

EMTECH

Reporte final LifeStore

Introducción a Python

Keila Rubí Puente González
13-9-2021

Índice

Introducción.....	2
Definición del código.....	2
Solución al problema.....	10
Conclusión.....	15
Código en GitHub	15

Introducción

A través de los años la tecnología ha ido evolucionando y con ello cada vez más personas tienen acceso a un aparato electrónico y a internet. Esto ha ocasionado un cambio en las industrias, en la manera de darse a conocer y de vender productos, por lo tanto, es importante para las empresas adaptarse a estos cambios para evitar quedarse rezagadas y como consecuencia tengan que salir del mercado.

LifeStore es una de las empresas que se ha adaptado a la época, es una tienda virtual con una amplia gama de productos electrónicos en venta, sin embargo, la gerencia le ha informado que hay un problema con el inventario y quieren conocer información detallada sobre las ventas y búsquedas de sus productos ya que, en el último trimestre sus ventas bajaron.

Por lo tanto, el presente reporte contiene el análisis para la elaboración de la propuesta de estrategia a la empresa. Se observará que dicho análisis comprende listas de los productos más vendidos y rezagados, asimismo las reseñas, que son las calificaciones que los clientes dan a los productos y finalmente ingresos mensuales, ingreso anual y meses con más ventas al año.

El código que se mostrará contiene los comentarios de la lógica que se siguió para la solución del problema.

Definición del código

```
##### LOGIN #####
#LA CONTRASEÑA ES 4412
```

```
'''al ejecutarse valida la contraseña, en caso de estar incorrecta dice que
se intente de nuevo, si es correcta entonces se ejecuta el código y se imprime
el menú, por eso el break, para que salga del ciclo.'''
```

```
while (True):
    entrar = int(input("Por favor, ingrese la contraseña: "))
    #validando la contraseña
    if entrar == 4412:
        print("\n\n Bienvenido, seleccione una opción")
        break
    else:
        print ("\n\n Contraseña incorrecta,intente de nuevo")
```

```

lifestore_products = [
    [1, 'Procesador AMD ...]

lifestore_sales = [
    [1, 1, 5, '24/07/2020', 0], ...]

lifestore_searches = [
    [1, 1], ...]

```

#guardando los productos e información primero

#####

''' se guardó de la siguiente manera dado que al intentar por la forma vista en tutoría no lo aceptó el programa

ventaspor_producto= [ID, # ventas, precio, categoría, nombre]

busquedas_producto= [ID, # busquedas, categoría, nombre]

'''

ventaspor_producto= [[producto[0],0,0,"s","s"] for producto in lifestore_products]

busquedas_producto=[[producto[0],0,"s","s"] for producto in lifestore_products]

#AGREGANDO LA CANTIDAD DE VENTAS

'''ventaspor_producto[ID, # ventas, precio, categoría, nombre]

aquí se contaron las ventas siendo que no hubo devolución, se fue agregando la cuenta de las ventas, el precio, la categoría y el nombre limitado a 45 caracteres'''

for venta in lifestore_sales:

if venta[1] == ventaspor_producto[venta[1]-1][0] and venta[4]!=1:

ventaspor_producto[venta[1]-1][1]+=1

ventaspor_producto[venta[1]-1][2] = lifestore_products[venta[1]-1][2]

ventaspor_producto[venta[1]-1][3] = lifestore_products[venta[1]-1][3]

ventaspor_producto[venta[1]-1][4] = lifestore_products[venta[1]-1][1][:45]

'''busquedas_producto[ID, # busquedas, categoría, nombre]

lo mismo que lo anterior se hace con las busquedas'''

for busqueda in lifestore_searches:

if busqueda[1] == busquedas_producto[busqueda[1]-1][0]:

busquedas_producto[busqueda[1]-1][1]+=1

busquedas_producto[busqueda[1]-1][2] = lifestore_products[busqueda[1]-1][3]

