

PROYECTO 1

INTRODUCCIÓN A PYTHON

EMMANUEL ACERO CASILDO

INDICE

INTRODUCCIÓN	2
DEFINICIÓN DE CÓDIGO	2
SOLUCIÓN AL PROBLEMA.....	10
CONCLUSIÓN	13

INTRODUCCIÓN

En el presente reporte se redacta a detalle el camino seguido y los resultados del proceso de análisis final que en el que se trabajó , el cual contempla una interfaz de usuario en consola con navegación intuitiva que muestra la información solicitada por la tienda virtual “**LifeStore**”, en relación a aspectos importantes de sus productos y ventas.

DEFINICIÓN DE CÓDIGO

Para realizar los procesos analíticos solicitados por “**LifeStore**” a partir de las listas predefinidas con la información a procesar, se optó por usar la librería **pandas**, la cual facilita en enorme medida el manejo de datos a través de tablas llamadas **DataFrames** de una manera eficiente y con poco código, por lo que nuestro código al evitar usar ciclos “**for**” aumenta en gran magnitud su rapidez por lo que la entrega de resultados será casi inmediata y en un formato amigable.

Aquí se muestran las librerías utilizadas en el desarrollo.

```
1 import pandas as pd
2 from time import sleep
3 from os import system, name
4 from datetime import datetime
5
6 from lifestore file import *
```

Al inicio del desarrollo se definen variables globales útiles para guardar las credenciales de acceso disponibles y estados del proceso principal como se muestra a continuación:

```
# Diccionario que contiene los usuarios disponibles y sus respectivas contraseñas
# {'usuario':'contraseña'}
user_credentials = {'admin':'adminpass', 'client':'clientpass', 'general_user':'genuserpass'}

alive = True # Valor bandera para controlar el cierre del ciclo while del programa
active_session = False # Nos indica si una sesión esta activa o inactiva
user_on_session = None # Se almacena el usuario que ha iniciado sesión
pre_signin_menu_printed = False # Almacenamos si el menu pre inicio de sesión ha sido mostrado
post_signin_menu_printed = False # Almacenamos si el menu post inicio de sesión ha sido mostrado
```

En seguida, justo antes de iniciar el proceso principal de nuestra interfaz, se comienza con el preprocesamiento de los datos crudos disponibles.

Primero las listas de listas pasan a convertirse a DataFrames con la misma estructura de datos establecida. Para el DataFrame de productos, coloca el id del producto como el

índice del DataFrame y para el caso de la columna **'date'** DataFrame de ventas se pasa a ser un objeto **"np.datetime[ns]"** el cual nos va a servir de mucha utilidad para poder realizar filtros de fechas de una manera bastante sencilla.

```
18 # Conversión de lista de listas a DataFrames
19 df_lifystore_products = pd.DataFrame(lifystore_products, columns=['id_product', 'name', 'price', 'category', 'stock'])
20 df_lifystore_sales = pd.DataFrame(lifystore_sales, columns=['id_sale', 'id_product', 'score', 'date', 'refund'])
21 df_lifystore_searches = pd.DataFrame(lifystore_searches, columns=['id_search', 'id_product'])
22
23 df_lifystore_products.set_index('id_product', inplace=True) # Se configura la columna id_product como índice
24 df_lifystore_sales['date'] = pd.to_datetime(df_lifystore_sales['date'], format='%d/%m/%Y') # Se transforma la columna 'date'
25 # en formato 'datetime'
```

En seguida, se obtienen la información estadística de interés de las ventas y búsquedas. Esto se realiza a través del método **'groupby'**, el cual es muy útil cuando se requiere hacer agrupaciones de datos y aplicar agregaciones como el promedio o la suma con la finalidad de obtener datos estadísticos de interés de esa agrupación.

```
26 # Obtención de estadísticas de ventas
27 sales_statistics = df_lifystore_sales.groupby(by='id_product').agg({'id_product': ['count'],
28                             'refund': ['sum'],
29                             'score': ['mean']})
30 sales_statistics.columns = sales_statistics.columns.map('_'.join)
31 sales_statistics = sales_statistics.rename(columns={"id_product_count": "total_sales",
32                                                  "refund_sum": "total_refunds"}) # Se renombran columnas a nombres
33 # más significativos
34 sales_statistics['score_mean'] = sales_statistics['score_mean'].round(1) # Se redondea el score promedio a una decima
35
36 # Obtención de estadísticas de búsquedas
37 searches_statistics = df_lifystore_searches.groupby(by='id_product').agg({'id_product': ['count']})
38 searches_statistics.columns = searches_statistics.columns.map('_'.join)
39 searches_statistics = searches_statistics.rename(columns={"id_product_count": "total_searches"}) # Se renombran columnas a nombres
40 # más significativos
```

