



REPORTE

Análisis y clasificación de datos con Python

por

Ricardo Ramos-Aguilar

6 de septiembre del 2020

Supervisor

Javier Ramírez

Índice general

List of Figures	IX
1. Capítulo 1	1
1.1. Introducción	1
2. Definición del código	3
2.1. Código	4
3. Resultados	9
3.1. Solución al problema	13
Bibliografía	14

Índice de figuras

2.1. Flujo de trabajo del algoritmo	3
2.2. Segmento de código 1	4
2.3. Segmento de código 2	5
2.4. Segmento de código 3	6
2.5. Segmento de código 4	7
2.6. Segmento de código 5	8
3.1. Resultados de la opción 1.	10
3.2. Otros resultados de la opción 1.	11
3.3. Resultados de la opción 2.	12
3.4. Resultados de la opción1	12

1 — Capítulo 1

1.1 Introducción

En la vida cotidiana desempeñamos eventos, acontecimientos o actividades en las que usamos mucha información. La apertura de un crédito en el banco, lleva previamente un análisis de información nuestra para la aprobación de este; la inscripción a una escuela consiste de un proceso en el que se recaba diferente información como nombre, dirección, estudios previos, entre otras cosas; para obtener un credencial o una licencia de conducir, necesitamos de ciertos requisitos y documentos que cotejen que se cumple con lo solicitado.

Las actividades mencionadas en el párrafo anterior describen el manejo de información, sin embargo, se sabe ¿Qué es información?, ¿Realmente ellos recaban información?. Es probable que en los eventos mencionados si se haga una recolección de información, ya que previamente expertos sugieren que esos son aspectos importantes para inscribir a alguien en la escuela, aprobar una licencia de conducir o aceptar la apertura de un crédito. Donde las decisión se toman a partir de lo que llamamos información; a este concepto comúnmente se le conoce como un conjunto de datos con algún significado que estructura el pensamiento de los seres vivos [1]. Por lo que, a partir de los datos puede obtenerse información.

En la actualidad, con el uso de Internet la creación de datos es inmensa. Estos pueden ser de tipo numéricos como transacciones bancarias, sonidos, imágenes, videos, entre otros. De todo esto y más se puede obtener información útil en diferentes aplicaciones; lo cual es realizado por profesionistas llamados científicos de datos, encargados de obtener información los datos que manejemos en Internet. Con objetivos basados en la toma de decisiones en general, que van desde la ayu-

da para la selección de una serie en Netflix hasta ver el sugerir aproximaciones del comportamiento del Covid-19. Las aplicaciones son muy extensas y diversas, y para lograr esto se hace una combinación de de habilidades como estadística, informáticas, conocimiento empresarias, entre otras [2].

En este reporte se presenta un análisis fundamental de datos en una tienda virtual; en el que a partir de una base de datos se sugieren listar los productos más vendidos, más buscados, con las peores reseñas y los ingresos obtenidos. Todo esto realizado con el lenguaje de programación Python, el cual es muy útil con objetivos de este tipo.

La organización del documento es la siguiente: en el capítulo siguiente se describe el algoritmo usado para los requerimientos; posteriormente, se muestran los resultados y se propone una solución; finalmente, se muestran las conclusiones.

2 — Definición del código

La realización del algoritmo sigue ciertos paso mostrados en la figura 2.1. Inicialmente, a partir de un conjunto de datos que contiene tres tablas: `lifestore_searches`, `lifestore_sales` y `lifestore_products`, las cuales son usadas para la generación de los requerimientos. Antes de entrar a la ejecución de pide el acceso con un Login de usuario y contraseña; si son los correctos o los dados el alta se muestra un menú con tres opciones: 1. Productos más vendidos y productos rezagados, 2. Productos por reseña en el servicio, 3. Total de ingresos y ventas promedio mensuales, total anual y meses con más ventas al año y 0. Salir. Con la selección de alguno de estos se muestra lo requerido, con la impresión de listas. El conjunto de datos contiene tres tablas con los siguientes atributos: `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)] y `lifestore_products` = [id_product, name, price, category, stock].

