

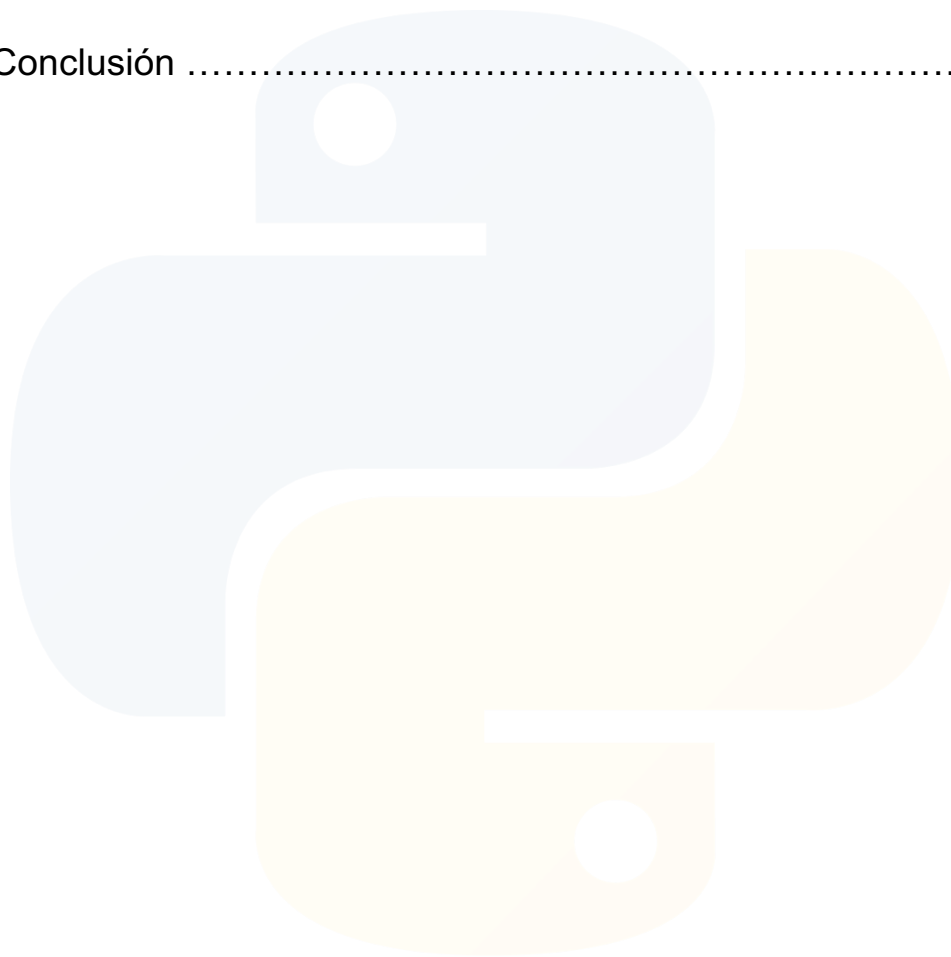
Proyecto 1

INTRODUCCIÓN A PYTHON

Itzel Yareni Otamendi Ortiz
EMTECH|

Índice

I.	Introducción.....	2
II.	Definición del código.....	3
III.	Solución del problema.....	25
IV.	Conclusión	26



Introducción

El objetivo del proyecto es llegar a un análisis de resultados en una tienda virtual de artículos electrónicos, por medio de listas de datos de los productos, características de ventas, clasificación y búsquedas.

Por medio de un código en Python usando definición de variables y listas, operadores lógicos y condicionales para la clasificación de datos, identificamos los productos más vendidos, productos que tienen menores ventas, categorías con menores búsquedas, productos por reseña, con los resultados planteamos una estrategia de los productos que no funcionan en ventas y los que son irrelevantes para los clientes y reducir el inventario según nuestros ingresos y ventas mensuales.

De esta forma podremos identificar métricas y el comportamiento de las tendencias para mejorar los resultados dentro de empresa y generar conclusiones efectivas y rápidas, lo que nos permite obtener información que se puede utilizar para mejorar procesos, aumentar la eficiencia y tomar decisiones efectivas en la tienda. Implementar esta tecnología en una organización como la del caso permite convertir los datos en elementos valiosos para conocer mejor a sus clientes y transformar su modelo de negocio.



Definición del Código

```
# 0 Login de usuario
# Para acceder al reporte, se requiere de ingresar nombre y
contraseña autorizados, los cuáles guardaremos en las siguientes
variables
usuario_correcto="Jaime"
contraseña_correcta="Pythonistas"

# Creamos la variable login para identificar cuándo debe continuar
pidiendo el nombre y contraseña, en caso de que no sean los
correctos.
# Se solicita ingresar el nombre y contraseña, en caso de ser
correctos desplegará el informe, de lo contrario, indicarpa si el
error es el usuario o la contraseña.
login = "True"
while (login == "True"):
    usuario = input("Para ver reporte, ingrese su nombre ")
    if usuario == usuario_correcto:
        contraseña = input("Por favor, ingrese su contraseña ")
        if contraseña == contraseña_correcta:
            print("Acceso exitoso ")
            print ()
            print("Reporte de Ventas ")
            print ()
            # 1.1.a. 15 productos con mayores ventas
            # Creamos una lista vacía que iremos llenando con el id
            producto y cada entrada referente a las ventas la inicializamos en
            cero.
            producto_ventas=[]
            for producto in lifestore_products:
                lista_interna=[producto[0],0]
                producto_ventas.append(lista_interna)
            # Imprimimos para validar la lista.
            #print(producto_ventas)

            # De la lista de ventas, tomamos el id producto y la variable
            que indica si hubo devolución, con ésta última crearemos la variable
            validez que será verdadera si el producto se devolvió. Así, sólo
            consideraremos cómo una venta aquellos que no se devolvieron, es
            decir, con validez cómo verdadera. Así la entrada de la lista
```



