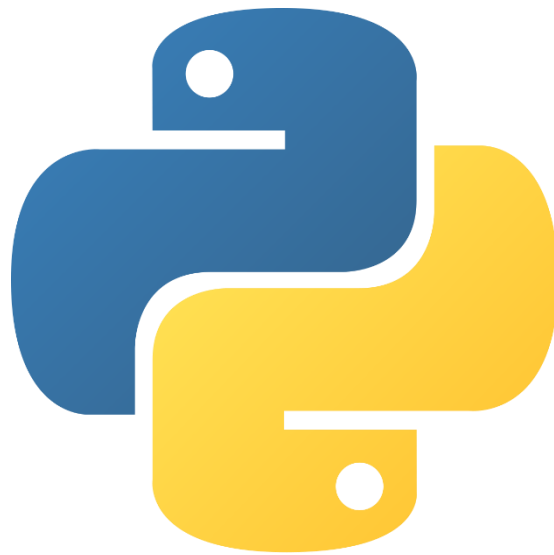




PROYECTO 1.

INTRODUCCIÓN A PYTHON



ALEJANDRO BERNAL NAJERA

Índice

Índice	2
Introducción	3
Definición del código	4
- CODIGO	4
- RESULTADOS	11
Solución al problema	15
Conclusión	15
Código Github	15

Introducción

En esta primera parte del curso se estuvo viendo una pequeña parte de lo que es Python, en la primera parte fueron los fundamentos de programación donde se tocaron temas generales que cualquier lenguaje de programación tiene pero aplicados a Python así como listas, como manipularlas y los diferentes operadores que existe. En la parte final se vieron las sentencias de control de flujo como if, while y for así como las opciones que se pueden usar en estos como continue y break.

Al final para poner en práctica todo lo que se aprendió se planteó el problema de una tienda virtual que maneja diferentes productos electrónicos, y simulando que la gerencia de ventas se da cuenta que hay acumulación de inventario aparte de que hay productos que siempre se buscan y hay otros que no lo cual hace que sus ventas estén bajando, para eso nos piden la ayuda para mostrar tablas las cuales se explicaran en el documento para observar la tendencia que están teniendo, el propósito de este proyecto es encontrar una solución a este problema y tratar que haya más ventas de esos productos que no se vendan.

Definición del código

- CODIGO

```
#-----LOGIN-----

# (Usuario = alex /contraseña = 123)

while(True):
    #el usuario ingresa el nombre y contraseña
    user = input("Ingrese su usuario: ")
    passw = input("Ingrese su contraseña: ")

    #Se verifica que el usuario y contraseña esten correctas para salir del while
    if (user == "alex" and passw == "123"):
        break
    #en caso que sea incorrecto muestra el mensaje y regresa a pedir los datos
    else:
        print("\n\n**ERROR: USUARIO/CONTRASEÑA INCORRECTOS**\n\n")
```

```
#-----LISTAS-----

#lifestore_searches = [id_search, id product]
#lifestore_sales = [id_sale, id_product, score (from 1 to 5), date, refund (1 for
true or 0 to false)]
#lifestore_products = [id_product, name, price, category, stock]

lifestore_products = [[1, 'Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Quad-Core,
16MB L2 Cache', 3019, 'procesadores', 16]...]
lifestore_sales = [[1, 1, 5, '24/07/2020', 0]...]
lifestore_searches = [[1, 1]...]
```

```

#Primero se tiene toda la logica de las tablas para que esten listas en el momento
que las pidan despues de hacer el login
#-----TABLA 1-----

#se obtiene todas ventas solo los id de producto donde no se haya devuelto
allSales = [x[1] for x in lifestore_sales if x[4]!=1]
#se crea una lista para contar las ventas de los productos [idprod,0]
prod_sales = [[x[0],0] for x in lifestore_products]

#por cada idProducto en las ventas se suma 1 cada vez que se repita el id
for sales in allSales:
    prod_sales[sales-1][1] += 1

#funcion para obtener el segundo elemento de una lista
def segundoElemento(lista):
    return lista[1]

#Ordenar de mayor a menor dado el segundo elemento
prod_sales.sort(reverse=True, key=segundoElemento)
#obtener los primeros 15 productos
prod_sales = prod_sales[:15]

# [ #de ventas , nombreProducto ]
#el nombre del producto se corta a 50 caracteres para una mejor visualizacion en las
tablas
Tabla1 = [ [x[1], lifestore_products[x[0]-1][1][:50]] for x in prod_sales ]