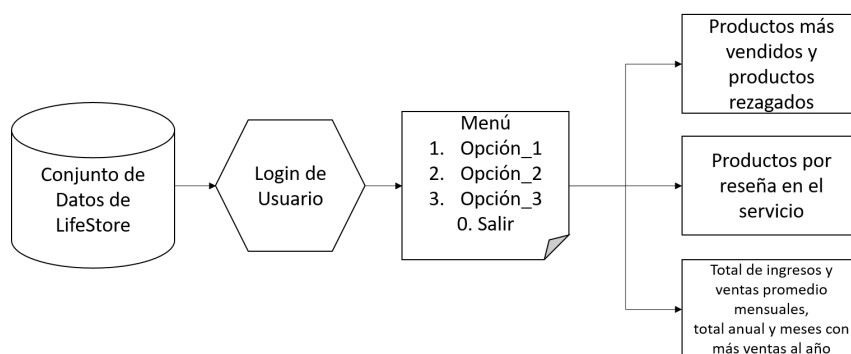


Figura 2.1: Flujo de trabajo del algoritmo

En esta primera parte (ver figura 2.2) se crea una variable llamada *opcm* que es ocupada en la opción de un menú, debido a que el usuario pueda elegir que visualizar. Para los comentarios, diferentes secciones se agruparon con la nomenclatura *P#* donde *P* representa una parte de código y *#* el número de parte. En esta parte nombrada **Registro y solicitud de usuarios (P0)** se hace la validación de *Login*, dando previamente de alta dos usuario con sus contraseñas, las cuales están relacionadas por el índice de las listas, es decir, el usuario *Ricardo* tiene la contraseña *hola123*. Posteriormente, se le solicita al usuario escribir el usuario y contraseña para validar el acceso. Si el usuario no existe o la contraseña lo menciona individualmente, por medio de *if's* anidados, esta parte se ve en la figura ??.

Figura 2.2: Segmento de código 1

La figura 2.3 muestra un parte del menú y la otra correspondiente a la declaración de variables, esto no se hizo en un principio debido a que no se recomienda usar variables que no se fueran a usar, y eso va a depender de la opción seleccionada.

```
##### Esta sección empieza con un menu de lo que se visualizara a partir de los 3 requerimientos (P2)#####
while(opcm!=0):
    print('_____')
    print('      Menu')
    print('_____')
    print('1. Productos más vendidos y productos rezagados')
    print('2. Productos por reseña en el servicio')
    print('3. Totales de ingresos y ventas')
    print('0. Salir')
    opcm=int(input('Teclee la opción: '))

##### Opción 1 Productos más vendidos y productos rezagados (P4) #####
if opcm==1:
    #Declaración de variables ### se hace aquí debido a que están en un menu

    lVentas=[] #Lista de productos vendidos
    fVentas={} #Diccionario para el conteo de ventas por producto
    l50pmv=[] #Lista de 50 productos mas vendidos. PUNTO 1.1.1

    lBuscados=[] #Lista de todos los productos buscados
    fBuscados={} #Diccionario para el conteo de búsquedas por producto
    l100pmb=[] #Lista de 50 productos mas vendidos. PUNTO 1.1.2

    iMvendidos=[] #índice de menos vendidos
    iMbuscados=[] #índice de menos buscado
    l1Mpuntuado=[] #Lista de listas de los más puntuados
```

Figura 2.3: Segmento de código 2

En el segmento de código mostrado en la figura 2.4 tiene como finalidad mostrar los 50 productos con mayores ventas y los 100 con mayores búsquedas. Para la primer parte se hizo un conteo de frecuencias con un diccionario a partir de los ids de producto, de la tabla `lifestore_sales`, con esto se sabía que productos se habían vendido y que cantidad. Posteriormente, se ordenaron y con el índice se buscó el producto para imprimir su nombre y la cantidad que de productos vendida.

