

2020

PROYECTO 1: Introducción a Python



Neith López Bautista

Proyecto LifeStore

Santander Emtech

ÍNDICE

<i>Introducción</i>	P.2
<i>Definición del código</i>	P.4
- Productos más vendidos y productos rezagados.	P. 5
- Productos por reseña en el servicio.	P. 8
	P. 9
- Total de ingresos y ventas promedio mensuales, total anual y meses con más ventas al año.	
<i>Resultados</i>	P.11
<i>Solución al problema</i>	P. 15
<i>Conclusiones</i>	P.15
<i>Link GitHub</i>	P.15

INTRODUCCIÓN

El presente proyecto, tiene como objetivo aplicar los conocimientos adquiridos en el curso de “Programación en Python para análisis y clasificación de datos”.

Los datos que se usan pertenecen a una tienda virtual llamada “LifeStore”, cuyos productos se agrupan básicamente en ocho categorías: procesadores, tarjetas de video, tarjetas madre, discos duros, memorias usb, pantallas, bocinas, audífonos.

La problemática gira en torno a un incremento en el inventario de la empresa, por ello será necesario hacer un diagnóstico a nivel producto con la finalidad de obtener conclusiones puntuales.

Para ello, se mostraran los siguientes resultados:

- 1.** Productos más vendidos y productos rezagados: Listado con los 50 productos con mayores ventas, listado con los 96 productos con mayores búsquedas. Listado de las ventas por categorías.
- 2.** Productos por reseña en el servicio: Listado con los 20 productos con las mejores reseñas, listado con 20 productos con las peores reseñas.
- 3.** Total de ingresos y ventas promedio mensuales, total anual y meses con más ventas al año.

A continuación se podrá consultar el código ejecutado en Python de cada consigna, así como los resultados, posibles soluciones y conclusiones derivadas del análisis.

Definición del código.

Para realizar cada una de las consignas, primero se importaron las listas contenidas en el archivo LifeStore-SalesList, utilizando el siguiente código:

```
0
9  from lifestore file import lifestore_products, lifestore_sales,
   lifestore_searches
10
```

Cabe recordar que dichas listas contenían la siguiente información a considerar en lo subsiguiente:

```
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]
```

Posteriormente se definió a los administradores, quiénes con el usuario y contraseña predefinida, estarán habilitados para conocer la información disponible sobre la situación de ventas de la empresa.

En caso de no tener los datos correctos se imprimirá un mensaje con la frase “Datos no encontrados” y por tanto se le negará el acceso. Caso contrario, se le dará la bienvenida a LifeStore y posteriormente se desplegará un menú con los tres apartados disponibles:

1. **Productos más vendidos y productos rezagados:** Listado con los 50 productos con mayores ventas, listado con los 96 productos con mayores búsquedas. Listado de las ventas por categorías.
2. **Productos por reseña en el servicio:** Listado con los 20 productos con las mejores reseñas, listado con 20 productos con las peores reseñas.
3. **Total de ingresos y ventas promedio mensuales, total anual y meses con más ventas al año.**

Para ello se realizó el siguiente código, en el que es posible observar que primero se hicieron listas definiendo los usuarios y contraseñas, después mediante la función “input” se recabó la información del usuario, la cual se verificó mediante un bucle “for”; posteriormente con sentencias “if”, se