producto_ventas que en el paso anterior habíamos inicializado en cero, la llenaremos conforme las ventas válidas que se fueron dando para cada producto.

```
for venta in lifestore_sales:
    producto_vendido=venta[1]
    devuelto=venta[4]
    if devuelto==0:
        validez=True
    else:
        validez=False
    if validez==True:
        producto_ventas[producto_vendido-1][1]+=1
# Imprimimos para validar que se lleno correctamente la lista
con las ventas
#print(producto_ventas)

# Una vez que tenemos la lista de productos con sus respectivas
ventas, la ordenamos de forma descendente, para visualizar primero
las ventas más grandes. Luego, dado que sólo requerimos conocer los
15 productos más vendidos, de la lista total extraemos sólo los
primeros 15 elementos e imprimimos.
mayores_ventas=sorted(producto_ventas, key=lambda prod_vta:
prod_vta[1], reverse=True)
mayores_ventas15=mayores_ventas[0:15]
print("Los 15 productos más vendidos son:")
for i in range(len(mayores_ventas15)):
    print("El producto con ID "+ str(mayores_ventas15[i][0]) +"
tuvo "+str(mayores_ventas15[i][1])+ " ventas" )
print ()

# 1.1.b. 20 productos con mayores búsquedas
# Análogamente, cómo lo hicimos para el punto anterior,
generaremos una lista vacía que llenaremos con el id producto y cada
entrada referente a las búsquedas la inicializamos en cero.
producto_busquedas=[]
for producto2 in lifestore_products:
    lista_interna2=[producto2[0],0]
    producto_busquedas.append(lista_interna2)
# Imprimimos para verificar
#print(producto_busquedas)
```



De la lista de búsquedas, tomamos el id producto y la entrada de la lista producto_búsquedas que en el paso anterior habíamos inicializado en cero, la llenaremos conforme se registro una búsqueda por producto.

```
for busqueda in lifestore_searches:
    producto_buscado=busqueda[1]
    producto_búsquedas[producto_buscado-1][1]+=1
# Verificamos el llenado de la lista
#print(producto_búsquedas)
```

Una vez que tenemos la lista de productos con sus respectivas búsquedas, la ordenamos de forma descendente, para visualizar primero los productos que se buscaron con mayor frecuencia. Como sólo requerimos conocer los 20 productos más buscados, de la lista total extraemos sólo los primeros 20 elementos y los imprimimos.

```
mayores_búsquedas=sorted(producto_búsquedas, key=lambda
prod_busq: prod_busq[1], reverse=True)
mayores_búsquedas20=mayores_búsquedas[0:20]
print("Los 20 productos más buscados son:")
for j in range(len(mayores_búsquedas20)):
    print("El producto con ID "+ str(mayores_búsquedas20[j][0])
+" se buscó "+str(mayores_búsquedas20[j][1])+ " veces" )
print ()
```

Como paso intermedio, generaremos una lista de categorías únicas para identificarlas

```
lista_categorias=[]
for elemento in lifestore_products:
    categoria=[elemento[3]]
    lista_categorias.append(categoria)
# Visualizamos la lista vacía
#print(lista_categorias)
```

Recorremos la lista de categorías que recién creamos con el objetivo de ingresar en una lista vacía aquellos que no se repitan. Ésta lista nos sirve para conocer de manera rápida todas las categorías posibles.

```
categorias_unicas = []
for item in lista_categorias:
    if item not in categorias_unicas:
        categorias_unicas.append(item)
```



```

    # Imprimimos para ver que son categorías únicas como paso
    temporal.
    #print(categorias_unicas)

    # 1.2.a. 5 productos con menores ventas por categoría
    # Creamos una lista que contenga el producto, categoría y
    ventas, éste último estará inicializado en cero.
    producto_ventas_cat=[]
    for producto_cat in lifestore_products:
        lista_interna_cat=[producto_cat[0],producto_cat[3],0]
        producto_ventas_cat.append(lista_interna_cat)
    # Imprimimos para validar la lista vacía
    #print(producto_ventas_cat)

    # Acumularemos las ventas en la última entrada de la lista
    anterior, si y sólo si se trata de una venta válida, es decir, si no
    hubo devolución.
    for venta_cat in lifestore_sales:
        producto_vendido_cat=venta_cat[1]
        devuelto_cat=venta_cat[4]
        if devuelto_cat==0:
            validez_cat=True
        else:
            validez_cat=False
        if validez_cat==True:
            producto_ventas_cat[producto_vendido_cat-1][2]+=1
    # Imprimimos para verificar el llenado
    #print(producto_ventas_cat)

    # Creamos listas vacías por categoría, las cuáles identificamos
    con ayuda de la lista categorias_unicas que creamos en los pasos
    anteriores.
    ventas_procesadores=[]
    ventas_tarjetasvideo=[]
    ventas_tarjetasmadre=[]
    ventas_discosduros=[]
    ventas_memoriasusb=[]
    ventas_pantallas=[]
    ventas_bocinas=[]
    ventas_audifonos=[]