```

```

#-----TABLA 2-----

#se obtiene todas busquedas solo los id de producto
allSearches = [x[1] for x in lifestore_searches]
#se crea una lista para contar las busquedas de los productos [idprod,0]
prod_searches = [[x[0],0] for x in lifestore_products]

#por cada idProducto en las busquedas se suma 1 cada vez que se repita el id
for searches in allSearches:
    prod_searches[searches-1][1] += 1

#Ordenar de mayor a menor dado el segundo elemento
prod_searches.sort(reverse=True, key=segundoElemento)
#obtener los primeros 20 productos
prod_searches2 = prod_searches[:20]

# [ #de busquedas , nombreProducto ]
#el nombre del producto se corta a 50 caracteres para una mejor visualizacion en las
tablas
Tabla2 = [ [x[1], lifestore_products[x[0]-1][1][:50]] for x in prod_searches2 ]

```

```

#-----TABLA 3-----

#se crea una lista para contar las ventas de los productos [idprod,0,''] donde el
tercer valor es la categoria
prod_sales_cat = [[x[0],0,''] for x in lifestore_products]

#por cada idProducto en las ventas se suma 1 cada vez que se repita el id
for sales in allSales:
    prod_sales_cat[sales-1][1] += 1

#esta lista es para quitar los productos que tengan 0 ventas
prod_sales2 = [x for x in prod_sales_cat if x[1]!=0]

#for para obtener la categoria en los productos y cambiarla
for x in prod_sales2:
    x[2] = lifestore_products[x[0]-1][3]

# [ #de ventas , nombreProducto ]
#el nombre del producto se corta a 50 caracteres para una mejor visualizacion en las
tablas
salesTables = [[x[2], lifestore_products[x[0]-1][1][:50], x[1]] for x in prod_sales2
]

def tercerElemento(lista):
    return lista[2]

def primerElemento(lista):
    return lista[0]

#Primero ordenamos la lista por la cantidad de ventas y despues por categoria
salesTables.sort(key=tercerElemento)
salesTables.sort(key=primerElemento)

#limitar que cada categoria solo imprima 5 productos ya despues de estar ordenadas
#utilizo un diccionario para obtener el resultado en la categoria y asi irle sumando
cada vez para que solo sean 5
contador = {x:0 for x in list(set(x[0] for x in salesTables)) }
Tabla3=[]
for x in salesTables:
    if contador[x[0]] < 5:
        Tabla3.append(x)
        contador[x[0]]+=1

```

```

#-----TABLA 4-----

#La lista que ya teniamos de busquedas donde esta el id con el numero de busquedas
prod_searches.sort(key=segundoElemento)
#sacamos los productos que tienen busquedas 0
prod_searches2 = [x for x in prod_searches if x[1]!=0]
#obtener los primeros 20 productos
prod_searches2 = prod_searches2[:20]

# [ #de busquedas , nombreProducto ]
#el nombre del producto se corta a 50 caracteres para una mejor visualizacion en las
tablas
Tabla4 = [ [x[1], lifestore_products[x[0]-1][1][:50]] for x in prod_searches2 ]