desplegaron las opciones del menú que el usuario-administrador eligió y mediante bucle “while”, se realizaron las consignas. Cabe mencionar que esta pantalla muestra de manera general el código para facilitar su apreciación, pero dentro de cada sentencia if, de la respectiva opción elegida, se escribió el código de cada listado particular, los cuales se mostraran en lo subsiguiente.

```
15  #A continuación se definirán los administradores, quiénes serán los únicos
    #habilitados para ver la información disponible sobre la situación de ventas de
    #la empresa.
16  usuarios_admon = [{"A", "5657"}, {"B", "7879"}]
17  #Se le pedirá al usuario que ingrese los datos de usuario y contraseña
18  usr_inpt = input("ingresa el usuario: ")
19  pss_inpt = input("Ingresa la contraseña: ")
20
21  #una vez obtenidos los datos se procederá a corroborar dichos datos con la
    #lista usuarios_admon, con la finalidad de permitirles o no el acceso.
22  admon_perm = 0
23
24  for usuario in usuarios_admon:
25      if usuario[0] == usr_inpt and usuario[1] == pss_inpt:
26          print("Bienvenido a LifeStore")
27          admon_perm = 1
28          break
29      else:
30          print("Datos no encontrados")
31          continue
32
33
34  if admon_perm == 1:
35
36      print("Tienes permisos para acceder a los reportes de ventas. \nPor favor
        #elige una de las siguientes opciones (A,B,C): \n \n 1.A: Productos más
        #vendidos y productos rezagados \n 2.B: Productos por reseña de servicio \n
        #3.C: Total de ingresos y ventas")
37
38      reporte_selec = input("Opción: ")
39      op_aceptada = 0
40      while op_aceptada != 1:
41          if reporte_selec == "A":
42              print("Seleccionaste la opción A")
43              op_aceptada = 1
44
45
46          elif reporte_selec == "B":
47              print("Seleccionaste la opción B")
48              op_aceptada = 1
49
50          elif reporte_selec == "C":
51              print("Seleccionaste la opción C")
52              op_aceptada = 1
53
54          else:
55              print("Opción no disponible")
56              reporte_selec = input("Selecciona una opción disponible: ")
57
58  #En caso de no tener el usuario y contraseñas correctas, no será posible ver
```

```

else:
    print("Acceso denegado, por favor corrobora tus credenciales con
    el equipo de LifeStore")

```

De manera particular, las tres consignas se realizaron de la siguiente manera:

1. Productos más vendidos y productos rezagados:

Este apartado se despliega si se elige del menú la opción A:

- Listado con los 50 productos con mayores ventas: se utiliza como bandera “contador”, se define la variable “total_sales” como vacía, en la se ira recogiendo la información de los bucles. Mediante el bucle for se hace la cuenta de cada venta por producto, esto a través de la iteración que se hace en las listas, luego los resultados se añaden a la lista que en un inicio estaba vacía.

```

42     reporte_selec = input ("Opción: ")
43     op_aceptada = 0
44     while op_aceptada != 1:
45         if reporte_selec == "A":
46             print("Seleccionaste la opción A")
47             op_aceptada = 1
48
49             #LISTA UNO: Lista con los 50 productos con mayores ventas.
50             contador = 0
51             total_sales= [] #En esta lista se pondrá la cuenta de cada
                    iteración. [[id1, contador], [id2, contador]]
52
53             for producto in lifestore_products:
54                 for venta in lifestore_sales:
55                     if producto [0]==venta[1]:
56                         contador +=1
57
58             lista_uno = [producto[0], producto [1], contador]
59             total_sales.append (lista_uno)
60             contador = 0
61
62             #for total in total_sales:
63                 #print("\nID producto:", total[0], "\n" , "Nombre: ", total
                    [1], "\n", "No. Ventas:", total[2]) # resultado, lista de
                    todos los productos, sin ordenar

```

Una vez obtenida la cuenta del número de ventas por productos, se ordenan de mayor a menor, mediante el bucle while, en el cual, se asume que la primer sublista es el número mayor, después se compara con el resto de valores de la lista, y si la segunda posición es mayor entonces se guarda en la variable “sales_ordenados” y la primera sublista se quita. Es decir que itera sobre todos los valores de ventas hasta lograr ordenarse.

Finalmente mediante la función “in range”, se imprimen los 50 productos con mayores ventas y se le da formato a la lista. En éste caso se muestra el ID, el nombre del producto y el número de ventas para cada uno.

```
65 #ordenar de mayor a menor según el número de ventas.
66 sales_ordenados= []
67
68 while total_sales:
69     mayor = total_sales [0][2]
70     lista_actual = total_sales[0]
71     for sal in total_sales:
72         if sal[2]> mayor:
73             mayor = sal[2]
74             lista_actual= sal
75     sales_ordenados.append(lista_actual)
76     total_sales.remove(lista_actual)
77
78     #imprimir la lista con los 50 productos más vendidos.
79     print
80     print ("-----")
81     print ("-----")
82     print ("\n", "TOP 50 PRODUCTOS MÁS VENDIDOS", "\n")
83     print ("-----")
84     print ("-----")
85
86     for indice in range (0,50):
87         print("\n", "ID más vendido: ", sales_ordenados[indice][0])
88         print ("Nombre",sales_ordenados[indice][1])
89         print ("No. ventas",sales_ordenados[indice][2])
```

- Listado con los 96 productos con mayores búsquedas.