```



Revisaremos cada una de las entradas de la lista producto_ventas_cat y las clasificaremos en la lista que corresponda según su categoría.

```
for vta_cat in producto_ventas_cat:
    if vta_cat[1]=="procesadores":
        objeto_pro=vta_cat
        ventas_procesadores.append(objeto_pro)
    elif vta_cat[1]=="tarjetas de video":
        objeto_vid=vta_cat
        ventas_tarjetasvideo.append(objeto_vid)
    elif vta_cat[1]=="tarjetas madre":
        objeto_mad=vta_cat
        ventas_tarjetasmadre.append(objeto_mad)
    elif vta_cat[1]=="discos duros":
        objeto_dis=vta_cat
        ventas_discosduros.append(objeto_dis)
    elif vta_cat[1]=="memorias usb":
        objeto_usb=vta_cat
        ventas_memoriasusb.append(objeto_usb)
    elif vta_cat[1]=="pantallas":
        objeto_pan=vta_cat
        ventas_pantallas.append(objeto_pan)
    elif vta_cat[1]=="bocinas":
        objeto_boc=vta_cat
        ventas_bocinas.append(objeto_boc)
    else:
        objeto_aud=vta_cat
        ventas_audifonos.append(objeto_aud)
```

Una vez que tenemos la lista de productos con sus respectivas ventas, la ordenamos de forma ascendente, para visualizar primero los productos que se vendieron menos. Como sólo requerimos conocer los 5 productos menos vendidos, de la lista total extraemos sólo los primeros 5 elementos y los imprimimos.

```
# Lista de Procesadores
menores_ventas_procesadores=sorted(ventas_procesadores,
key=lambda proc_item: proc_item[2])
menores_ventas_procesadores5=menores_ventas_procesadores[0:5]
print("Los 5 productos menos vendidos en la categoría
Procesadores son:")
for i in range(len(menores_ventas_procesadores5)):
```




```

        print("El producto con ID "+
str(menores_ventas_procesadores5[i][0]) +" tuvo
"+str(menores_ventas_procesadores5[i][2])+ " ventas" )
        print ()

# Lista de Tarjetas de Video
menores_ventas_tarjetasvideo=sorted(ventas_tarjetasvideo,
key=lambda vid_item: vid_item[2])
menores_ventas_tarjetasvideo5=menores_ventas_tarjetasvideo[0:5]
print("Los 5 productos menos vendidos en la categoría Tarjetas
de Video son:")
for i in range(len(menores_ventas_tarjetasvideo5)):
    print("El producto con ID "+
str(menores_ventas_tarjetasvideo5[i][0]) +" tuvo
"+str(menores_ventas_tarjetasvideo5[i][2])+ " ventas" )
    print ()

# Lista de Tarjetas Madre
menores_ventas_tarjetasmadre=sorted(ventas_tarjetasmadre,
key=lambda mad_item: mad_item[2])
menores_ventas_tarjetasmadre5=menores_ventas_tarjetasmadre[0:5]
print("Los 5 productos menos vendidos en la categoría Tarjetas
Madre son:")
for i in range(len(menores_ventas_tarjetasmadre5)):
    print("El producto con ID "+
str(menores_ventas_tarjetasmadre5[i][0]) +" tuvo
"+str(menores_ventas_tarjetasmadre5[i][2])+ " ventas" )
    print ()

# Lista de Discos Duros
menores_ventas_discosduros=sorted(ventas_discosduros,
key=lambda dis_item: dis_item[2])
menores_ventas_discosduros5=menores_ventas_discosduros[0:5]
print("Los 5 productos menos vendidos en la categoría Discos
Duros son:")
for i in range(len(menores_ventas_discosduros5)):
    print("El producto con ID "+
str(menores_ventas_discosduros5[i][0]) +" tuvo
"+str(menores_ventas_discosduros5[i][2])+ " ventas" )
    print ()

```



```

# Lista de Memorias USB
menores_ventas_memoriasusb=sorted(ventas_memoriasusb,
key=lambda usb_item: usb_item[2])
menores_ventas_memoriasusb5=menores_ventas_memoriasusb[0:5]
print("Los 5 productos menos vendidos en la categoría Memorias
USB son:")
for i in range(len(menores_ventas_memoriasusb5)):
    print("El producto con ID "+
str(menores_ventas_memoriasusb5[i][0]) +" tuvo
"+str(menores_ventas_memoriasusb5[i][2])+ " ventas" )
    print ()

# Lista de Pantallas
menores_ventas_pantallas=sorted(ventas_pantallas, key=lambda
pan_item: pan_item[2])
menores_ventas_pantallas5=menores_ventas_pantallas[0:5]
print("Los 5 productos menos vendidos en la categoría Pantallas
son:")
for i in range(len(menores_ventas_pantallas5)):
    print("El producto con ID "+
str(menores_ventas_pantallas5[i][0]) +" tuvo
"+str(menores_ventas_pantallas5[i][2])+ " ventas" )
    print ()

# Lista de Bocinas
menores_ventas_bocinas=sorted(ventas_bocinas, key=lambda
boc_item: boc_item[2])
menores_ventas_bocinas5=menores_ventas_bocinas[0:5]
print("Los 5 productos menos vendidos en la categoría Bocinas
son:")
for i in range(len(menores_ventas_bocinas5)):
    print("El producto con ID "+
str(menores_ventas_bocinas5[i][0]) +" tuvo
"+str(menores_ventas_bocinas5[i][2])+ " ventas" )
    print ()

# Lista de Audífonos
menores_ventas_audifonos=sorted(ventas_audifonos, key=lambda
aud_item: aud_item[2])
menores_ventas_audifonos5=menores_ventas_audifonos[0:5]