busquedas_producto[busqueda[1]-1][3] = lifestore_products[busqueda[1]-1][1][:45]

```
#####
```

```
'''ahora en una nueva lista se guardan las ventas que son >=1'''
```

```
ventas_producto = []
```

```
for i in ventaspor_producto:
    if i[1]!=0:
        ventas_producto.append(i)
```

```
'''lo siguiente es agregar a una lista solo la cantidad de ventas y el nombre
del producto para facilitar la impresión'''
```

```
ventas_producto2 = [[venta[1],venta[4]] for venta in ventas_producto]
```

```
'''lo mismo se hace con las busquedas, se eliminan los productos no buscados'''
```

```
busqueda_products = []
for i in busquedas_producto:
    if i[1]!=0:
        busqueda_products.append(i)
```

```
'''se hace lo mismo que con la lista ventas, agregar a una lista solo la cantidad de
busquedas y el nombre del producto para facilitar la impresión'''
```

```
busqueda_products2 = [[busq[1],busq[3]] for busq in busqueda_products]
```

```
#ORDENAR CON SORT E IMPRIMIR 15,20,5,20
```

```
''' ahora se crean las funciones para ordenar las listas de acuerdo con
la cantidad de ventas o búsquedas'''
```

```
def cantidades(v):
    return v[0]
```

```
#el parámetro reverse es booleano e inidica si se debe ordenar de manera descendente o no.
#el parámetro key es para indicar en base a qué lo ordenaremos.
```

```
'''se ordenan esas nuevas listas con solo la cantidad de ventas y búsquedas
con su respectivo nombre de producto'''
```

```
ventas_producto2.sort(reverse = True, key = cantidades)
busqueda_products2.sort(reverse = True, key =  cantidades)
```

```
'''aquí ya se asignan los 15 y 20 productos de ventas y búsquedas mayores
respectivamente, dado que están ordenados de manera descendente'''
```

```
ventas_mayores = ventas_producto2[:15]
busquedas_mayores = busqueda_products2[:20]
```

```
#####
# #CATEGORÍAS, 5 CON MENORES VENTAS Y 20 PRODUCTOS CON MENORES BUSQUEDAS

''' se hacen listas con la cantidad de ventas y búsquedas, la categoría y el nombre
del producto para facilitar la impresión posteriormente'''
ventas_concat = [[venta[1],venta[3],venta[4]] for venta in ventas_producto]
busquedas_concat = [[busq[1],busq[2],busq[3]] for busq in busqueda_products]

''' Se ordenan las listas anteriores de manera ascendente con la función definida
anteriormente para tomar los que tienen menores ventas y los 20 con menores búsquedas con
su categoría y nombres incluidos'''
ventas_concat.sort(reverse = False, key = cantidades)
busquedas_concat.sort(reverse = False, key = cantidades)

'''se asignan las cantidades correspondientes a las variables que guardan las ventas
y las búsquedas, tomando el complemento de las listas de las mayores, es decir,
el resto de las ventas menores'''
cate_v = ventas_concat
cate_b = busquedas_concat[:20]

'''se ordenan las listas anteriores primeramente por cantidad de ventas
después por orden alfabético de acuerdo con la categoría a la que pertenecen.'''
def orden(k):
    return k[1]

cate_v.sort(key= cantidades)
cate_v.sort(key= orden)

cate_b.sort(key= cantidades)
cate_b.sort(key= orden)

#####
#DOS LISTAS POR RESEÑAS, MEJORES Y PEORES, 10 C/U DESDE LA LISTA DE VENTAS
'''resenas = [ID, espacio para la suma de las reseñas, nombre]
cantidad = [ID, espacio para la suma de las cantidades de reseñas]
promedio_resenas = [ID, promedio, nombre]
la primera lista es para guardar el ID y el nombre e ir acumulando la suma de las
reseñas por cada producto, la segunda es para ir guardando la cantidad de veces que
se dio reseña de cada producto, y la última es para guardar el promedio'''
resenas = [[producto[0],0,producto[1][:45]] for producto in lifestore_products]
cantidad = [[producto[0],0] for producto in lifestore_products]
promedio_resenas = [[producto[0],0,producto[1]] for producto in lifestore_products]
```

```

#sumando las reseñas
for prod in lifestore_sales:
    if prod[1] == resenas[prod[1]-1][0]:
        resenas[prod[1]-1][1] = resenas[prod[1]-1][1] + prod[2]

#sumando la cantidad de reseñas
for produ in lifestore_sales:
    if produ[1] == cantidad[produ[1]-1][0]:
        cantidad[produ[1]-1][1] += 1

#sacando los promedios
for i in lifestore_sales:
    if i[1] == cantidad[i[1]-1][0] and cantidad[i[1]-1][1] != 0:
        promedio_resenas[i[1]-1][1] = (resenas[i[1]-1][1]) / (cantidad[i[1]-1][1])

'''utilizando la función orden para ordenar'''
promedio_resenas.sort(reverse = True, key = orden)
'''en una lista se guarda solo el promedio de reseña y el nombre del producto'''
prom_nombre = [[pro[1], pro[2]] for pro in promedio_resenas]

mejores_resenas = prom_nombre[:10]

'''en una nueva lista se guardan las reseñas sin contar las que están en cero'''
prl = []
#mientras no sea igual a cero se guarda en una nueva lista
for j in prom_nombre:
    if j[0] != 0:
        prl.append(j)

'''Para ordenar ascendente se utiliza la función cantidades ya que ordena de acuerdo
con el primer elemento de la lista'''
prl.sort(reverse = False, key = cantidades)
''' se asignan las peores reseñas a una lista'''
peores_resenas = prl[:10]

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

'''ven_mensuales[mes, total, #ventas, promedio]
se define la siguiente lista con el número de mes, los demás espacios son para guardar
la suma de las ventas según el precio de cada producto vendido por mes y el siguiente
espacio es para llevar una cuenta de las ventas por mes, el último es para guardar el
promedio'''
ven_mensuales = [[m, 0, 0, 0] for m in range(1, 13)]

