas y otro para las peores, considerando los productos con devolución (no considerar productos sin reseñas)

Se muestran los 5 productos con mejores reseñas ordenando df_resenas descendientemente y mostrando los primeros 5.

```
# Productos con mejores reseñas 5
df_mejorescore = df_resenas.sort_values(by='promedio', ascending=False).head(5)
df_mejorescore
```

	id_product	name	promedio
0	1	Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Q...	5.0
14	21	Tarjeta de Video MSI AMD Mech Radeon RX 5500 X...	5.0
39	85	Logitech Audífonos Gamer G635 7.1, Alámbrico, ...	5.0
38	84	Logitech Audífonos Gamer G332, Alámbrico, 2 Me...	5.0
36	67	TV Monitor LED 24TL520S-PU 24, HD, Widescreen,...	5.0

Y para los productos con las peores reseñas realizamos un proceso similar, pero ordenando ascendientemente.

```
[ ] # Productos con peores reseñas
df_peorscore = df_resenas.sort_values(by='promedio', ascending=True).head(5)
df_peorscore
```

	id_product	name	promedio
24	45	Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151...	1.000000
12	17	Tarjeta de Video Gigabyte AMD Radeon R7 370 OC...	1.000000
19	31	Tarjeta Madre AORUS micro ATX B450 AORUS M (re...	1.833333
25	46	Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2,...	2.000000
40	89	Cougar Audífonos Gamer Phontum Essential, Alám...	3.000000

2.3. TOTAL DE INGRESOS Y VENTAS PROMEDIO MENSUALES, TOTAL ANUAL Y MESES CON MÁS VENTAS AL AÑO

Para obtener el total de ingresos, tomamos mes y año de la columna que previamente asignamos de tipo fechay seleccionamos únicamente los que tengan en su columna refund valor cero, es decir, los que no hayan tenido devoluciones, y realizamos join para obtener el nombre del producto.


```
[ ] # Ingresos (quitando devoluciones)
df_sale = df_sales
df_sale['mes'] = df_sale['date'].dt.strftime('%m')
df_sale['anio'] = df_sale['date'].dt.strftime('%Y')
df_sale = df_sale[df_sale['refund'] == 0]
df_sale = pd.merge(df_sale, df_products, on='id_product')
```

El total de ingresos anuales lo obtenemos con la suma

```
[ ] # Total de ingresos anuales
df_ingresos = df_sale.groupby(by=['anio']).sum()
df_ingresos.filter(['price'])
```

	price
anio	
2020	737916

El total de ventas con conteo

```
[ ] # Total de ventas anuales
df_ingresos = df_sale.groupby(by=['anio']).count()
df_ingresos.filter(['price'])
```

	price
anio	
2020	274

El total de ingresos mensuales lo agrupamos la suma por mes

```
[ ] # Total de ingresos mensuales
df_ingresos = df_sale.groupby(by=['mes']).sum()
df_ingresos.filter(['price'])
```

price	
mes	
01	72983
02	107674
03	131649
04	169578
05	69071
06	57314
07	38037
08	7655
09	36916
10	16361
11	21021
12	9657

Para ventas mensuales se agrupa el conteo por mes

```
[ ] # Total de ventas mensuales
df_ingresos = df_sale.groupby(by=['mes']).count()
df_ingresos.filter(['price'])
```

price	
mes	
01	37
02	36
03	41
04	72
05	19
06	16
07	13
08	5
09	14
10	9
11	9
12	3

Meses con más ventas

```
[ ] # Meses con más ventas
df_ingresos = df_sale.groupby(by=['mes']).sum()
df_ingresos = df_ingresos.sort_values(by='price', ascending=False)
df_ingresos.filter(['price'])
```

	price
mes	
04	169578
03	131649
02	107674
01	72983
05	69071
06	57314
07	38037
09	36916
11	21021
10	16361
12	9657
08	7655

Autenticación

```
[ ] # Autenticación
usuario = 'yess'
contra = 'yess'
intentos = 0
acceso = False

while not acceso:
    intentos +=1
    if intentos ==4:
        exit()
    if input('Usuario: ') == usuario and input('Contraseña: ') == contra:
        acceso = True
        print('Acceso autorizado')
    else:
        print(f'Tienes {3 - intentos} intentos restantes')
```

Usuario: yess
 Contraseña: yess
 Acceso autorizado

2.4. ENLACE A REPOSITORIO GITHUB



Enlace a repositorio GitHub:

<https://github.com/yessmetal/EMTECH-PROYECTO-01>

3. SOLUCIÓN AL PROBLEMA

Se sugiere tomar en cuenta los productos más vendidos para analizar si se cuenta con el suficiente stock, por mencionar un ejemplo el producto más vendido es SSD Kingston A400 de 120Gb, DTA III, 2.5", este tuvo 50 ventas, pero no se considera necesario tener un mayor número de unidades a corto plazo ya que se cuenta con un stock de 300 unidades, se requeriría surtir en mediano plazo, en cambio la Tarjeta Madre ASRock Micro ATX B450M tuvo ventas altas de 18 unidades pero en stock hay cero en existencia.

En cuanto al número de búsquedas, convendría revisar qué tantos de los productos buscados son vendidos, lo que puede observarse es que la categoría de discos duros, tarjeta madre, procesadores, figuran entre las más buscadas. En cuanto a los productos con menos ventas puede verse que se tiene la Tarjeta Madre ASRock ATX H110, su precio es elevado, es de 2,869 y se tienen 25 en existencia, se sugeriría ofrecer un descuento o promoción para así liberar stock e incrementar número de ventas.

4. CONCLUSIONES

Como se pudo observar en el análisis de los datos con los que contamos, se puede realizar una buena estrategia de venta, ya que contamos con los datos de cuántos productos se han vendido, este podría ser de nuestros principales indicadores para proceder a conocer qué calificaciones tienen los productos más vendidos y si algunos de ellos han tenido devoluciones y procurar mantener estos productos en stock. En cuanto a búsquedas, podemos realizar publicidad o accesos directos en productos que estén bien calificados, pero probablemente no están teniendo éxitos en búsquedas.