```

```

#-----TABLA 5-----

#lista para obtener el id del producto y la reseña que haya tenido en la compra
resena_producto = [[x[1],x[2]] for x in lifestore_sales]
#lista que se utilizara para sacar el promedio
score_producto = [[x[0],0,0] for x in lifestore_products]

#en el segundo valor se va a ir haciendo la suma de todas las reseñas
#en el tercer valor se hace una suma en 1 para ver la cantidad de busquedas
for score in resena_producto:
    score_producto[score[0]-1][1] += score[1]
    score_producto[score[0]-1][2] += 1

#se limita la lista para que no tome los productos que no tienen venta y por lo
tanto no tienen reseña
score_producto2 = [x for x in score_producto if x[1]>0]
#en la lista anterior teniamos [id, sumatoria de las reseñas, numero de reseñas]
#aqui solamente obtenermos el promedio diviendo la sumatoria de las reseñas entre el
numero de reseñas que hay y se redondea el valor
score_producto3 = [ [ x[0], round(x[1]/x[2],2) ] for x in score_producto2 ]

#se ordena de mayor a menor
score_producto3.sort(reverse=True, key=segundoElemento)

#dado la lista anterior solo se obtiene los nombres de producto
bestScoreTable = [ [x[1], lifestore_products[x[0]-1][1][:50]] for x in
score_producto3]

#se eligen las 10 mejores
Tabla5 = bestScoreTable[:10]

```

```

#-----TABLA 6-----

#de la tabla de las mejores reseñas ordenarlas de menor a mayor y obtener las 10
peores
bestScoreTable.sort()
Tabla6 = bestScoreTable[:10]

```

```
#-----TABLA 7-----

#obtenemos todas las ventas [ id producto, mes] el mes se convierte a int cuando
solo se elige el esos dos numeros en la fecha
allSalesDate = [[x[1],int(x[3][3:5])] for x in lifestore_sales if x[4]!=1]
#se crea una lista con los meses y 0 en los ingresos
Tabla7 = [[x,0] for x in range(1,13)]

#en cada venta se checha de cual mes es y hace una sumatoria con el costo del
producto correspondiente
for sales in allSalesDate:
    Tabla7[sales[1]-1][1] += lifestore_products[sales[0]-1][2]
```

```
#-----TABLA 8-----

#se crea una lista con los meses y 0 el numero de ventas
meses_sales = [[x,0] for x in range(1,13)]

#en cada venta se checha de cual mes es y hace una sumatoria en 1 por cada venta que
tuvo
for sales in allSalesDate:
    meses_sales[sales[1]-1][1] += 1

#para obtener las ventas promedio mensuales se divide el total de ingresos del mes /
la cantidad vendida en el mes
Tabla8 = []
for mes in Tabla7:
    if mes[1]!=0:
        Tabla8.append( [mes[0], mes[1] / meses_sales[mes[0]-1][1]])
    else:
        Tabla8.append(mes)
```

```
#-----TABLA 9-----

#se hace casi lo mismo que el punto 8 pero ahora se ordena de mayor a menor y se
quitan los meses que no tienen ventas
meses_sales.sort(reverse=True, key=segundoElemento)
Tabla9 = [ x for x in meses_sales if x[1]!=0]
```

```
#-----TABLA 10-----

#Utilizando la lista de la primera tabla donde estan todas las ventas se hace una
sumatoria por cada id en allSales del costo de cada producto correspondiente
Tabla10 = sum([lifestore_products[x-1][2] for x in allSales])
```



```

#-----MENU-----

while (True):

    print("""
+-----MENU-----+
|
| 1 -> 15 PRODUCTOS CON MAYORES VENTAS.
| 2 -> 20 PRODUCTOS CON MAYOR BÚSQUEDAS.
| 3 -> 5 PRODUCTOS CON MENORES VENTAS POR CATEGORIA.
| 4 -> 20 PRODUCTOS CON MENORES BÚSQUEDAS.
| 5 -> 10 PRODUCTOS CON LAS MEJORES RESEÑAS.
| 6 -> 10 PRODUCTOS CON LAS PEORES RESEÑAS.
| 7 -> TOTAL DE INGRESOS MENSUALES
| 8 -> VENTAS PROMEDIO MENSUALES
| 9 -> MESES CON MAS VENTAS AL AÑO (2020)
| 10 -> TOTAL ANUAL (2020)
| 0 -> PARA SALIR
|
+-----+
""")

    opcion=input("Elige la opcion deseada: ")

    #se comprueba que el input de opcion este en el rango de las opciones y que sea
    un numero
    if opcion.isnumeric() and int(opcion)>=0 and int(opcion)<=10:

        #se convierte a int para mejor manejo en los if
        opcion = int(opcion)