Una vez obtenida la información estadística anterior, se une al DataFrame de productos por medio del método **join**, el cual une dos DataFrames a partir de información en común, en este caso, el **id** del producto. Al realizar el **join**, tendremos algunos registros con valor **NaN**, que son valores nulos. Estos pasan a ser ceros con el método **"fillna"** y se reformatean a unas columnas tipo entero ya que al realizar el **join**, estás columnas pasaron a ser formato float.

```
36 # Obtención de estadísticas de búsquedas
37 searches_statistics = df_lifystore_searches.groupby(by='id_product').agg({'id_product': ['count']})
38 searches_statistics.columns = searches_statistics.columns.map('_'.join)
39 searches_statistics = searches_statistics.rename(columns={"id_product_count": "total_searches"}) # Se renombran columnas a nombres
40 # más significativos
41
42 # Unión de estadísticas a dataframe de productos
43 df_lifystore_products = df_lifystore_products.join(sales_statistics).join(searches_statistics) # Se realiza un join con los estadísticos
44 # de ventas y búsquedas.
45 df_lifystore_products[['total_sales', 'total_refunds', 'total_searches']] = \
46     df_lifystore_products[['total_sales', 'total_refunds', 'total_searches']].fillna(0).astype(int) # Los valores NaN pasan ser 0 y
47 # se convierte a entero los registros
48 # de las columnas correspondientes
```

Finalmente, se obtienen los estadísticos de ventas mensuales. Para relacionar el **id** del producto con su correspondiente precio se utilizó el método **"merge"** que funciona igual que el método **"join"**. Así, se une la columna de precios del DataFrame de productos con el de ventas. Por consiguiente, mediante **"group_by"** se obtienen los estadísticos.

```

50 # Obtención de estadísticos de ventas mensuales
51 start_date = datetime(2020, 1, 1) # Fecha a partir de la cual se tomará información
52 df_merged = pd.merge(df_lifstore_sales, # Se realiza un merge para unir el precio correspondiente
53                      df_lifstore_products['price'], # al id del producto
54                      left_on='id_product',
55                      right_index=True)
56 df_merged = df_merged.loc[df_merged['refund']==0] # Se filtran solo los registros que no hayan tenido devoluciones
57
58 month_statistics = df_merged.groupby(pd.Grouper(key="date", freq='M')).agg({'id_product': ['count'],
59                                                                           'price': ['sum']})
60 month_statistics.columns = month_statistics.columns.map('_'.join)
61 month_statistics = month_statistics.rename(columns={"id_product_count": "sales",
62                                                  "price_sum": "income"})
63
64 month_statistics = month_statistics.loc[~(month_statistics==0).all(axis=1)] # Se filtran solo los meses que tengan información
65 month_statistics = month_statistics.loc[month_statistics.index > start_date] # Se filtra la información a partir de 'start_date'

```

También se declara la función “**clear()**” que nos permitirá eliminar información de consola y así evitar una acumulación constante de información.

```

67 # Función que limpia la consola
68 def clear():
69     # Para windows
70     if name == 'nt':
71         system('cls')
72     # Para mac y linux (aquí, os.name es 'posix')
73     else:
74         system('clear')

```

Nuestro proceso principal empieza a partir de la sección “**if __name__ == __main__**”. Básicamente nuestra interfaz vive dentro de un ciclo while, en donde la primera condición que se evalúa es si el usuario ha iniciado sesión, por lo que siempre que se inicie el programa se deberá iniciar sesión antes de poder acceder a los datos requeridos y para ello se desplegará el primer menú que tiene tanto la opción de inicio de sesión como la de salir del programa. La opción a acceder debe de ser la correcta, si no es así se mostrará un mensaje de error.

```

82 while alive:
83
84     attemps = 0 # Contador para almacenar el número de intentos de inicio de sesión
85     attemps_limit = 3 # Limite de intentos para inicio de sesión
86
87     if not active_session:
88
89         if not pre_signin_menu_printed:
90             print("\t¡Bienvenido a Podruct Analysis Tool!\n")
91             print("Ingresa el número de la opción deseada:\n")
92             print("1) INICIAR SESION")
93             print("2) SALIR\n")
94             pre_signin_menu_printed = True
95
96     option_selected = input("-> ") # Almacena la opción del menú seleccionada

```

```

¡Bienvenido a Podruct Analysis Tool!

Ingresa el número de la opción deseada:

1) INICIAR SESION
2) SALIR

-> 

```

```

¡Bienvenido a Podruct Analysis Tool!

Ingresa el número de la opción deseada:

1) INICIAR SESION
2) SALIR

-> a

La opción ingresada no es válida.
Ingresa una opción válida...

-> 

```

```

Ingresa tus credenciales:

-> USER: a
-> PASSWORD: a

El usuario o la contraseña son invalidos... :(
Ingresa tu credenciales nuevamente:

-> USER: a
-> PASSWORD: a

El usuario o la contraseña son invalidos... :(
Ingresa tu credenciales nuevamente:

-> USER: a
-> PASSWORD: a

Numero de intentos límite alcanzado ...
Volviendo al menu principal ^.^

```

Para iniciar sesión, se tendrán tres intentos. Sí, no se logró un inicio de sesión correcto aparecerá un mensaje de error y se regresará al menú principal.