La la impresión de los 100 más buscados, el proceso fue similar sin embargo, aquí se usó la tabla `lifestore_searches`.

```
##### LOS MÁS VENDIDOS (P7) #####
for v in lifestore_sales: #generación de lista de ventas
    lVentas.append(v[1])

for p in lVentas: #conteo de la cantidad vendida por producto
    if (p in fVentas):
        fVentas[p] += 1
    else:
        fVentas[p] = 1

mVendidos = sorted(fVentas.items(), reverse=True, key=lambda x: x[1]) #más vendidos
print("\n\nPRODUCTOS MÁS VENDIDOS ORDENADOS DESCENDENTEMENTE")

for mV in mVendidos:
    pTemporal=mV[0] #buscar indice
    producto=lifestore_products[pTemporal-1] #buscar producto
    po=producto[1] #usar solo el nombre
    l50pmv.append([po[0:20],mV[1]])
print("Producto          Cantidad vendida")
for p in l50pmv:
    print(p[0], "          ", p[1])

##### FIN DE SECCIÓN (P7) #####

##### LOS 100 MÁS BUSCADO (P8) #####
for b in lifestore_searches: #generación de lista de búsquedas
    lBuscados.append(b[1])

for p in lBuscados: #conteo de la cantidad de búsqueda por producto
    if (p in fBuscados):
        fBuscados[p] += 1
    else:
        fBuscados[p] = 1

mBuscados = sorted(fBuscados.items(), reverse=True, key=lambda x: x[1]) #más buscados
print("\n\nPRODUCTOS MÁS BUSCADOS ORDENADOS DESCENDENTEMENTE")

for mB in mBuscados:
    pTemporal=mB[0] #buscar indice
    producto=lifestore_products[pTemporal-1] #buscar producto
    po=producto[1] #usar solo el nombre
    l100pmb.append([po[0:20],mB[1]]) #juntar ambos para imprimirlos
print("Producto          Cantidad de veces buscado")
for p in l100pmb:
    print(p[0], "          ", p[1])
```

Figura 2.4: Segmento de código 3

En la sección de los menos vendidos y de menos búsquedas, mostrado en la figura 2.5, el proceso fue similar a lo anterior. La diferencia radica en que lo no vendido es el complemento de lo vendido, por lo que para ordenar se utilizó el operador *notin* para obtener lo excluido, finalmente, se ordena en sentido contrario para visualizar los resultados de menor a mayor.

```

##### FIN DE SECCIÓN (P8) #####

##### LOS MENOS VENDIDOS (P9) #####
for mV in mVendidos:
    pTemporal=mV[0] #buscar indice
    iMvendidos.append(pTemporal-1) #Lista de Los indices de Los producos mas vendidos

menosVendidos= [i for j, i in enumerate(lifestore_products) if j not in iMvendidos] #eliminando de La Lista Los más vendidos

print("\n\nPRODUCTOS MENOS VENDIDOS")
print("Producto", "Categoria")
for p in menosVendidos:
    print(p[1][0:30], " ", p[3])

##### FIN DE SECCIÓN (P9) #####

##### LOS MENOS BUSCADOS (P10) #####
for mB in mBuscados:
    pTemporal=mB[0] #buscar indice
    iMbuscados.append(pTemporal-1) #Lista de Los indices de Los producos mas vendidos

menosBuscados= [i for j, i in enumerate(lifestore_products) if j not in iMbuscados] #eliminando de La Lista Los más vendidos

print("\n\nPRODUCTOS MENOS BUSCADOS")
print("Producto", "Categoria")
for p in menosBuscados:
    print(p[1][0:30], " ", p[3])

##### FIN DE SECCIÓN (P10) #####

##### FIN DE SECCIÓN (P4) #####