```



```

        print("Los 5 productos menos vendidos en la categoría Audífonos
son:")
        for i in range(len(menores_ventas_audifonos5)):
            print("El producto con ID "+
str(menores_ventas_audifonos5[i][0]) +" tuvo
"+str(menores_ventas_audifonos5[i][2])+ " ventas" )
            print ()

# 1.2.b. 20 productos con menores búsquedas
# Creamos una lista que contenga el producto, categoría y
búsquedas, la última entrada la inicializamos en cero.
producto_busquedas_cat=[]
for producto_busq_cat in lifestore_products:

lista_internabusq_cat=[producto_busq_cat[0],producto_busq_cat[3],0]
    producto_busquedas_cat.append(lista_internabusq_cat)
# Imprimimos para validar
#print(producto_busquedas_cat)

# Acumularemos las búsquedas en la última entrada de la lista
anterior.
for busqueda_cat in lifestore_searches:
    producto_buscado_cat=busqueda_cat[1]
    producto_busquedas_cat[producto_buscado_cat-1][2]+=1
# Validamos llenado
#print(producto_busquedas_cat)

# Creamos listas vacías por categoría, las cuáles identificamos
con ayuda de la lista categorias_unicas que creamos en los pasos
anteriores.
busquedas_procesadores=[]
busquedas_tarjetasvideo=[]
busquedas_tarjetasmadre=[]
busquedas_discosduros=[]
busquedas_memoriasusb=[]
busquedas_pantallas=[]
busquedas_bocinas=[]
busquedas_audifonos=[]

```



Revisaremos cada una de las entradas de la lista producto_buscado_cat y las clasificaremos en la lista que corresponda según su categoría.

```
for bus_cat in producto_busquedas_cat:
    if bus_cat[1]=="procesadores":
        objeto_pro_b=bus_cat
        busquedas_procesadores.append(objeto_pro_b)
    elif bus_cat[1]=="tarjetas de video":
        objeto_vid_b=bus_cat
        busquedas_tarjetasvideo.append(objeto_vid_b)
    elif bus_cat[1]=="tarjetas madre":
        objeto_mad_b=bus_cat
        busquedas_tarjetasmadre.append(objeto_mad_b)
    elif bus_cat[1]=="discos duros":
        objeto_dis_b=bus_cat
        busquedas_discosduros.append(objeto_dis_b)
    elif bus_cat[1]=="memorias usb":
        objeto_usb_b=bus_cat
        busquedas_memoriasusb.append(objeto_usb_b)
    elif bus_cat[1]=="pantallas":
        objeto_pan_b=bus_cat
        busquedas_pantallas.append(objeto_pan_b)
    elif bus_cat[1]=="bocinas":
        objeto_boc_b=bus_cat
        busquedas_bocinas.append(objeto_boc_b)
    else:
        objeto_aud_b=bus_cat
        busquedas_audifonos.append(objeto_aud_b)
```

Una vez que tenemos la lista de productos con sus búsquedas correspondientes, la ordenamos de forma ascendente, para visualizar primero los productos que se buscaron menos. Como sólo requerimos conocer los 5 productos menos buscados, de la lista total extraemos sólo los primeros 5 elementos y los imprimimos.

Lista de Procesadores

```
menores_busquedas_procesadores=sorted(busquedas_procesadores,
key=lambda proc_item_b: proc_item_b[2])
```

```
menores_busquedas_procesadores5=menores_busquedas_procesadores[0:5]
```

```
print("Los 5 productos menos buscados en la categoría  
Procesadores son:")
```



```

        for i in range(len(menores_busquedas_procesadores5)):
            print("El producto con ID "+
str(menores_busquedas_procesadores5[i][0]) +" tuvo
"+str(menores_busquedas_procesadores5[i][2])+ " búsquedas" )
            print ()

        # Lista de Tarjetas Madre
        menores_busquedas_tarjetasmadre=sorted(busquedas_tarjetasmadre,
key=lambda mad_item_b: mad_item_b[2])

        menores_busquedas_tarjetasmadre5=menores_busquedas_tarjetasmadre[0:5]
        print("Los 5 productos menos buscados en la categoría Tarjetas
Madre son:")
        for i in range(len(menores_busquedas_tarjetasmadre5)):
            print("El producto con ID "+
str(menores_busquedas_tarjetasmadre5[i][0]) +" tuvo
"+str(menores_busquedas_tarjetasmadre5[i][2])+ " búsquedas" )
            print ()

        # Lista de Tarjetas de Video
        menores_busquedas_tarjetasvideo=sorted(busquedas_tarjetasvideo,
key=lambda vid_item_b: vid_item_b[2])

        menores_busquedas_tarjetasvideo5=menores_busquedas_tarjetasvideo[0:5]
        print("Los 5 productos menos buscados en la categoría Tarjetas
de Video son:")
        for i in range(len(menores_busquedas_tarjetasvideo5)):
            print("El producto con ID "+
str(menores_busquedas_tarjetasvideo5[i][0]) +" tuvo
"+str(menores_busquedas_tarjetasvideo5[i][2])+ " búsquedas" )
            print ()

        # Lista de Discos Duros
        menores_busquedas_discosduros=sorted(busquedas_discosduros,
key=lambda dis_item_b: dis_item_b[2])

        menores_busquedas_discosduros5=menores_busquedas_discosduros[0:5]
        print("Los 5 productos menos buscados en la categoría Discos
Duros son:")
        for i in range(len(menores_busquedas_discosduros5)):