Para éste listado, se realizó un proceso muy similar a la lista anterior, pero en ésta lista se comparó el número de búsquedas y no de ventas. A grandes rasgos, lo que se hizo fue tomar el primer ID e ir iterando sobre la lista de búsquedas, en cada coincidencia se iban sumando las

búsquedas.

```
91     contador = 0
92     total_searches= [] #En esta lista se pondrá la cuenta de cada iteración. [[id1, contador]
    , [id2, contador]]
93
94     for producto in lifestore_products:
95         for search in lifestore_searches:
96             if producto [0]==search[1]:
97                 contador +=1
98
99     lista_dos = [producto[0], producto [1], contador]
100    total_searches.append (lista_dos)
101    contador = 0
102
103    #for total in total_searches:
104        #print("\nID producto:", total[0], "No. búsquedas: ", total[2])
105    #ordenar de mayor a menor según el número de búsquedas.
106    searches_ordenados= []
107
108    while total_searches:
109        mayor = total_searches [0][2]
110        lista_actual = total_searches[0]
111        for sear in total_searches:
112            if sear[2]> mayor:
113                mayor = sear[2]
114                lista_actual= sear
115        searches_ordenados.append(lista_actual)
116        total_searches.remove(lista_actual)
117    #imprimir la lista con los 50 productos más vendidos.
118    #print (len(searches_ordenados))
```

Posteriormente se ordenaron las búsquedas totales de mayor a menor y se imprimieron los resultados mediante la función “in range” con un formato que facilitara su lectura.

```
119
120
121     print("-----")
122     print ("\\n", "      TOP 96 PRODUCTOS CON MAYORES BÚSQUEDAS      ", "\\n")
123     print("-----")
124
125     for indice in range (0,96):
126         print("\\n", "ID más buscado: ", searches_ordenados[indice][0])
127         print ("Nombre",searches_ordenados[indice][1])
128         print ("No. búsquedas",searches_ordenados[indice][2])
```

- **Listado por categoría**

Para realizar éste listado, primero se realizó un ciclo for con la finalidad de obtener el nombre de las categorías ('procesadores', 'tarjetas de video', 'tarjetas madre', 'discos duros', 'memorias usb', 'pantallas', 'bocinas', 'audifonos'). Posteriormente usando otro bucle for se contaron el número de ventas por categorías (misma lógica que se ha explicado con anterioridad).

Finalmente se imprimieron los resultados con la función “print”


```

#LISTA CON LOS PRODUCTOS MENOS VENDIDOS POR CATEGORÍA

#Obtener las categorías contenidas en el código

categorias=[]
categorias_contenidas= []
for categoria in lifestore_products:
    if categorias_contenidas == categoria[3]:
        continue
    else:
        categorias.append(categoria[3])
        categorias_contenidas=categoria[3]

print("Las categorías de los productos son: ", "\n", (categorias))

for categoria in categorias:
    cuenta_venta_categoria=0
    for producto in lifestore_products:
        for ventas in lifestore_sales:
            if ventas[1]==producto[0] and producto[3]==categoria:
                cuenta_venta_categoria+=1
print("-----")
print(" \n", "      LISTADO POR CATEGORÍAS DE PRODUCTOS      ", "\n")
print("-----")
print("Categoría: ", categoria, " - ", "No. ventas: ", cuenta_venta_categoria)

```

2. Productos por reseña en el servicio:

Listado con los 20 productos con las mejores reseñas, listado con 20 productos con las peores reseñas.

Los listados se despliegan si en el menú de opciones se elige la opción "B".