```

Figura 2.5: Segmento de código 4

En la opción 2, la cual muestra los productos por reseña de servicio se ocuparon nuevamente diccionarios, y orden, como se ve en la figura 2.6. Durante el uso de los diccionarios se hicieron sumas de los puntajes o reseñas y del número de ventas. El conteo de las ventas funciono como auxiliar para calcular el promedio de las reseñas. En esta parte se piden los primeros 20, por lo que se aplico un break dentro del for.

```
##### FIN DE SECCION (P4) #####

##### Opción 2 Productos por reseña en el servicios (P5) #####
elif opcm==2:
    lVentasRef=[] #lista de productos vendidos y refund
    fScores={} #Diccionario para sumar las reseñas(scores)
    fSpromedio={} #Diccionario con scores promedio por idproductos
    fVentas={} #Frecuencia de ventas
    ##### LOS MÁS VENDIDOS (P7.1) #####
    for v in lifestore_sales: #generación de lista de ventas y refunds
        lVentasRef.append([v[1],v[2]])

    for p,s in lVentasRef: #conteo de la cantidad vendida por producto y su score
        if (p in fVentas):
            fVentas[p] += 1 #numero de ventas
            fScores[p]+=s #suma de puntajes
        else:
            fVentas[p] = 1
            fScores[p]=s
    ##### FIN DE SECCIÓN (P.7.1) #####
    ##### LOS MÁS PUNTUADO (P11) #####
    for vvv in fScores:
        fSpromedio[vvv]=fScores[vvv]/fVentas[vvv] #se hace diccionario con el promedio de lo puntutado
                                                #con su indice de producto

    mPuntuados = sorted(fSpromedio.items() , reverse=True, key=lambda x: x[1]) #más puntuados ordenados
    menosPuntuados = sorted(fSpromedio.items() , key=lambda x: x[1]) #más puntuados ordenados

    print("\n\n20 PRODUCTOS MEJOR PUNTEADOS DESCENDENTEMENTE")
    print("Producto Puntaje #productos")
    p20=1 #primero 20
    for mP in mPuntuados:
        pTemporal=mP[0] #buscar indice
        producto=lifestore_products[pTemporal-1] #"buscar producto"
        po=producto[1] #usar solo el nombre
        print(po[0:20],",", "{:.2f}".format(mP[1]),",", fVentas[mP[0]])
        p20=p20+1
        if p20==21:
            break
    print("\n\n20 PRODUCTOS PEOR PUNTEADOS ASCENDENTEMENTE")
    print("Producto Puntaje #productos")
    p20=1 #primero 20
    for mP in menosPuntuados:
        pTemporal=mP[0] #buscar indice
        producto=lifestore_products[pTemporal-1] #"buscar producto"
        po=producto[1] #usar solo el nombre
        print(po[0:20],",", "{:.2f}".format(mP[1]),",", fVentas[mP[0]])
        p20=p20+1
        if p20==21:
            break
    ##### FIN DE SECCIÓN (P11) #####
##### FIN DE SECCIÓN (P5) #####
```

Figura 2.6: Segmento de código 5

La parte final, en la que se muestran los ingresos (ver figura ??), sigue una metodología similar a lo anterior, con la diferencia de que se aplica un diccionario que se compara con los meses. Estos se usan como conteo de frecuencias y se suman las cantidades, se promedian.

3 — Resultados

De lo implementado se obtuvieron diferentes tablas, las cuales fueron copiadas a unas hojas de calculo, con la finalidad de tener una mejor visualización.

De la figura 3.1 se pedía generar un listado de los 50 productos con mayores ventas y uno con los 100 productos con mayor búsquedas. Sin embargo, aquí existió un error en los requerimientos debido a solo existen 96 productos y se pidieron 96, por otra parte se solicitan los 50 productos con mayores ventas, y solo son 42. La table de la derecha muestra los productos más buscado lo cual refleja que existe dependencia de lo buscado con lo más comprado.

3 – Resultados

PRODUCTOS MÁS VENDIDOS ORDENADOS DESCENDENTEMENTE			
Producto	Cantidad vendida		
SSD Kingston A400 1	50		
Procesador AMD Ryzen	42		
Procesador Intel Cor	20		
Tarjeta Madre ASRock	18		
SSD Adata Ultimate S	15		
Tarjeta Madre ASUS m	14		
Procesador AMD Ryzen	13		
Procesador AMD Ryzen	13		
SSD XPG SX8200 Pro	11		
Tarjeta de Video ASU	9		
SSD Kingston A2000 N	9		
Procesador Intel Cor	7		
Tarjeta Madre AORUS	6		
Tarjeta Madre MSI AT	6		
Tarjeta de Video Gig	5		
Procesador Intel Cor	4		
Procesador Intel Cor	3		
Tarjeta de Video ASU	3		
Kit SSD Kingston KC6	3		
SSD Kingston UV500	3		
Procesador AMD Ryzen	2		
Tarjeta de Video MSI	2		
Tarjeta de Video Sap	2		
Tarjeta Madre ASUS A	2		
SSD Western Digital	2		
Logitech Bocinas par	2		
Logitech Audifonos G	2		
MSI GeForce Z10 1GB	1		
Tarjeta de Video Asu	1		
Tarjeta de Video Gig	1		
Tarjeta de Video MSI	1		
Tarjeta de Video Zot	1		
Tarjeta Madre Gigaby	1		
Tarjeta Madre ASRock	1		
Tarjeta Madre Gigaby	1		
SSD Crucial MX500 1	1		
Kit Memoria RAM Cors	1		
TCL Smart TV LED 55S	1		
TV Monitor LED 24TL5	1		
Logitech Audifonos G	1		
Cougar Audifonos Gam	1		
HyperX Audifonos Gam	1		