```

Todas las opciones es exactamente lo mismo lo unico que cambia es la tabla que se imprime

```
#-----OPCIONES PARA IMPRIMIR-----

    if opcion == 1:
        print('\n'*30)
        Tabla = """
\n\n+-----15 PRODUCTOS CON MAYORES VENTAS-----+
|#VENTAS                PRODUCTO                |
|-----|
{}
+-----+
\n"""

        #por cada fila se imprimen los datos y se juntan en una misma con format
        #con los espacios 5 para el id y 55 para el nombre del producto
        Tabla = (Tabla.format('\n'.join("| {:<5} {:<55} |".format(*fila) for
fila in Tabla1)))
        print (Tabla)

        #se repite cada vez que el input de regresar no sea 1 y si es 1 sigue
        son el ciclo normal
        while (True):
            regresar = input("Ingresa 1 para regresar al menú: ")

            if regresar.isnumeric() and int(regresar) == 1:
                print('\n'*30)
                break
            else:
                print("***ERROR: Ingresa un numero Valido**\n")
```

Después de tener todos los if con su tabla correspondiente, está la opción de 0 para salir y como última parte del código es un else que conecta con el primer if que checa si lo que se ingresó para el menú es incorrecto

```
elif opcion == 0:
    print("SALIENDO.....")
    break

#Else en caso que ingrese otra cosa que no este en el Menu
else:
    print('\n'*30)
    print("*****ERROR: INGRESA UN NUMERO / NUMERO VALIDO*****")
```

- RESULTADOS

-----15 PRODUCTOS CON MAYORES VENTAS-----	
#VENTAS	PRODUCTO
49	SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm
42	Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-C
20	Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Q
18	Tarjeta Madre ASRock Micro ATX B450M Steel Legend,
15	SSD Adata Ultimate SU800, 256GB, SATA III, 2.5'',
13	Procesador AMD Ryzen 3 3200G con Gráficos Radeon V
13	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING
12	Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB
11	SSD XPG SX8200 Pro, 256GB, PCI Express, M.2
9	Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPE
9	SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0, M2
7	Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8
6	Tarjeta Madre MSI ATX B450 TOMAHAWK MAX, S-AM4, AM
5	Tarjeta de Video Gigabyte NVIDIA GeForce GT 1030,
4	Procesador Intel Core i5-9600K, S-1151, 3.70GHz, S

-----20 PRODUCTOS CON MAYOR BÚSQUEDAS-----	
#BUSQUEDAS	PRODUCTO
263	SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm
107	SSD Adata Ultimate SU800, 256GB, SATA III, 2.5'',
60	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING
55	Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-C
41	Procesador AMD Ryzen 3 3200G con Gráficos Radeon V
35	Logitech Audífonos Gamer G635 7.1, Alámbrico, 1.5
32	TV Monitor LED 24TL520S-PU 24, HD, Widescreen, HDM
31	Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8
30	Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Q
30	SSD XPG SX8200 Pro, 256GB, PCI Express, M.2
27	SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0, M2
25	Tarjeta Madre MSI ATX B450 TOMAHAWK MAX, S-AM4, AM
24	Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB
23	Tarjeta Madre ASRock Micro ATX B450M Steel Legend,
20	Procesador Intel Core i5-9600K, S-1151, 3.70GHz, S
15	Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPE
15	Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT ME
15	TCL Smart TV LED 55S425 54.6, 4K Ultra HD, Widescr
11	Tarjeta de Video Gigabyte NVIDIA GeForce GT 1030,
11	SSD Kingston UV500, 480GB, SATA III, mSATA