Mediante un bucle for se hace la cuenta de las reseñas por productos, recordando que éstas tienen un score del 1-5, siendo 5 el valor máximo, es así que con dicha calificación se obtiene un promedio, sumando los valores y dividiendo entre el número de reseñas.

```

#MENÚ 2: PRODUCTOS POR RESEÑA EN EL SERVICIO
elif reporte_selec == "B":
    print ("Seleccionaste la opción B")
    op_aceptada = 1
    #Mostrar un listado con las mejores reseñas y otra para las peores,
    considerando los productos con devolución.
    #Para obtener el listado, primero se deberá obtener el promedio por producto
    del score de las reseñas (0-5), en dónde 5 es la máxima calificación.
    contador = 0
    sum_score = 0
    mean_products = []
    for producto in lifestore_products:
        for venta in lifestore_sales:
            if producto[0] == venta[1]:
                contador += 1
                sum_score += venta[2]
            promedio = sum_score / contador
            lista_E = [producto[0], producto[1], promedio]
            mean_products.append(lista_E)
    #print(mean_products)

```

Posteriormente se ordenan de mayor a menor (y viceversa para la segunda lista, sólo se cambian los signos)

```

score_ordenados= []

while mean_products:
    mayor = mean_products [0][2]
    lista_actual = mean_products[0]
    for prom in mean_products:
        if prom [2]> mayor:
            mayor = prom [2]
            lista_actual= prom
    score_ordenados.append(lista_actual)
    mean_products.remove(lista_actual)
#imprimir la lista con los 50 productos más vendidos.
#print (len(score_ordenados))

```

Se imprimen los resultados de cada lista previamente ordenada de la forma deseada.

```

print("-----")
print ("\n","      TOP 20: PRODUCTOS CON MEJOR SCORE      ", "\n")
print("-----")
for indice in range (0,20):
    print("\n", "ID_Score: ", score_ordenados[indice][0])
    print ("Nombre",score_ordenados[indice][1])
    print ("Score promedio (alto): ",score_ordenados[indice][2])

```

```

print("-----")
print ("\n","      20 PRODUCTOS CON PEOR SCORE      ", "\n")
print("-----")
for indice in range (0,20):
    print("\n", "ID: ", score_bajos[indice][0])
    print ("Nombre",score_bajos[indice][1])
    print ("Score promedio (bajo): ",score_bajos[indice][2])

```

3. Total de ingresos anuales

Para obtener el total de ingresos anuales, se realizó el siguiente código:

```

#Menú ingresos:
#Se retoma la lista total_sales del menú A, ya que contiene una lista con el
número de ventas por productos, el cuál se deberá multiplicar por el precio
contenido en la lista lifestore_products
contador = 0
total_sales= [] #En esta lista se pondrá la cuenta de cada iteración. [[id1,
contador], [id2, contador]]

for producto in lifestore_products:
    for venta in lifestore_sales:
        if producto [0]==venta[1]:
            contador +=1

    lista_uno = [producto[0], producto [1], contador]
    total_sales.append (lista_uno)
    contador = 0

ingresos_totales = 0
ingre_prod = []
for venta in total_sales:
    for producto in lifestore_products:
        if venta[0] == producto[0]:
            producto_de_ventas = int(venta[2]*producto[2])
            ingresos_totales += producto_de_ventas
print('El ingreso total es de: ', ingresos_totales, 'MXN')

```

Se retomó parte del código del primer menú, en el cuál se contabilizó el número de ventas mediante el bucle for (explicado en la primera sección), posteriormente las ventas se multiplicaron por el precio para obtener la suma final del año 2020. Éste último paso se hizo poniendo como bandera los ingresos_ totales, se definió la variable ingre_prod como vacía, con la finalidad de resguardar los resultados de cada iteración del bucle for, y mediante sentencias if (dentro del bucle), si los ID coincidían entonces se multiplicaba el número de ventas por el precio, dando como resultado que el “El ingreso total en 2020 es de: 760,177 MXN”

Resultados

A continuación se mostrarán parcialmente las listas resultantes, si se desea conocer el contenido completo, por favor ejecute el código.

1. Productos más vendidos y productos rezagados:

- Lista con los 50 productos más vendidos

Los resultados arrojaron que a partir del producto número 43 no se realizaron ventas, es decir 42 de 96 productos hicieron por lo menos una venta, es decir 43% sí vendió mientras que el restante 57% no materializó ventas. Además se observa que los primeros lugares fueron productos como procesadores y tarjetas de memoria principalmente.