Los usuarios disponibles y sus contraseñas están definidos en esta variable global que se asigna antes de iniciar el ciclo principal.

```

8 # Diccionario que contiene los usuarios disponibles y sus respectivas contraseñas
9 # {'usuario': 'contraseña'}
10 user_credentials = {'admin': 'adminpass',
11                     'client': 'clientpass',
12                     'general_user': 'genuserpass'}

```

En esta sección del código se muestra el proceso lógico que se sigue para determinar un inicio de sesión correcto.

```

100     try:
101         if int(option_selected) == 1:
102             clear()
103             print("\nIngresa tus credenciales:\n")
104
105             # Se arranca un subciclo while para el inicio de sesión
106             while attemps < attemps_limit:
107                 user = input("-> USER: ") # Se captura el nombre del usuario
108                 password = input("-> PASSWORD: ") # Se captura la contraseña del nombre del usuario ingresado
109
110                 if user in user_credentials and user_credentials[user] == password:
111                     active_session = True # Se actualiza ante un inicio de sesión exitoso
112                     user_on_session = user
113                     clear()
114                     break # Se detiene el subciclo while para el inicio de sesión
115                 else:
116                     # Ante inicio de sesión fallidos se acumulan los intentos evaluando si se ha llegado
117                     # a un límite
118                     if attemps < attemps_limit - 1:
119                         print("\nEl usuario o la contraseña son invalidos... :('")
120                         print("Ingresa tu credenciales nuevamente:\n")
121                         attemps += 1
122                     else:
123                         print("\nNumero de intentos límite alcanzado ...")
124                         print("Volviendo al menu principal ^.^")
125                         sleep(2)
126                         attemps += 1
127                         pre_signin_menu_printed = False
128                         clear()
129
130             elif int(option_selected) == 2:
131                 print("\nGracias por tu visita, esperamos verte pronto :) ...")
132                 alive = False
133
134     except ValueError:
135         print("\nLa opción ingresada no es valida.")
136         print("Ingresa una opción valida...\n")

```

Una vez se haya concretado un inicio de sesión, se podrá acceder al menú donde se ejecutará la visualización de la información requerida y también se tendrá la opción ya sea de cerrar sesión y volver al menú de inicio de sesión o de terminar por completo el programa. Además, si no se coloca una opción disponible aparecerá el mismo mensaje que se presentó en el menú anterior.

- 1) PRODUCTOS CON MÁS VENTAS
- 2) PRODUCTOS CON MÁS BUSQUEDAS
- 3) PRODUCTOS CON MENOS VENTAS
- 4) PRODUCTOS CON MENOS BUSQUEDAS
- 5) PRODUCTOS REZAGADOS
- 6) PRODUCTOS CON BUENAS RESEÑAS
- 7) PRODUCTOS CON MALAS RESEÑAS
- 8) VENTAS MENSUALES
- 9) Cerrar sesion
- 10) Salir

Por lo tanto, a continuación se muestra cada sección del código que obtiene los datos a visualizar para cada opción con su respectivo resultado en consola.

```
# Se despliegan los 10 productos con más ventas
if int(option_selected) == 1:
    clear()
    top_sales = df_lifestore_products[['name',
                                       'category',
                                       'total_sales']].nlargest(10, 'total_sales')
    print(f"\n\t\t\t\t\t ***** PRODUCTOS CON MAYORES VENTAS *****\n")
    print(top_sales.to_markdown())
    input("\nPresiona enter para volver al menu principal :)")
    post_signin_menu_printed = False
    clear()
```

***** PRODUCTOS CON MAYORES VENTAS *****			
id_product	name	category	total_sales
54	SSD Kingston A400, 120GB, SATA III, 2.5", 7mm	discos duros	50
3	Procesador AMD Ryzen 5 2600, S-AM4, 3.60GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth	procesadores	42
5	Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)	procesadores	20
42	Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD	tarjetas madre	18
57	SSD Adata Ultimate SU800, 256GB, SATA III, 2.5", 7mm	discos duros	15
29	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD	tarjetas madre	14
2	Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth	procesadores	13
4	Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire	procesadores	13
47	SSD XPG SX8200 Pro, 256GB, PCI Express, M.2	discos duros	11
12	Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC, 6GB 192-bit GDDR6, PCI Express x16 3.0	tarjetas de video	9

Presiona enter para volver al menu principal :_

- **OPCIÓN 2: PRODUCTOS CON MÁS BUSQUEDAS**

```
# Se despliegan los 10 productos con más búsquedas
elif int(option_selected) == 2:
    clear()
    top_searches = df_lifestore_products[['name',
                                           'category',
                                           'total_searches']].nlargest(10, 'total_searches')
    print(f"\n\t\t\t\t\t\t\t\t\t*****\n\t\t\t\t\t\t\t\t\tPRODUCTOS CON MAS BUSQUEDAS *****\n")
    print(top_searches.to_markdown())
    input("\nPresiona enter para volver al menu principal :)")
    post_signin_menu_printed = False
    clear()
```

***** PRODUCTOS CON MAS BUSQUEDAS *****			
id_product	name	category	total_searches
54	SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm	discos duros	263
57	SSD Adata Ultimate 9UB80, 256GB, SATA III, 2.5'', 7mm	discos duros	107
29	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD	tarjetas madre	60
3	Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth	procesadores	55
4	Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire	procesadores	41
85	Logitech Audifonos Gamer G635 7.1, Alámbrico, 1.5 Metros, 3.5mm, Negro/Azul	audifonos	35
67	TV Monitor LED 24T1520S-PU 24, HD, Widescreen, HDMI, Negro	pantallas	32
7	Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)	procesadores	31
5	Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)	procesadores	30
47	SSD XPG SX8200 Pro, 256GB, PCI Express, M.2	discos duros	30

Presiona enter para volver al menu principal :)