-----5 PRODUCTOS CON MENORES VENTAS POR CATEGORIA-----		
CATEGORIA	PRODUCTO	#VENTAS
audifonos	Logitech Audífonos Gamer G332, Alámbrico, 2 Metros	1
audifonos	Cougar Audífonos Gamer Phontum Essential, Alámbric	1
audifonos	HyperX Audífonos Gamer Cloud Flight para PC/PS4/PS	1
audifonos	Logitech Audífonos Gamer G635 7.1, Alámbrico, 1.5	2
bocinas	Logitech Bocinas para Computadora con Subwoofer G5	2
discos duros	SSD Crucial MX500, 1TB, SATA III, M.2	1
discos duros	SSD Western Digital WD Blue 3D NAND, 2TB, M.2	2
discos duros	Kit SSD Kingston KC600, 1TB, SATA III, 2.5, 7mm	3
discos duros	SSD Kingston UV500, 480GB, SATA III, mSATA	3
discos duros	SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0, M2	9
memorias usb	Kit Memoria RAM Corsair Dominator Platinum DDR4, 3	1
pantallas	TCL Smart TV LED 55S425 54.6, 4K Ultra HD, Widescr	1
pantallas	TV Monitor LED 24TL520S-PU 24, HD, Widescreen, HDM	1
procesadores	Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Quad-	2
procesadores	Procesador Intel Core i9-9900K, S-1151, 3.60GHz, 8	3
procesadores	Procesador Intel Core i5-9600K, S-1151, 3.70GHz, S	4
procesadores	Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8	7
procesadores	Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB	12
tarjetas de video	MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PCI Ex	1
tarjetas de video	Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti P	1
tarjetas de video	Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC	1
tarjetas de video	Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti,	1
tarjetas de video	Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT ME	2
tarjetas madre	Tarjeta Madre Gigabyte XL-ATX TRX40 Designare, S-s	1
tarjetas madre	Tarjeta Madre ASUS ATX PRIME Z390-A, S-1151, Intel	2
tarjetas madre	Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1	3
tarjetas madre	Tarjeta Madre MSI ATX B450 TOMAHAWK MAX, S-AM4, AM	6
tarjetas madre	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING	13

-----20 PRODUCTOS CON MENORES BUSQUEDAS-----	
#BUSQUEDAS	PRODUCTO
1	Procesador Intel Core i3-8100, S-1151, 3.60GHz, Qu
1	MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PCI Ex
1	Tarjeta de Video VisionTek AMD Radeon HD5450, 2GB
1	Tarjeta Madre Gigabyte micro ATX Z390 M GAMING, S-
1	Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, In
1	SSD Samsung 860 EVO, 1TB, SATA III, M.2
1	Samsung Smart TV LED 43, Full HD, Widescreen, Negr
1	Ghia Bocina Portátil BX800, Bluetooth, Inalámbrico
1	Ginga Audífonos con Micrófono GI18ADJ01BT-R0, Blue
2	Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti P
2	SSD para Servidor Lenovo Thinksystem S4500, 480GB,
2	Acteck Bocina con Subwoofer AXF-290, Bluetooth, In
2	Genius GHP-400S Audífonos, Alámbrico, 1.5 Metros,
3	Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2G
3	ASUS T. Madre uATX M4A88T-M, S-AM3, DDR3 para Phen
3	Iogear Audífonos Gamer GHG601, Alámbrico, 1.2 Metr
4	Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti S
4	Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2, S-1
4	Seiki TV LED SC-39HS950N 38.5, HD, Widescreen, Neg
4	Samsung Smart TV LED UN55TU7000FXZX 55, 4K Ultra H