```



```

        print("El producto con ID "+
str(menores_busquedas_discosduros5[i][0]) +" tuvo
"+str(menores_busquedas_discosduros5[i][2])+ " búsquedas" )
        print ()

# Lista de Memorias USB
menores_busquedas_memoriasusb=sorted(busquedas_memoriasusb,
key=lambda usb_item_b: usb_item_b[2])

menores_busquedas_memoriasusb5=menores_busquedas_memoriasusb[0:5]
        print("Los 5 productos menos buscados en la categoría Memorias
USB son:")
        for i in range(len(menores_busquedas_memoriasusb5)):
            print("El producto con ID "+
str(menores_busquedas_memoriasusb5[i][0]) +" tuvo
"+str(menores_busquedas_memoriasusb5[i][2])+ " búsquedas" )
            print ()

# Lista de Pantallas
menores_busquedas_pantallas=sorted(busquedas_pantallas,
key=lambda pan_item_b: pan_item_b[2])
menores_busquedas_pantallas5=menores_busquedas_pantallas[0:5]
        print("Los 5 productos menos buscados en la categoría Pantallas
son:")
        for i in range(len(menores_busquedas_pantallas5)):
            print("El producto con ID "+
str(menores_busquedas_pantallas5[i][0]) +" tuvo
"+str(menores_busquedas_pantallas5[i][2])+ " búsquedas" )
            print ()

# Lista de Bocinas
menores_busquedas_bocinas=sorted(busquedas_bocinas, key=lambda
boc_item_b: boc_item_b[2])
menores_busquedas_bocinas5=menores_busquedas_bocinas[0:5]
        print("Los 5 productos menos buscados en la categoría Bocinas
son:")
        for i in range(len(menores_busquedas_bocinas5)):
            print("El producto con ID "+
str(menores_busquedas_bocinas5[i][0]) +" tuvo
"+str(menores_busquedas_bocinas5[i][2])+ " búsquedas" )
            print ()

```



```

# Lista de Audífonos
menores_busquedas_audifonos=sorted(busquedas_audifonos,
key=lambda aud_item_b: aud_item_b[2])
menores_busquedas_audifonos5=menores_busquedas_audifonos[0:5]
print("Los 5 productos menos buscados en la categoría Audífonos
son:")
for i in range(len(menores_busquedas_audifonos5)):
    print("El producto con ID "+
str(menores_busquedas_audifonos5[i][0]) +" tuvo
"+str(menores_busquedas_audifonos5[i][2])+ " búsquedas" )
    print ()

```

```

# 2.1 10 productos con las mejores reseñas
# Creamos una lista vacía que iremos llenando con el id
producto, ventas y puntaje de reseña.
producto_ventas_score=[]
for producto_score in lifestore_products:
    lista_interna_score=[producto_score[0],0,0]
    producto_ventas_score.append(lista_interna_score)
# Imprimimos para validar la lista que creamos
#print(producto_ventas_score)

```

De la lista de ventas, tomamos el id producto y la variable que indica si hubo devolución, con ésta última crearemos la variable validez que será verdadera si el producto se devolvió. Así, sólo consideraremos cómo una venta aquellos que no se devolvieron, es decir, con validez cómo verdadera. Así las entradas de la lista producto_ventas_score que en el paso anterior habíamos inicializado en cero (es decir, ventas y score), las llenaremos conforme las ventas válidas que se fueron dando para cada producto.

```

for venta_score in lifestore_sales:
    producto_vendido_score=venta_score[1]
    devuelto_score=venta_score[4]
    reseña=venta_score[2]
    if devuelto_score==0:
        validez_score=True
    else:
        validez_score=False
    if validez_score==True:
        producto_ventas_score[producto_vendido_score-1][1]+=1

```



```

        producto_ventas_score[producto_vendido_score-1][2]+=reseña
# Validamos que se lleno correctamente
#print(producto_ventas_score)

# Una vez que tenemos la lista de productos con sus respectivas
ventas y la suma del puntaje de las reseñas, calcularemos la reseña
promedio como la suma de las reseñas entre la suma de las ventas.
for valor in producto_ventas_score:
    suma_reseñas=valor[2]
    suma_ventas=valor[1]
    if suma_ventas>0:
        valor[2]=suma_reseñas/suma_ventas
# Validamos la reseña promedio
#print(producto_ventas_score)

# Ordenamos de forma descendente la lista anterior, para
visualizar primero las reseñas con mejor puntaje. Luego, dado que
sólo requerimos conocer los 10 productos mejor reseñados, de la lista
total extraemos sólo los primeros 10 elementos e imprimimos.
mejores_reseñas=sorted(producto_ventas_score, key=lambda
prod_vta_sco: prod_vta_sco[2], reverse=True)
mejores_reseñas10=mejores_reseñas[0:10]
print("Los 10 productos con mejores reseñas son:")
for i in range(len(mejores_reseñas10)):
    print("El producto con ID "+ str(mejores_reseñas10[i][0]) +"
tiene "+str(mejores_reseñas10[i][2])+ " de calificación" )
print ()

# 2.2 10 productos con las peores reseñas
# Ordenamos de forma ascendente la lista del punto anterior,
para visualizar primero las reseñas con peor puntaje. Luego, dado que
sólo requerimos conocer los 10 productos peor reseñados, de la lista
total extraemos sólo los primeros 10 elementos e imprimimos.
peores_reseñas=sorted(producto_ventas_score, key=lambda
prod_vta_sco2: prod_vta_sco2[2])
peores_reseñas10=peores_reseñas[0:10]
print("Los 10 productos con peores reseñas son:")
for i in range(len(peores_reseñas10)):
    print("El producto con ID "+ str(peores_reseñas10[i][0]) +"
tiene "+str(peores_reseñas10[i][2])+ " de calificación" )
print ()