- **OPCIÓN 3: PRODUCTOS CON MENOS VENTAS**

```
# Se despliegan los 10 productos con menos ventas
elif int(option_selected) == 3:
    clear()
    # Se eliminan las filas que contengan ceros en la columna 'total_sales'
    df_non_zero_sales = df_lifestore_products.loc[~(df_lifestore_products['total_sales']==0)]
    bottom_sales = df_non_zero_sales[['name',
                                       'category',
                                       'total_sales']].nsmallest(10, 'total_sales')
    print(f"\n\t\t\t\t\t ***** PRODUCTOS CON MENORES VENTAS *****\n")
    print(bottom_sales.to_markdown())
    input("\nPresiona enter para volver al menu principal :)")
    post_signin_menu_printed = False
    clear()
```

```

***** PRODUCTOS CON MENORES VENTAS *****
id_product | name | category | total_sales |
-----
10 | MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PCI Express 2.0 | tarjetas de video | 1 |
13 | Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix, 4GB 128-bit GDDR5, PCI Express 3.0 | tarjetas de video | 1 |
17 | Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2GB 256-bit GDDR5, PCI Express 3.0 | tarjetas de video | 1 |
22 | Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC, 4GB 128-bit GDDR5, PCI Express x16 3.0 | tarjetas de video | 1 |
28 | Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti, 6GB 192-bit GDDR6, PCI Express x16 3.0 | tarjetas de video | 1 |
40 | Tarjeta Madre Gigabyte XL-ATX TRX400 Designare, S-STRX4, AMD TRX40, 256GB DDR4 para AMD | tarjetas madre | 1 |
45 | Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, Intel H110, 32GB DDR4, para Intel | tarjetas madre | 1 |
46 | Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2, S-1151, Intel H110, 32GB DDR4 para Intel | tarjetas madre | 1 |
50 | SSD Crucial MX500, 1TB, SATA III, M.2 | discos duros | 1 |
60 | Kit Memoria RAM Corsair Dominator Platinum DDR4, 3200MHz, 16GB (2x 8GB), Non-ECC, CL16, XMP | memorias usb | 1 |

Presiona enter para volver al menu principal :)

```

- **OPCIÓN 4: PRODUCTOS CON MENOS BUSQUEDAS**


```
# Se despliegan los 10 productos con menos búsquedas
elif int(option_selected) == 4:
    clear()
    # Se eliminan las filas que contengan ceros en la columna 'total_searches'
    df_non_zero_searches = df_lifestore_products.loc[~(df_lifestore_products['total_searches']==0)]
    bottom_searches = df_non_zero_searches[['name',
                                            'category',
                                            'total_searches']].nsmallest(10, 'total_searches')
    print(f"\n\t\t\t\t\t ***** PRODUCTOS CON MENORES BUSQUEDAS *****\n")
    print(bottom_searches.to_markdown())
    input("\nPresiona enter para volver al menu principal :)")
    post_signin_menu_printed = False
    clear()
```

***** PRODUCTOS CON MENORES BUSQUEDAS *****			
id_product	name	category	total_searches
9	Procesador Intel Core i3-8100, S-1151, 3.60GHz, Quad-Core, 6MB Smart Cache (8va. Generación - Coffee Lake)	procesadores	1
10	MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PCI Express 2.0	tarjetas de video	1
27	Tarjeta de Video VisionTek AMD Radeon HD5450, 2GB GDDR3, PCI Express x16	tarjetas de video	1
35	Tarjeta Madre Gigabyte micro ATX Z390 M GAMING, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel	tarjetas madre	1
45	Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, Intel H110, 32GB DDR4, para Intel	tarjetas madre	1
59	SSD Samsung 860 EVO, 1TB, SATA III, M.2	discos duros	1
70	Samsung Smart TV LED 43, Full HD, Widescreen, Negro	pantallas	1
80	Ghia Bocina Portátil BX800, Bluetooth, Inalámbrico, 2.1 Canales, 31W, USB, Negro	bocinas	1
93	Ginga Audifonos con Micrófono GI18ADQ31BT-RO, Bluetooth, Alámbrico/Inalámbrico, 3.5mm, Rojo	audifonos	1
13	Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix, 4GB 128-bit GDDR5, PCI Express 3.0	tarjetas de video	2

Presiona enter para volver al menu principal :)

- **OPCIÓN 5: PRODUCTOS REZAGADOS**

```
# Se despliegan todos los productos rezagados que tienen 0 ventas
elif int(option_selected) == 5:
    clear()
    # Se filtran aquellos productos que no han tenido ventas
    df_zero_sales = df_lifestore_products.loc[(df_lifestore_products[['total_sales']]==0).all(axis=1), \
        [name, category, stock, total_sales]]
    print(f"\n\t\t\t\t\t***** PRODUCTOS REZAGADOS *****\n")
    print(df_zero_sales.to_markdown())
    input("\nPresiona enter para volver al menu principal :)")
    post_signin_menu_printed = False
    clear()
```