-----10 PRODUCTOS CON LAS MEJORES RESEÑAS-----	
#SCORE	PRODUCTO
5.0	Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Quad-
5.0	Procesador Intel Core i9-9900K, S-1151, 3.60GHz, 8
5.0	Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8
5.0	Procesador Intel Core i5-9600K, S-1151, 3.70GHz, S
5.0	Tarjeta de Video ASUS AMD Radeon RX 570, 4GB 256-b
5.0	Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT ME
5.0	Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC
5.0	Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500
5.0	Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti,
5.0	Tarjeta Madre Gigabyte XL-ATX TRX40 Designare, S-s

-----10 PRODUCTOS CON LAS PEORES RESEÑAS-----	
#SCORE	PRODUCTO
1.0	Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, In
1.0	Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2G
1.83	Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1
2.0	Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2, S-1
3.0	Cougar Audífonos Gamer Phontum Essential, Alámbric
4.0	HyperX Audífonos Gamer Cloud Flight para PC/PS4/PS
4.0	MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PCI Ex
4.0	Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti P
4.14	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING
4.23	Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB

-----TOTAL INGRESOS MENSUALES-----		
#MES	INGRESO MENSUAL	
1	117738	
2	107270	
3	162931	
4	191066	
5	91936	
6	36949	
7	26949	
8	3077	
9	0	
10	0	
11	0	
12	0	

-VENTAS PROMEDIO MENSUALES-	
#MES	VENTAS PROMEDIO
1	2264.1923
2	2681.75
3	3325.1224
4	2581.973
5	2704.0
6	3359.0
7	2449.9091
8	1025.6667
9	0
10	0
11	0
12	0

-MESES CON MÁS VENTAS AL AÑO-	
#MES	VENTAS
4	74
1	52
3	49
2	40
5	34
6	11
7	11
8	3

-TOTAL ANUAL (2020)-	
\$737916	

Solución al problema

Como se explicó en la introducción la tienda virtual tiene ciertos problemas y tengo 3 soluciones que creo que puede ayudar a la tienda a sobrellevar algo estos problemas:

1. Dado que no se están buscando y por lo tanto no se venden ciertos productos lo que ayudaría es darlos a conocer más, ya que es una tienda virtual en la página principal podrían tener una pop-up que ayude a visualizar este tipo de productos o promocionarlos en las redes sociales.
2. Complementando el primer punto a la hora de mostrar más esos productos en las redes sociales o en la página principal se podrían dar descuentos, ya que si no están dando ingresos y la gente no los busca es más llamativo ver algo con descuento y para que no quede tanto en stock podría venderse algo para tener ganancias más rápido.
3. Como ultima estrategia podría decir que ya que tenemos este análisis checar bien esos productos que no están en las tablas y aparte de mejorar los que tienen peores reseñas podrían no comprar tanto de los productos que no tienen ventas y no tener un inventario lleno de algo que no se vende mucho.

Conclusión

Con este proyecto se han cumplido los objetivos que se mencionaron al comienzo del mismo, obteniendo resultados que los validan. Este proyecto trae consigo muchas más habilidades que solo las de un programador, también tuvimos que analizar cada una de los puntos antes y resolver los problemas que iban saliendo con cada parte.

Quiero añadir que este proyecto fue una buena manera de poner a prueba cada parte de lo visto en los cursos, pero también para cierto código y hacerlo más optimo se recurrió a buscar ciertas herramientas que nos ayudan a mejorar y teniendo en cuenta que apenas esta es una introducción me pareció que es algo bueno para empezar con el área de ciencia de datos.

Código Github

[Proyectos-Emtech/PROYECTO-01- BERNAL-ALEJANDRO.py at main · Alex-Bernal27/Proyectos-Emtech \(github.com\)](https://github.com/Alex-Bernal27/Proyectos-Emtech/blob/main/PROYECTO-01-BERNAL-ALEJANDRO.py)