```



```
'''llenando el total de acuerdo con el precio de cada producto vendido por mes y contando el número de ventas.
```

```
El primer for es para ir por cada lista en lifestore_sales, el siguiente para recorrer en cada lista de los meses'''
```

```
for ventaa in lifestore_sales:
    for mees in ven_mensuales:
        if int(ventaa[3][3:5]) == mees[0] and ventaa[4]==0:
            ven_mensuales[mees[0]-1][1]+= lifestore_products[ventaa[1]-1][2]
            ven_mensuales[mees[0]-1][2]+=1
```

```
'''venta promedio por mes'''
```

```
for i in ven_mensuales:
    if i[2]!=0:
        i[3]= i[1]/i[2]
```

```
'''se crea una lista que solo contenga el numero de mes, el total de ingresos y la venta promedio en el mes'''
```

```
totaling_vtaprom = [[m[0],m[1],m[3]] for m in ven_mensuales]
```

```
'''total anual es la suma del total por mes'''
```

```
ta=0
for i in ven_mensuales:
    ta+= i[1]
```

```
'''se ordena la lista de manera descendente, para esto primero se crea una lista que contenga solo los meses y la cantidad de ventas'''
```

```
meses_mayoresv = [[m[0],m[2]] for m in ven_mensuales]
```

```
'''se utiliza la función orden nuevamente para ordenar'''
```

```
meses_mayoresv.sort(reverse = True, key = orden)
```

```
'''ahora se define el menú para que esté apareciendo y si se selecciona una opción que no es parte de éste, muestra un mensaje de que intente nuevamente'''
```

```
while (True):
    menu = int(input(''''          MENÚ

    1. Productos con mayores ventas
    2. Productos con mayor cantidad de búsquedas
    3. Categorías de los productos con menores ventas
    4. Categorías de los productos con menores búsquedas
    5. Productos con mejores reseñas
    6. Productos con peores reseñas
    7. Total de ingresos y ventas promedio mensuales
    8. Total anual
    9. Meses con más ventas al año

    Opción:  '''
    ))
```


[illegible]

[illegible]

Solución al problema

Con el análisis llevado a cabo se llegó a los siguientes resultados:

■ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ LAS 15 VENTAS MAYORES ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ♦ ■	
CANTIDAD	NOMBRE PRODUCTO
49	SSD Kingston A400, 120GB, SATA III, 2.5'', 7m
42	Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz,
20	Procesador Intel Core i3-9100F, S-1151, 3.60G
18	Tarjeta Madre ASRock Micro ATX B450M Steel Le
15	SSD Adata Ultimate SU800, 256GB, SATA III, 2.
13	Procesador AMD Ryzen 3 3200G con Gráficos Rad
13	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS G
12	Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz,
11	SSD XPG SX8200 Pro, 256GB, PCI Express, M.2
9	Tarjeta de Video ASUS NVIDIA GeForce GTX 1660
9	SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0
7	Procesador Intel Core i7-9700K, S-1151, 3.60G
6	Tarjeta Madre MSI ATX B450 TOMAHAWK MAX, S-AM
5	Tarjeta de Video Gigabyte NVIDIA GeForce GT 1
4	Procesador Intel Core i5-9600K, S-1151, 3.70G
■ ♦ ■	