PRODUCTOS MÁS BUSCADOS ORDENADOS DESCENDENTEMENTE			
Producto	Cantidad de veces buscado		
SSD Kingston A400 1	263		
SSD Adata Ultimate S	107		
Tarjeta Madre ASUS m	60		
Procesador AMD Ryzen	55		
Procesador AMD Ryzen	41		
Logitech Audifonos G	35		
TV Monitor LED 24TL5	32		
Procesador Intel Cor	31		
Procesador Intel Cor	30		
SSD XPG SX8200 Pro	30		
SSD Kingston A2000 N	27		
Tarjeta Madre MSI AT	25		
Procesador AMD Ryzen	24		
Tarjeta Madre ASRock	23		
Procesador Intel Cor	20		
Tarjeta de Video ASU	15		
Tarjeta de Video MSI	15		
TCL Smart TV LED 55S	15		
Tarjeta de Video Gig	11		
SSD Kingston UV500	11		
Procesador AMD Ryzen	10		
Procesador Intel Cor	10		
Tarjeta de Video Sap	10		
Tarjeta Madre AORUS	10		
Tarjeta Madre Gigaby	10		
Kit SSD Kingston KC6	10		
Logitech Audifonos G	10		
SSD Crucial MX500 1	7		
Cougar Audifonos Gam	7		
Logitech Bocinas par	6		
HyperX Audifonos Gam	6		
Tarjeta de Video ASU	5		
Tarjeta de Video MSI	5		
Tarjeta de Video Vis	5		
Tarjeta de Video Zot	5		
SSD Western Digital	5		
Tarjeta de Video EVG	4		
Tarjeta Madre Gigaby	4		
Seiki TV LED SC-39HS	4		
Samsung Smart TV LED	4		
Tarjeta de Video Gig	3		
ASUS T. Madre uATX M	3		
Logear Audifonos Gam	3		
Tarjeta de Video Asu	2		
SSD para Servidor Le	2		
Acteck Bocina con Su	2		
Genius GHP-400S Audi	2		
Procesador Intel Cor	1		
MSI GeForce Z10 1GB	1		
Tarjeta de Video Vis	1		
Tarjeta Madre Gigaby	1		
Tarjeta Madre ASRock	1		
SSD Samsung 860 EVO	1		
Samsung Smart TV LED	1		
Ghia Bocina Portátil	1		
Ginga Audifonos con	1		

Figura 3.1: Resultados de la opción 1.

La figura 3.2 muestra los resultados de lo menos vendido (tabla de la izquierda) y lo menos buscado (tabla de la derecha), aquí también se nota una fuerte dependencia de lo menos buscado con lo menos comprado.