ID	PRODUCTO	NO. VENTAS
1	SSD Kingston A400, 120GB, SATA III, 2.5", 7mm	50
2	Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth	42
3	Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)	20
4	Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD	18
5	SSD Adata Ultimate SU800, 256GB, SATA III, 2.5", 7mm	15
6	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD	14
7	Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth	13
8	Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire	13

- Lista con los productos con mayores búsquedas:

De 96 productos, 56 de ellos registraron por lo menos una búsqueda, esto equivale a decir que 41% de los artículos disponibles no fueron interesantes para el público. Además los resultados resaltan que el líder en búsquedas es una SSD Kingston, lo cuál podría ser un punto a explotar.

	NOMBRE	NO. BUSQUEDA
1	SSD Kingston A400, 120GB, SATA III, 2.5", 7mm	263
2	SSD Adata Ultimate SU800, 256GB, SATA III, 2.5", 7mm	107
3	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD	60

4	Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth	55
5	Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire	41
6	Logitech Audífonos Gamer G635 7.1, Alámbrico, 1.5 Metros, 3.5mm, Negro/Azul	35
7	TV Monitor LED 24TL520S-PU 24, HD, Widescreen, HDMI, Negro	32
8	Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)	31
9	Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)	30
10	SSD XPG SX8200 Pro, 256GB, PCI Express, M.2	30

- Listado de ventas por categorías

Los resultados arrojan que el número de ventas se concentró en procesadores con 104 ventas, seguido de discos duros y tarjetas madre, mientras que las pantallas, bocinas y memorias usb registraron los menores niveles.

Categoría	Venta
Procesadores	104
Discos duros	94
Tarjetas madre	49
Tarjeta de vídeo	26
Pantallas	2
Bocinas	2
Memorias USB	1

2. Productos por reseña en el servicio:

- Listado con los 20 productos con las mejores reseñas

Como ya se explicó, para enlistar se realizó el promedio del score, derivado de ello se observa que los 20 mejores rankeados reportaron un promedio que oscilo entre 4.6 y 5. Resalta que un procesador AMD reportó una calificación perfecta.

	NOMBRE	PROMEDIO_SCORE
1	Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Quad-Core, 16MB L2 Cache	5.0
2	Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER	

	EVO OC, 6GB 192-bit GDDR6, PCI Express x16 3.0	4.7094017094017095
3	Tarjeta de Video ASUS AMD Radeon RX 570, 4GB 256-bit GDDR5, PCI Express 3.0	4.703703703703703
4	Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix, 4GB 128-bit GDDR5, PCI Express 3.0	4.703389830508475
5	Tarjeta de Video EVGA NVIDIA GeForce GT 710, 2GB 64-bit GDDR3, PCI Express 2.0	4.703389830508475
6	Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming, 6GB 192-bit GDDR6, PCI 3.0	4.703389830508475
7	Tarjeta de Video EVGA NVIDIA GeForce RTX 2060 SC ULTRA Gaming, 6GB 192-bit GDDR6, PCI Express 3.0	4.703389830508475
8	Procesador Intel Core i5-9600K, S-1151, 3.70GHz, Six-Core, 9MB Smart Cache (9na. Generación - Coffee Lake)	4.701923076923077
9	Procesador Intel Core i3-8100, S-1151, 3.60GHz, Quad-Core, 6MB Smart Cache (8va. Generación - Coffee Lake)	4.701923076923077
10	MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PCI Express 2.0	4.695238095238095
11	Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)	4.69
12	Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth	4.684210526315789
13	Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti, 6GB 192-bit GDDR6, PCI Express x16 3.0	4.676923076923077
14	Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500 XT Gaming, 8GB 128-bit GDDR6, PCI Express 4.0	4.674418604651163
15	Tarjeta de Video VisionTek AMD Radeon HD 5450, 1GB DDR3, PCI Express x16 2.1	4.674418604651163
16	Tarjeta de Video VisionTek AMD Radeon HD5450, 2GB GDDR3, PCI Express x16	4.674418604651163
17	Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2GB 256-bit GDDR5, PCI Express 3.0	4.6722689075630255
18	Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC, 4GB 128-bit GDDR5, PCI Express x16 3.0	4.669291338582677
19	Tarjeta de Video MSI Radeon X1550, 128MB 64 bit GDDR2, PCI Express x16	4.669291338582677
20	Tarjeta de Video PNY NVIDIA GeForce RTX 2080, 8GB 256-bit GDDR6, PCI Express 3.0	4.669291338582677

- listado con 20 productos con las peores reseñas.