```




```

# 3.1 Ventas
# Cómo paso intermedio, generaremos una lista de mes-año únicas
para identificarlos
    lista_mes_año=[]
    for entrada in lifestore_sales:
        date_m=entrada[3][-7:][:2]
        date_a=entrada[3][-4:]
        date_ma=date_m+"-"+date_a
        #print(date_ma)
        lista_mes_año.append(date_ma)
    #print(lista_mes_año)

# Tomamos elementos únicos de la lista
mes_año_unicas = []
for item_ma in lista_mes_año:
    if item_ma not in mes_año_unicas:
        mes_año_unicas.append(item_ma)
# Validamos lista
#print(mes_año_unicas)

# Creamos una lista de productos y mes-año de ventas válidas,
es decir, que no fueron devueltas.
ventas_ma=[]
ventas_ma_dev=[]
for venta_ma in lifestore_sales:
    date_m2=venta_ma[3][-7:][:2]
    date_a2=venta_ma[3][-4:]
    date_ma2=date_m2+"-"+date_a2
    lista_interna_ma=[venta_ma[1],date_ma2,venta_ma[4]]
    if lista_interna_ma[2]==0:
        ventas_ma.append(lista_interna_ma)
    else:
        ventas_ma_dev.append(lista_interna_ma)
# Validamos ventas válidas y devoluciones
#print(ventas_ma)
#print(ventas_ma_dev)

# Generamos una lista por mes_año para más adelante llenar los
productos que se vendieron en esa fecha.
mes_año_productos = []

```



```

for elemento in mes_año_unicas:
    ma_prods = [elemento, []]
    mes_año_productos.append(ma_prods)
# Validamos lista vacía
#print(mes_año_productos)

# Llenamos la lista anterior con los productos vendidos en cada
mes-año.
for lista in mes_año_productos:
    mes_año_de_lista = lista[0]
    for producto_vend_ma in ventas_ma:
        id_del_producto = producto_vend_ma[0]
        mes_año_del_producto = producto_vend_ma[1]
        if mes_año_del_producto == mes_año_de_lista:
            lista[1].append(id_del_producto)
# Validamos mes año
#print(mes_año_productos)

# Hacemos el conteo del número de productos vendidos por cada
mes-año.
productos_vendidos_mes_año=[]
for llave in mes_año_productos:
    prod_vendidos_mes_año=len(llave[1])
    lista_intermedia_ma=[llave[0],prod_vendidos_mes_año]
    productos_vendidos_mes_año.append(lista_intermedia_ma)
# Validamos el conteo de ventas
#print(productos_vendidos_mes_año)

# Hacemos una lista con los meses con más ventas
mes_año_mayores_ventas=sorted(productos_vendidos_mes_año,
key=lambda cta: cta[1], reverse=True)
print("Los meses con más ventas son: ")
for z in range(len(mes_año_mayores_ventas)):
    print(str(mes_año_mayores_ventas[z][0]) +" con
"+str(mes_año_mayores_ventas[z][1])+ " productos vendidos" )
print ()

# Calculamos las ventas promedio mensuales
ventas_totales=0
for y in range(len(mes_año_mayores_ventas)):
    ventas_totales+=mes_año_mayores_ventas[y][1]

```



```

ventas_promedio_mensuales=ventas_totales/len(mes_año_mayores_ventas)
    print("En promedio, se venden "+
str(int(ventas_promedio_mensuales)) +" productos cada mes")
    print ()

# 3.2 Ingresos
# Cómo paso intermedio, generaremos una lista de id-producto y
precio únicos para identificarlos
lista_producto_precio=[]
for precio in lifestore_products:
    prod_precio=[precio[0],precio[2]]
    lista_producto_precio.append(prod_precio)
# Validamos estructura de lista vacía
#print(lista_producto_precio)

# Llenamos la lista con los precios
producto_precio_unicos = []
for item_pp in lista_producto_precio:
    if item_pp not in producto_precio_unicos:
        producto_precio_unicos.append(item_pp)
# Verificamos llenado
#print(producto_precio_unicos)

# Generamos una lista por mes_año para llenar los productos que
se vendieron en esa fecha y su precio.
mes_año_productos_precio = []
for elemento_pp in mes_año_productos:
    pp_prods = [elemento_pp]
    mes_año_productos_precio.append(pp_prods)
# Validamos
#print(mes_año_productos_precio)

# Calcularemos los ingresos para cada mes-año
# Ingresos de 0720
# Creamos lista con los precios de los productos vendidos en
cada mes
a_precio=[]
ventas_0720=mes_año_productos_precio[0][0][1]
for a in ventas_0720:
    prod_a = a

```



```

    for prod_aa in producto_precio_unicos:
        id_del_producto_a = prod_aa[0]
        precio_del_producto_a = prod_aa[1]
        if id_del_producto_a == prod_a:
            a_precio.append(precio_del_producto_a)
    consolidado_a=[mes_año_productos_precio[0][0][0],a_precio]

    # Inicializo mi variable de ingresos del mes en cero y la voy
    acumulando con la suma de los precios de cada vector
    ingresos_0720=0
    for ing in a_precio:
        ingresos_0720+=ing
    ma_ing_0720=[mes_año_productos_precio[0][0][0],ingresos_0720]

    # Ingresos de 0220
    # Creamos lista con los precios de los productos vendidos en
    cada mes
    b_precio=[]
    ventas_0220=mes_año_productos_precio[1][0][1]
    for b in ventas_0220:
        prod_b = b
        for prod_bb in producto_precio_unicos:
            id_del_producto_b = prod_bb[0]
            precio_del_producto_b = prod_bb[1]
            if id_del_producto_b == prod_b:
                b_precio.append(precio_del_producto_b)
    consolidado_b=[mes_año_productos_precio[1][0][0],b_precio]

    # Inicializo mi variable de ingresos del mes en cero y la voy
    acumulando con la suma de los precios de cada vector
    ingresos_0220=0
    for ing_b in b_precio:
        ingresos_0220+=ing_b
    ma_ing_0220=[mes_año_productos_precio[1][0][0],ingresos_0220]

    # Ingresos de 0520
    # Creamos lista con los precios de los productos vendidos en
    cada mes
    c_precio=[]
    ventas_0520=mes_año_productos_precio[2][0][1]
    for c in ventas_0520:

```



```

    prod_c = c
    for prod_cc in producto_precio_unicos:
        id_del_producto_c = prod_cc[0]
        precio_del_producto_c = prod_cc[1]
        if id_del_producto_c == prod_c:
            c_precio.append(precio_del_producto_c)
    consolidado_c=[mes_año_productos_precio[2][0][0],c_precio]

    # Inicializo mi variable de ingresos del mes en cero y la voy
    acumulando con la suma de los precios de cada vector
    ingresos_0520=0
    for ing_c in c_precio:
        ingresos_0520+=ing_c
    ma_ing_0520=[mes_año_productos_precio[2][0][0],ingresos_0520]

    # Ingresos de 0120
    # Creamos lista con los precios de los productos vendidos en
    cada mes
    d_precio=[]
    ventas_0120=mes_año_productos_precio[3][0][1]
    for d in ventas_0120:
        prod_d = d
        for prod_dd in producto_precio_unicos:
            id_del_producto_d = prod_dd[0]
            precio_del_producto_d = prod_dd[1]
            if id_del_producto_d == prod_d:
                d_precio.append(precio_del_producto_d)
    consolidado_d=[mes_año_productos_precio[3][0][0],d_precio]

    # Inicializo mi variable de ingresos del mes en cero y la voy
    acumulando con la suma de los precios de cada vector
    ingresos_0120=0
    for ing_d in d_precio:
        ingresos_0120+=ing_d
    ma_ing_0120=[mes_año_productos_precio[3][0][0],ingresos_0120]

    # Ingresos de 0420
    # Creamos lista con los precios de los productos vendidos en
    cada mes
    e_precio=[]
    ventas_0420=mes_año_productos_precio[4][0][1]

```



```

for e in ventas_0420:
    prod_e = e
    for prod_ee in producto_precio_unicos:
        id_del_producto_e = prod_ee[0]
        precio_del_producto_e = prod_ee[1]
        if id_del_producto_e == prod_e:
            e_precio.append(precio_del_producto_e)
    consolidado_e=[mes_año_productos_precio[4][0][0],e_precio]

    # Inicializo mi variable de ingresos del mes en cero y la voy
    acumulando con la suma de los precios de cada vector
    ingresos_0420=0
    for ing_e in e_precio:
        ingresos_0420+=ing_e
    ma_ing_0420=[mes_año_productos_precio[4][0][0],ingresos_0420]

    # Ingresos de 0320
    # Creamos lista con los precios de los productos vendidos en
    cada mes
    f_precio=[]
    ventas_0320=mes_año_productos_precio[5][0][1]
    for f in ventas_0320:
        prod_f = f
        for prod_ff in producto_precio_unicos:
            id_del_producto_f = prod_ff[0]
            precio_del_producto_f = prod_ff[1]
            if id_del_producto_f == prod_f:
                f_precio.append(precio_del_producto_f)
        consolidado_f=[mes_año_productos_precio[5][0][0],f_precio]

        # Inicializo mi variable de ingresos del mes en cero y la voy
        acumulando con la suma de los precios de cada vector
        ingresos_0320=0
        for ing_f in f_precio:
            ingresos_0320+=ing_f
        ma_ing_0320=[mes_año_productos_precio[5][0][0],ingresos_0320]

        # Ingresos de 0620
        # Creamos lista con los precios de los productos vendidos en
        cada mes
        g_precio=[]

```



```

ventas_0620=mes_año_productos_precio[6][0][1]
for g in ventas_0620:
    prod_g = g
    for prod_gg in producto_precio_unicos:
        id_del_producto_g = prod_gg[0]
        precio_del_producto_g = prod_gg[1]
        if id_del_producto_g == prod_g:
            g_precio.append(precio_del_producto_g)
consolidado_g=[mes_año_productos_precio[6][0][0],g_precio]

# Inicializo mi variable de ingresos del mes en cero y la voy
acumulando con la suma de los precios de cada vector
ingresos_0620=0
for ing_g in g_precio:
    ingresos_0620+=ing_g
ma_ing_0620=[mes_año_productos_precio[6][0][0],ingresos_0620]

# Ingresos de 1120
# Creamos lista con los precios de los productos vendidos en
cada mes
h_precio=[]
ventas_1120=mes_año_productos_precio[7][0][1]
for h in ventas_1120:
    prod_h = h
    for prod_hh in producto_precio_unicos:
        id_del_producto_h = prod_hh[0]
        precio_del_producto_h = prod_hh[1]
        if id_del_producto_h == prod_h:
            h_precio.append(precio_del_producto_h)
consolidado_h=[mes_año_productos_precio[7][0][0],h_precio]

# Inicializo mi variable de ingresos del mes en cero y la voy
acumulando con la suma de los precios de cada vector
ingresos_1120=0
for ing_h in h_precio:
    ingresos_1120+=ing_h
ma_ing_1120=[mes_año_productos_precio[7][0][0],ingresos_1120]

# Ingresos de 0920
# Creamos lista con los precios de los productos vendidos en
cada mes