***** PRODUCTOS REZAGADOS *****				
id_producto	name	category	stock	total_sales
9	Procesador Intel Core i3-8100, S-1151, 3.60GHz, Quad-Core, GMB Smart Cache (Bva. Generación - Coffee Lake)	procesadores	35	0
14	Tarjeta de Video EVGA NVIDIA GeForce GT 710, 2GB, 64-bit GDDR3, PCI Express 2.0	tarjetas de video	36	0
15	Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming, 6GB 192-bit GDDR6, PCI 3.0	tarjetas de video	15	0
16	Tarjeta de Video EVGA NVIDIA GeForce RTX 2060 SC Ultra Gaming, 6GB 192-bit GDDR6, PCI Express 3.0	tarjetas de video	10	0
19	Tarjeta de Video Gigabyte NVIDIA GeForce RTX 1650 OC Low Profile, 4GB 128-bit GDDR5, PCI Express 3.0 x16	tarjetas de video	8	0
20	Tarjeta de Video Gigabyte NVIDIA GeForce RTX 2060 SUPER WINDFORCE OC, 8 GB 256 bit GDDR6, PCI Express x16 3.0	tarjetas de video	10	0
23	Tarjeta de Video MSI Radeon X1550, 128MB 64 bit GDDR2, PCI Express x16	tarjetas de video	10	0
24	Tarjeta de Video PNY NVIDIA GeForce RTX 2080, 8GB 256-bit GDDR6, PCI Express 3.0	tarjetas de video	2	0
26	Tarjeta de Video Visiontek AMD Radeon HD 5450, 1GB DOR3, PCI Express x16 2.1	tarjetas de video	189	0
27	Tarjeta de Video Visiontek AMD Radeon HD5450, 2GB DOR3, PCI Express x16	tarjetas de video	10	0
30	Tarjeta Madre ASUS ATX Z390 ELITE, S-1151, Intel Z990, HDMI, 64GB DOR4 para Intel	tarjetas madre	50	0
32	Tarjeta Madre ASRock Z390 Phantom Gaming 4, S-1151, Intel Z390, HDMI, 64GB DOR4 para Intel	tarjetas madre	10	0
34	Tarjeta Madre ASUS ATX ROG STRIX B550-F GAMING Wi-Fi, S-AM4, AMD B550, HDMI, max. 128GB DOR4 para AMD	tarjetas madre	2	0
35	Tarjeta Madre Gigabyte micro ATX Z390 M GAMING, S-1151, Intel Z390, HDMI, 64GB DOR4 para Intel	tarjetas madre	30	0
36	Tarjeta Madre Gigabyte micro ATX Z490 GAMING X (rev. 1.0), Intel Z490, HDMI, 128GB DOR4 para Intel	tarjetas madre	10	0
37	Tarjeta Madre ASRock ATX Z490 STEEL LEGEND, S-1200, Intel Z490, HDMI, 128GB DOR4 para Intel	tarjetas madre	60	0
38	Tarjeta Madre Gigabyte Micro ATX H310M D3, S-1151, Intel H310, HDMI, 64GB DOR4 para Intel	tarjetas madre	15	0
39	ASUS T. Madre uATX M4A88T-M, S-AM3, DOR3 para Phenom II/Athlon II/Sempron 100	tarjetas madre	98	0
41	Tarjeta Madre ASUS micro ATX Prime H370M-Plus/CSM, S-1151, Intel H370, HDMI, 64GB DOR4 para Intel	tarjetas madre	286	0
43	Tarjeta Madre ASUS ATX ROG STRIX Z390-F GAMING, S-1151, Intel Z390, HDMI, 64GB DOR4 para Intel	tarjetas madre	5	0
53	SSD Addlink Technology S70, S12GB, PCI Express 3.0, M.2	discos duros	1	0
55	SSD para Servidor Supremacore SSD-DM128-SMCMW1, 128GB, SATA III, mSATA, Gbbit/s	discos duros	10	0
56	SSD para Servidor Lenovo Thinksystem S4500, 480GB, SATA III, 3.5", 7mm	discos duros	3	0
58	SSD para Servidor Lenovo Thinksystem S4510, 480GB, SATA III, 2.5", 7mm	discos duros	16	0
59	SSD Samsung 860 EVO, 1TB, SATA III, M.2	discos duros	10	0
61	Kit Memoria RAM Corsair Vengeance LPX DOR4, 2400MHz, 32GB, Negro-ECC, CL16	memorias usb	5	0
62	Makena Smart TV LED 3252 32", HD, Widescreen, Gris	pantallas	6	0
63	Seiki TV LED SC-39H5950N 38.5, HD, Widescreen, Negro	pantallas	146	0
64	Samsung TV LED LH430QREBGGXG 43, 4K Ultra HD, Widescreen, Negro	pantallas	71	0
65	Samsung Smart TV LED UN7000U1000FXZ 70, 4K Ultra HD, Widescreen, Negro	pantallas	7	0
66	Makena Smart TV LED 4052 40", Full HD, Widescreen, Negro	pantallas	230	0
69	Hisense Smart TV LED 40H5500F 39.5, Full HD, Widescreen, Negro	pantallas	94	0
70	Hisense Smart TV LED 43, Full HD, Widescreen, Negro	pantallas	10	0
71	Samsung Smart TV LED UN32J4290AF 32, HD, Widescreen, Negro	pantallas	3	0
72	Hisense Smart TV LED 50HF 49.5, 4K Ultra HD, Widescreen, Negro	pantallas	11	0
73	Samsung Smart TV LED UN55U7000FXZ 55, 4K Ultra HD, Widescreen, Negro/Gris	pantallas	4	0
75	Lenovo Barra de Sonido, Alámbrico, 2.5W, USB, Negro	bocinas	11	0
80	Acteck Bocina con Subwoofer XVS-200, Bluetooth, Inalámbrico, 2.1, 18W RMS, 180W PMPO, USB, Negro	bocinas	4	0
77	Verbatia Bocina Portátil Mini, Bluetooth, Inalámbrico, 3W RMS, USB, Blanco	bocinas	1	0
78	Ghia Bocina Portátil BX300, Bluetooth, Inalámbrico, 40W RMS, USB, Rojo - Resistente al Agua	bocinas	2	0
79	Naceb Bocina Portátil NA-B301, Bluetooth, Inalámbrico, USB 2.0, Rojo	bocinas	31	0
80	Ghia Bocina Portátil BX800, Bluetooth, Inalámbrico, 2.1 Canales, 31W, USB, Negro	bocinas	15	0
81	Ghia Bocina Portátil BX900, Bluetooth, Inalámbrico, 2.1 Canales, 34W, USB, Negro - Resistente al Agua	bocinas	20	0
82	Ghia Bocina Portátil DX400, Bluetooth, Inalámbrico, 34 RMS, USB, Negro	bocinas	31	0
83	Ghia Bocina Portátil BX500, Bluetooth, Inalámbrico, 3W RMS, USB, Gris	bocinas	16	0