El listado con los 20 peores scores contiene productos con valor de entre 4.3 y 4.51, siendo el peor un procesador. Sin embargo destaca que pese a tener los valor más bajos, la calificación sigue siendo bastante acercada a la máxima que es 5.

	NOMBRE	PROMEDIO_SCORE
1	Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth	4.333333333333333
2	Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2, S-1151, Intel H110, 32GB DDR4 para Intel	4.4916201117318435
3	SSD XPG SX8200 Pro, 256GB, PCI Express, M.2	4.494736842105263
4	SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0, M2	4.50251256281407
5	Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, Intel H110, 32GB DDR4, para Intel	4.50561797752809
6	Kit SSD Kingston KC600, 1TB, SATA III, 2.5, 7mm	4.50990099009901
7	SSD Crucial MX500, 1TB, SATA III, M.2	4.512315270935961
8	Tarjeta Madre ASUS ATX PRIME Z390-A, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel	4.5131578947368425
9	Tarjeta Madre ASUS ATX ROG STRIX B550-F GAMING WI-FI, S-AM4, AMD B550, HDMI, max. 128GB DDR4 para AMD	4.5131578947368425
10	Tarjeta Madre Gigabyte micro ATX Z390 M GAMING, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel	4.5131578947368425
11	Tarjeta Madre Gigabyte micro ATX Z490M GAMING X (rev. 1.0), Intel Z490, HDMI, 128GB DDR4 para Intel	4.5131578947368425
12	Tarjeta Madre ASRock ATX Z490 STEEL LEGEND, S-1200, Intel Z490, HDMI, 128GB DDR4 para Intel	4.5131578947368425
13	Tarjeta Madre Gigabyte Micro ATX H310M DS2 2.0, S-1151, Intel H310, 32GB DDR4 para Intel	4.5131578947368425
14	ASUS T. Madre uATX M4A88T-M, S-AM3, DDR3 para Phenom II/Athlon II/Sempron 100	4.5131578947368425
15	Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD	4.513333333333334
16	Tarjeta Madre ASRock Z390 Phantom Gaming 4, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel	4.513333333333334
17	SSD Kingston UV500, 480GB, SATA III, mSATA	4.514563106796117
18	Tarjeta Madre Gigabyte XL-ATX TRX40 Designare, S-sTRX4, AMD TRX40, 256GB DDR4 para AMD	4.516339869281046
19	Tarjeta Madre ASUS micro ATX Prime H370M-Plus/CSM, S-1151, Intel H370, HDMI, 64GB DDR4 para Intel	4.516339869281046
20	SSD Western Digital WD Blue 3D NAND, 2TB, M.2	4.51923077

3. Total de ingresos : El ingreso total en 2020 es de: 760,177 MXN”

Solución al problema

El panorama general, da cuenta que los productos con mayores búsquedas, mejores reseñas coinciden con los productos con mayores ventas. Esto quiere decir por una parte que los productos que no registraron búsquedas ni ventas, quizá deberían de mostrarse cerca de los artículos vendidos, es posible que el público no los conozca.

También se podrían hacer paquetes de promoción con productos que no se venden, para familiarizar al público.

Además de aumentar la cartera de productos en aquellas categorías que evidentemente tienen ventas, ya que la probabilidad de que se vendan es alta.

Conclusiones

El análisis de datos en Python es una herramienta poderosa para poder explotar los datos y llegar a conclusiones certeras y bastante específicas, éste estudio de caso fue un ejemplo de ello, ya que se cruzaron varias listas para poder obtener los resultados.

Además en las soluciones se observa que aunque existen productos sin vender es posible mejorar las estrategias de marketing así como mejores elecciones de productos para solucionar su problema en el corto plazo.

Respecto a los ingresos, se sugiere mostrar datos de años previos, ya que sin ellos es complicado decir arbitrariamente si sus ingresos mejoraron o empeoraron, es necesario un punto de comparación.

Link GitHub:

<https://github.com/NeithLB/Emtech/blob/master/PROYECTO-01-LOPEZ-NEITH.py>