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

LOS 5 PRODUCTOS CON MENORES VENTAS		
CANTIDAD	CATEGORÍA	NOMBRE PRODUCTO
1	audifonos	Logitech Audifonos Gamer G332, Alámbrico, 2 M
1	audifonos	Cougar Audifonos Gamer Phontum Essential, Alá
1	audifonos	HyperX Audifonos Gamer Cloud Flight para PC/P
2	audifonos	Logitech Audifonos Gamer G635 7.1, Alámbrico,
2	bocinas	Logitech Bocinas para Computadora con Subwoof
1	discos duros	SSD Crucial MX500, 1TB, SATA III, M.2
2	discos duros	SSD Western Digital WD Blue 3D NAND, 2TB, M.2
3	discos duros	Kit SSD Kingston KC600, 1TB, SATA III, 2.5, 7
3	discos duros	SSD Kingston UV500, 480GB, SATA III, mSATA
9	discos duros	SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0
11	discos duros	SSD XPG SX8200 Pro, 256GB, PCI Express, M.2
15	discos duros	SSD Adata Ultimate SU800, 256GB, SATA III, 2.
49	discos duros	SSD Kingston A400, 120GB, SATA III, 2.5'', 7m
1	memorias usb	Kit Memoria RAM Corsair Dominator Platinum DD
1	pantallas	TCL Smart TV LED 55S425 54.6, 4K Ultra HD, Wi
1	pantallas	TV Monitor LED 24TL520S-PU 24, HD, Widescreen
2	procesadores	Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz,
3	procesadores	Procesador Intel Core i9-9900K, S-1151, 3.60G
4	procesadores	Procesador Intel Core i5-9600K, S-1151, 3.70G
7	procesadores	Procesador Intel Core i7-9700K, S-1151, 3.60G
12	procesadores	Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz,
13	procesadores	Procesador AMD Ryzen 3 3200G con Gráficos Rad
20	procesadores	Procesador Intel Core i3-9100F, S-1151, 3.60G
42	procesadores	Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz,
1	tarjetas de video	MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, P
1	tarjetas de video	Tarjeta de Video Asus NVIDIA GeForce GTX 1050
1	tarjetas de video	Tarjeta de Video MSI NVIDIA GeForce GTX 1050
1	tarjetas de video	Tarjeta de Video Zotac NVIDIA GeForce GTX 166
2	tarjetas de video	Tarjeta de Video MSI AMD Mech Radeon RX 5500
2	tarjetas de video	Tarjeta de Video Sapphire AMD Pulse Radeon RX
3	tarjetas de video	Tarjeta de Video ASUS AMD Radeon RX 570, 4GB
5	tarjetas de video	Tarjeta de Video Gigabyte NVIDIA GeForce GT 1
9	tarjetas de video	Tarjeta de Video ASUS NVIDIA GeForce GTX 1660
1	tarjetas madre	Tarjeta Madre Gigabyte XL-ATX TRX40 Designare
2	tarjetas madre	Tarjeta Madre ASUS ATX PRIME Z390-A, S-1151,
3	tarjetas madre	Tarjeta Madre AORUS micro ATX B450 AORUS M (r
6	tarjetas madre	Tarjeta Madre MSI ATX B450 TOMAHAWK MAX, S-AM
13	tarjetas madre	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS G
18	tarjetas madre	Tarjeta Madre ASRock Micro ATX B450M Steel Le

LOS 20 PRODUCTOS CON MENORES BÚSQUEDAS		
CANTIDAD	CATEGORÍA	NOMBRE PRODUCTO
1	audifonos	Ginga Audifonos con Micrófono GI18ADJ01BT-R0,
2	audifonos	Genius GHP-400S Audifonos, Alámbrico, 1.5 Met
3	audifonos	Iogear Audifonos Gamer GHG601, Alámbrico, 1.2
1	bocinas	Ghia Bocina Portátil BX800, Bluetooth, Inalám
2	bocinas	Acteck Bocina con Subwoofer AXF-290, Bluetooth
1	discos duros	SSD Samsung 860 EVO, 1TB, SATA III, M.2
2	discos duros	SSD para Servidor Lenovo Thinksystem S4500, 4
1	pantallas	Samsung Smart TV LED 43, Full HD, Widescreen,
4	pantallas	Seiki TV LED SC-39HS950N 38.5, HD, Widescreen
4	pantallas	Samsung Smart TV LED UN55TU7000FXZX 55, 4K U1
1	procesadores	Procesador Intel Core i3-8100, S-1151, 3.60GH
1	tarjetas de video	MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, P
1	tarjetas de video	Tarjeta de Video VisionTek AMD Radeon HD5450,
2	tarjetas de video	Tarjeta de Video Asus NVIDIA GeForce GTX 1050
3	tarjetas de video	Tarjeta de Video Gigabyte AMD Radeon R7 370 0
4	tarjetas de video	Tarjeta de Video EVGA NVIDIA GeForce GTX 1660
1	tarjetas madre	Tarjeta Madre Gigabyte micro ATX Z390 M GAMIN
1	tarjetas madre	Tarjeta Madre ASRock ATX H110 Pro BTC+, S-115
3	tarjetas madre	ASUS T. Madre uATX M4A88T-M, S-AM3, DDR3 para
4	tarjetas madre	Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2