- **OPCIÓN 6: PRODUCTOS CON BUENAS RESEÑAS**

```
# Se despliegan los productos con las mejores reseñas, en donde se calculo
# un promedio del score de las reseñas que ha tenido un producto
elif int(option_selected) == 6:
    clear()
    top_scores = df_lifystore_products[['name',
                                         'category',
                                         'total_refunds',
                                         'score_mean']].nlargest(10, 'score_mean')
    print(f"\n\t\t\t\t\t\t\t\t\t\t ***** PRODUCTOS CON BUENAS RESEÑAS *****\n")
    print(top_scores.to_markdown())
    input("\nPresiona enter para volver al menu principal :)")
    post_signin_menu_printed = False
    clear()
```

***** PRODUCTOS CON BUENAS RESERAS *****				
id_product	name	category	total_refunds	score_mean
1	Procesador AMD Ryzen 3 3300X 5-AM4, 3.80GHz, Quad-Core, 16MB L2 Cache	procesadores	0	5
6	Procesador Intel Core i9-9900K, 5-1151, 3.60GHz, 8-Core, 16MB Smart Cache (9na. Generación Coffee Lake)	procesadores	0	5
7	Procesador Intel Core i7-9700K, 5-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)	procesadores	0	5
8	Procesador Intel Core i5-9600K, 5-1151, 3.70GHz, Six-Core, 9MB Smart Cache (9na. Generación Coffee Lake)	procesadores	0	5
11	Tarjeta de Video ASUS AMD Radeon RX 570, 4GB 256-bit GDDR5, PCI Express 3.0	tarjetas de video	0	5
21	Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC, 8GB 128-bit GDDR6, PCI Express 4.0	tarjetas de video	0	5
22	Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC, 4GB 128-bit GDDR5, PCI Express x16 3.0	tarjetas de video	0	5
25	Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500 XT Gaming, 8GB 128-bit GDDR6, PCI Express 4.0	tarjetas de video	0	5
28	Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti, 6GB 192-bit GDDR6, PCI Express x16 3.0	tarjetas de video	0	5
40	Tarjeta Madre Gigabyte XL-ATX TRX40 Designare, 5-STRX4, AMD TRX40, 256GB DDR4 para AMD	tarjetas madre	0	5

Presiona enter para volver al menu principal :)

- **OPCIÓN 7: PRODUCTOS CON MALAS RESEÑAS**

```
# Se despliegan los productos con las peores reseñas, en donde se calculo  
# un promedio del score de las reseñas que ha tenido un producto  
elif int(option_selected) == 7:  
    clear()  
    bottom_scores = df_lifestore_products[['name',  
                                           'category',  
                                           'total_refunds',  
                                           'score_mean']].nsmallest(5, 'score_mean')  
    print(f"\n\t\t\t\t\t\t\t\t\t\t ***** PRODUCTOS CON MALAS RESEÑAS *****\n")  
    print(bottom_scores.to_markdown())  
    input("\nPresa enter para volver al menu principal :)")  
    post_signin_menu_printed = False  
    clear()
```

***** PRODUCTOS CON MALAS RESEÑAS *****				
id_product	name	category	total_refunds	score_mean
17	Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2GB 256-bit GDDR5, PCI Express 3.0	tarjetas de video	1	1
45	Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, Intel H110, 32GB DDR4, para Intel	tarjetas madre	1	1
31	Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD	tarjetas madre	3	1.8
46	Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2, S-1151, Intel H110, 32GB DDR4 para Intel	tarjetas madre	1	2
89	Cougar Audifonos Gamer Phantom Essential, Alámbrico, 1.9 Metros, 3.5mm, Negro.	audifonos	0	3

- **OPCIÓN 8: VENTAS MENSUALES**

```
# Se despliega una tabla con ventas mensuales del 2020, ya que solo de este año
# se cuenta con información concisa
elif int(option_selected) == 8:
    clear()
    print(f"\n***** VENTAS MENSUALES 2020 *****\n")
    print(month_statistics.to_markdown())
    input("\nPresiona enter para volver al menu principal :)")
    post_signin_menu_printed = False
    clear()
```