```



```

i_precio=[]
ventas_0920=mes_año_productos_precio[8][0][1]
for i in ventas_0920:
    prod_i = i
    for prod_ii in producto_precio_unicos:
        id_del_producto_i = prod_ii[0]
        precio_del_producto_i = prod_ii[1]
        if id_del_producto_i == prod_i:
            i_precio.append(precio_del_producto_i)
consolidado_i=[mes_año_productos_precio[8][0][0],i_precio]

# Inicializo mi variable de ingresos del mes en cero y la voy
acumulando con la suma de los precios de cada vector
ingresos_0920=0
for ing_i in i_precio:
    ingresos_0920+=ing_i
ma_ing_0920=[mes_año_productos_precio[8][0][0],ingresos_0920]

# Ingresos de 0820
# Creamos lista con los precios de los productos vendidos en
cada mes
j_precio=[]
ventas_0820=mes_año_productos_precio[9][0][1]
for j in ventas_0820:
    prod_j = j
    for prod_jj in producto_precio_unicos:
        id_del_producto_j = prod_jj[0]
        precio_del_producto_j = prod_jj[1]
        if id_del_producto_j == prod_j:
            j_precio.append(precio_del_producto_j)
consolidado_j=[mes_año_productos_precio[9][0][0],j_precio]

# Inicializo mi variable de ingresos del mes en cero y la voy
acumulando con la suma de los precios de cada vector
ingresos_0820=0
for ing_j in j_precio:
    ingresos_0820+=ing_j
ma_ing_0820=[mes_año_productos_precio[9][0][0],ingresos_0820]

# Consolidamos los ingresos mensuales en una lista

```




```

ingresos_totales=[ma_ing_0720,ma_ing_0220,ma_ing_0520,ma_ing_0120,ma_
ing_0420,ma_ing_0320,ma_ing_0620,ma_ing_1120,ma_ing_0920,ma_ing_0820]
    #print(ingresos_totales)

    # Hacemos una lista con los ingresos totales por mes
    mes_año_mayores_ingresos=sorted(ingresos_totales, key=lambda
cta_ing: cta_ing[1], reverse=True)
    print("Los ingresos totales mensuales son: ")
    for m in range(len(mes_año_mayores_ingresos)):
        print(str(mes_año_mayores_ingresos[m][0]) + " con
"+str(mes_año_mayores_ingresos[m][1])+ " de ingreso" )
    print ()

    # Ingreso anual
    # Sumamos los ingresos mensuales de la lista anterior
    ingreso_total_global=0
    for ing_k in ingresos_totales:
        ingreso_total_global+=ing_k[1]
    print("El ingreso total anual es: "+str(ingreso_total_global))
    print ()

    # Una vez que se visualiza el reporte, se cambia el valor de la
variable login para que se salga del ciclo while que pide usuario y
contraseña.
    login = 'False'
    # Si la contraseña es incorrecta, te hará saber el error.
    else:
        print("Contraseña incorrecta. Intenta de nuevo.")
    # Si el usuario es incorrecto, te pedirá verificar.
    else:
        print("Usuario Incorrecto. Verifica tu usuario.")
# Fin del código

```

Link del código:

<https://replit.com/@Yareni01/ImmaculateWillingOutcome#PROYECTO-01-OTAMENDI-ITZEL.py>



Solución del problema

De acuerdo al análisis de los productos más vendidos y los más buscados, identificamos que el producto más vendido es también el más buscado, es decir, el ID Producto=54 que corresponde a "SSD Kingston A400, 120GB, SATA III, 2.5", 7mm" de la categoría "Discos Duros".

Así mismo, el top 15 de los productos más vendidos aparece en el top 20 de los productos más buscados, lo que nos hace inferir que entre mayores búsquedas tenga un producto, mayores posibilidades de venta tendrá, o bien que los productos más vendidos son los más buscados.

Respecto a los productos de las categorías con menores búsquedas y las de menores ventas, las categorías de no tuvieron ventas son Tarjetas de Video, Tarjetas Madre, Discos Duros, Pantallas, Bocinas y Audífonos, cuyo comportamiento es muy similar en el caso de los productos menos buscados en las mismas categorías.

Con base en los productos mejor reseñados, no se visualiza una relación clara con los productos ni más vendidos ni con los más buscados. Así el puntaje de las reseñas no cuenta con el volumen suficiente de ventas o búsquedas para ser considerada una variable de peso en la toma de decisiones, se deberá esperar a que la variable alcance una madurez en cuanto a reseñas por parte de los clientes. Análogamente, los productos que aparecen con las peores reseñas, es decir con un puntaje de cero, no se incluyen en las listas de los productos más vendidos, así, dado que no se venden, no existe una reseña para ellos.

En cuanto a las ventas e ingresos mensuales, observamos que se corresponden directamente y que en el primer cuatrimestre del año se presentaron ingresos mensuales mayores a 100,000 mxn y ventas por arriba del promedio de productos mensuales, con el máximo en el mes de Abril 2020, representando el 26% de los ingresos totales anuales.

Así, considerando los puntos arriba mencionados, se sugiere retirar los productos de las categorías con búsquedas y ventas cero, ya que no existe una demanda a pesar de la oferta que se tiene. También se sugiere reducir el inventario acumulado en los meses de menores ventas, es decir, en Septiembre y Noviembre, mediante estrategias de marketing que motiven a la compra de los productos rezagados.

Conclusión

Uno de los beneficios que nos da un análisis de datos con este tipo de herramientas es la facilidad con la que se pueden leer resultados de el comportamiento de los clientes evaluando las mejores prácticas para el negocio y vale la pena seguir implementado estrategias que detecten las áreas de oportunidad, los riesgos que surgen y lograr el impulso de la empresa.

También facilita la toma de decisiones, el tener datos concisos ayuda a tomar decisiones inteligentes evaluando las características y variables dentro del ambiente, como por ejemplo si tenemos en mente lanzar un producto en que época seria la mejor opción evaluando los comportamientos anteriores, saber lo que desean los clientes, las necesidades que tienen o cual es el perfil del cliente que adquiere nuestros productos va a ser posible apartir del analisis de datos, e ir a lo seguro en el lazamiento o estrategia de negocios.