LOS 10 PRODUCTOS CON MEJORES RESEÑAS	
RESEÑA PROMEDIO	NOMBRE PRODUCTO
5.0	Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Quad-Core, 16MB L2 Cache
5.0	Procesador Intel Core i9-9900K, S-1151, 3.60GHz, 8-Core, 16MB Smart Cache (9na. Generación Coffee Lake)
5.0	Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)
5.0	Procesador Intel Core i5-9600K, S-1151, 3.70GHz, Six-Core, 9MB Smart Cache (9na. Generación - Coffee Lake)
5.0	Tarjeta de Video ASUS AMD Radeon RX 570, 4GB 256-bit GDDR5, PCI Express 3.0
5.0	Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC, 8GB 128-bit GDDR6, PCI Express 4.0
5.0	Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC, 4GB 128-bit GDDR5, PCI Express x16 3.0
5.0	Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500 XT Gaming, 8GB 128-bit GDDR6, PCI Express 4.0
5.0	Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti, 6GB 192-bit GDDR6, PCI Express x16 3.0
5.0	Tarjeta Madre Gigabyte XL-ATX TRX40 Designare, S-sTRX4, AMD TRX40, 256GB DDR4 para AMD

LOS 10 PRODUCTOS CON PEORES RESEÑAS	
RESEÑA PROMEDIO	NOMBRE PRODUCTO
1.0	Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2GB 256-bit GDDR5, PCI Express 3.0
1.0	Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, Intel H110, 32GB DDR4, para Intel
1.8333333333333333	Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
2.0	Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2, S-1151, Intel H110, 32GB DDR4 para Intel
3.0	Cougar Audifonos Gamer Phontum Essential, Alámbrico, 1.9 Metros, 3.5mm, Negro.
4.0	MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PCI Express 2.0
4.0	Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix, 4GB 128-bit GDDR5, PCI Express 3.0
4.0	HyperX Audifonos Gamer Cloud Flight para PC/PS4/PS4 Pro, Inalámbrico, USB, 3.5mm, Negro
4.142857142857143	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
4.230769230769231	Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth

Los productos más vendidos están contenidos en los productos más buscados, por lo tanto, la empresa debe prestar atención a los productos que más buscan los clientes ya que son los que más se llevan, esto implica un constante monitoreo para mantener el inventario con los productos necesarios.

Además, con los productos que casi no se vendieron se pueden ver dos opciones, la primera es que se pueden aprovechar esas reseñas positivas que muchos de ellos tuvieron para hacerles mayor publicidad y que la gente los adquiera en LifeStore y no en otra tienda y continuar vendiendo dichos productos si empiezan a vender más o la segunda, aprovechar esas reseñas para venderlos y discontinuarlos, queda a elección de la empresa si ponerlos o no en oferta, en ambas opciones se puede agregar un descuento a dichos productos.

También se deben de tomar en cuenta las peores reseñas y lo más conveniente es discontinuar los productos con peores reseñas y menores ventas, éstos se pueden incluir como “regalo” en la compra de algún producto o ponerlos en oferta.

La información de un año es relevante para saber cómo actuar a corto plazo, sin embargo, para una mayor precisión se puede conseguir una base de datos histórica para analizar la tendencia de compras de los clientes, por ejemplo, analizar qué productos suelen venderse más en cada mes, de esta forma se tendría previsto cuáles son los productos que deben de tenerse de preferencia en stock por temporada y de igual manera analizar la tendencia de compra de los productos para ver cuáles ventas están bajando con el paso del tiempo y tratar de predecir cuáles productos dejarán de comprarse eventualmente para ya no tener mucha cantidad de estos en stock.

Tomando en cuenta lo anterior, se pueden mejorar las ventas para evitar caer en pérdidas e incluso cerrar el negocio.

Conclusión

Es importante que la empresa se mantenga informada sobre lo que el cliente busca y que no descuide el inventario, con la información obtenida puede optar por cualquiera de las posibles soluciones planteadas.

Es un buen momento para mejorar en las estrategias de publicidad o de organización para continuar en el mercado.

En cuanto al contenido del curso, es satisfactorio ver lo que se puede llegar a hacer con conocimientos básicos de Python, fue una maravillosa entrada al mundo de Data Science.

Código en GitHub

https://github.com/KeilaPuenteG/PROYECTOS_EMTECH/blob/main/PROYECTO-01-PUENTE-KEILA.py