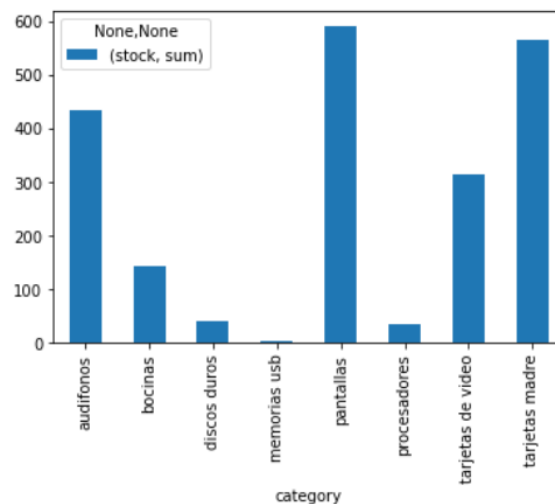
```
***** VENTAS MENSUALES 2020 *****

| date                | sales | income |
|:-----:|:-----:|:-----:|
| 2020-01-31 00:00:00 | 52    | 117738 |
| 2020-02-29 00:00:00 | 40    | 107270 |
| 2020-03-31 00:00:00 | 49    | 162931 |
| 2020-04-30 00:00:00 | 74    | 191066 |
| 2020-05-31 00:00:00 | 33    | 91677  |
| 2020-06-30 00:00:00 | 11    | 36949  |
| 2020-07-31 00:00:00 | 11    | 26949  |
| 2020-08-31 00:00:00 | 3     | 3077   |

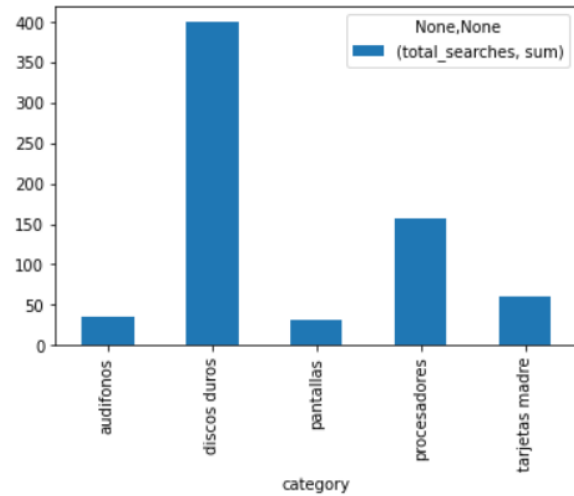
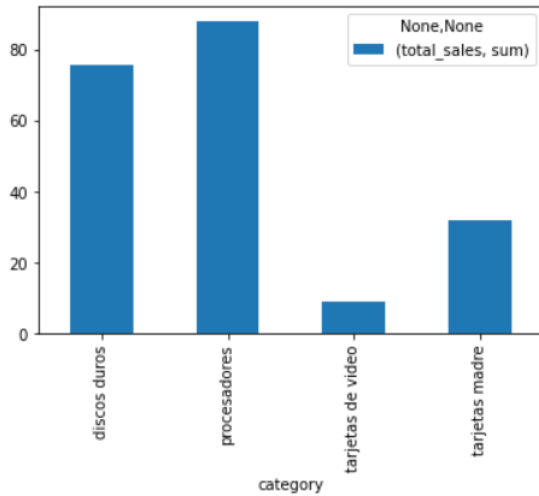
Presiona enter para volver al menu principal :)
```

SOLUCIÓN AL PROBLEMA

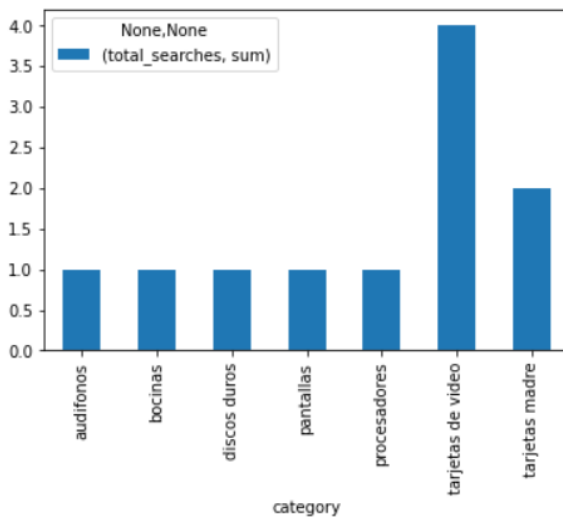
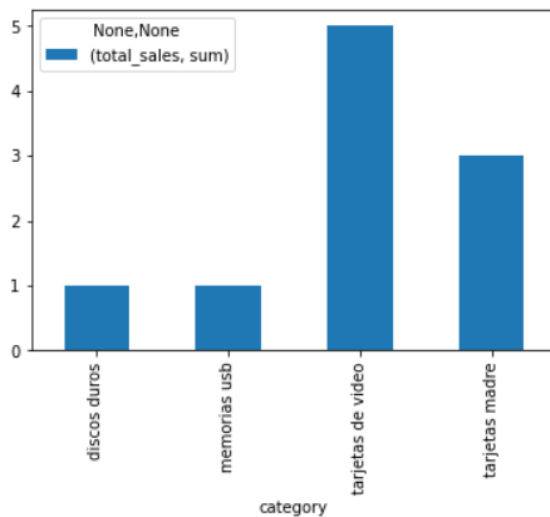
En base a los resultados arrojados por nuestro proceso anterior, podemos determinar que existe una gran cantidad de productos rezagados de los que no se ha registrado venta alguna y se puede apreciar que algunos de estos aún tienen una cantidad considerable de stock. Observando de una forma más general los productos rezagados en base a categoría y stock de estos productos nos resulta la siguiente gráfica.



Teniendo en cuenta las categorías con mayores ventas y mayores búsquedas en los siguientes gráficos.



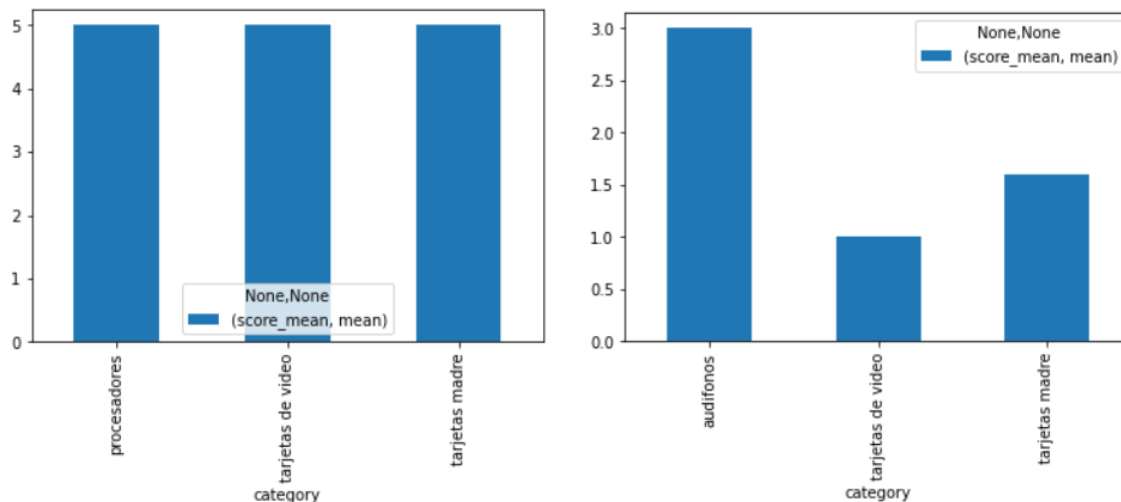
Así como las categorías de menores ventas y menores búsquedas en los siguientes gráficos.



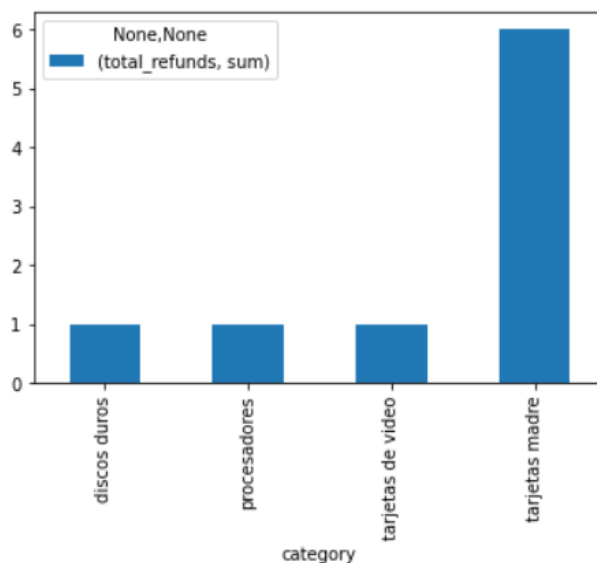
Podemos concluir que los productos a retirar definitivamente del mercado son las siguientes categorías: **bocinas, audifonos, memorias usb y pantallas**.

Por otra parte, se tendrá que prestar especial atención a los productos con categoría **tarjetas madres** y **tarjetas de video** que aunque tenemos que los productos con estas categorías tienen un lugar tanto en el top de ventas como en el de menores ventas y menores búsquedas, es muy probable que ciertos productos de cierto modelo se vendan más que otros.

Lo podemos ver en las siguientes graficas que muestran las categorías con mejores reseñas y las de peores reseñas, en donde nuevamente de ambos lados tenemos ambas categorías.



Incluso nos atrevemos a decir que ciertamente los productos de la categoría **tarjetas madre** están más relacionados con el total de devoluciones realizadas, como se ve en el siguiente gráfico.



Por lo tanto, se recomienda una reducción de precios de los productos con categoría **tarjetas madres y tarjetas de video** para reducir su acumulación, de esta forma se atraen más clientes que en base a los datos de búsquedas y ventas podemos inferir que son categorías de interés. Por otro lado, se tendrá que verificar la calidad de las tarjetas madres ya que como vimos, está asociada a la mayor cantidad de devoluciones.

CONCLUSIÓN

Al realizar esta actividad realmente me abrió mucho más mi campo de visión acerca de lo que el análisis de los datos nos pueden decir. En este caso detectamos mucha información que nos sería tan visible en primera instancia de la que pudimos sacar conclusiones e incluso estrategias a realizar para corregir aquellos datos negativos que nos surgieron. Estoy muy contento y orgulloso por el desarrollo de código y trabajo de análisis que pude efectuar.

REPOSITORIO GITHUB: <https://github.com/k1steel/emtech-projects.git>