PRODUCTOS MENOS VENDIDOS				Producto	Categoría
Procesador Intel Core i3-8100	procesadores			Tarjeta de Video EVGA NVIDIA G	tarjetas de video
Tarjeta de Video EVGA NVIDIA G	tarjetas de video			Tarjeta de Video EVGA NVIDIA G	tarjetas de video
Tarjeta de Video EVGA NVIDIA G	tarjetas de video			Tarjeta de Video Gigabyte NVID	tarjetas de video
Tarjeta de Video EVGA NVIDIA G	tarjetas de video			Tarjeta de Video Gigabyte NVID	tarjetas de video
Tarjeta de Video Gigabyte NVID	tarjetas de video			Tarjeta de Video MSI Radeon X1	tarjetas de video
Tarjeta de Video Gigabyte NVID	tarjetas de video			Tarjeta de Video PNY NVIDIA Ge	tarjetas de video
Tarjeta de Video MSI Radeon X1	tarjetas de video			Tarjeta Madre AORUS ATX Z390 E	tarjetas madre
Tarjeta de Video PNY NVIDIA Ge	tarjetas de video			Tarjeta Madre ASRock Z390 Phan	tarjetas madre
Tarjeta de Video VisionTek AMD	tarjetas de video			Tarjeta Madre ASUS ATX PRIME Z	tarjetas madre
Tarjeta de Video VisionTek AMD	tarjetas de video			Tarjeta Madre ASUS ATX ROG STR	tarjetas madre
Tarjeta Madre AORUS ATX Z390 E	tarjetas madre			Tarjeta Madre Gigabyte micro A	tarjetas madre
Tarjeta Madre ASRock Z390 Phan	tarjetas madre			Tarjeta Madre Gigabyte micro A	tarjetas madre
Tarjeta Madre ASUS ATX ROG STR	tarjetas madre			Tarjeta Madre ASRock ATX Z490	tarjetas madre
Tarjeta Madre Gigabyte micro A	tarjetas madre			Tarjeta Madre Gigabyte Micro A	tarjetas madre
Tarjeta Madre Gigabyte micro A	tarjetas madre			Tarjeta Madre ASUS micro ATX P	tarjetas madre
Tarjeta Madre ASRock ATX Z490	tarjetas madre			Tarjeta Madre ASUS ATX ROG STR	tarjetas madre
Tarjeta Madre Gigabyte Micro A	tarjetas madre			SSD Addlink Technology S70 51	discos duros
ASUS T. Madre uATX M4A88T-M	tarjetas madre			SSD para Servidor Supermicro S	discos duros
Tarjeta Madre ASUS micro ATX P	tarjetas madre			SSD para Servidor Lenovo Think	discos duros
Tarjeta Madre ASUS ATX ROG STR	tarjetas madre			SSD para Servidor Lenovo Think	discos duros
SSD Addlink Technology S70 51	discos duros			SSD Samsung 860 EVO 1TB SATA	discos duros
SSD para Servidor Supermicro S	discos duros			Kit Memoria RAM Corsair Vengeance	memorias usb
SSD para Servidor Lenovo Think	discos duros			Makena Smart TV LED 3252 32	pantallas
SSD para Servidor Lenovo Think	discos duros			Seiki TV LED SC-39HS950N 38.5	pantallas
SSD Samsung 860 EVO 1TB SATA	discos duros			Samsung TV LED LH43QMREBGCX	pantallas
Kit Memoria RAM Corsair Vengeance	memorias usb			Samsung Smart TV LED UN70RU71	pantallas
Makena Smart TV LED 3252 32	pantallas			Makena Smart TV LED 4052 40	pantallas
Seiki TV LED SC-39HS950N 38.5	pantallas			Hisense Smart TV LED 40H5500F	pantallas
Samsung TV LED LH43QMREBGCX	pantallas			Samsung Smart TV LED 43 Full	pantallas
Samsung Smart TV LED UN70RU71	pantallas			Samsung Smart TV LED UN32J429	pantallas
Makena Smart TV LED 4052 40	pantallas			Hisense Smart TV LED 50H8F 49.	pantallas
Hisense Smart TV LED 40H5500F	pantallas			Samsung Smart TV LED UN55U7C	pantallas
Samsung Smart TV LED 43 Full	pantallas			Lenovo Barra de Sonido Alámbr	bocinas
Samsung Smart TV LED UN32J429	pantallas			Acetek Bocina con Subwoofer AX	bocinas
Hisense Smart TV LED 50H8F 49.	pantallas			Verbatim Bocina Portátil Mini	bocinas
Samsung Smart TV LED UN55U7C	pantallas			Ghia Bocina Portátil BX300 BI	bocinas
Lenovo Barra de Sonido Alámbr	bocinas			Naceb Bocina Portátil NA-0301	bocinas
Acetek Bocina con Subwoofer AX	bocinas			Ghia Bocina Portátil BX800 BI	bocinas
Verbatim Bocina Portátil Mini	bocinas			Ghia Bocina Portátil BX900 BI	bocinas
Ghia Bocina Portátil BX300 BI	bocinas			Ghia Bocina Portátil BX400 BI	bocinas
Naceb Bocina Portátil NA-0301	bocinas			Ghia Bocina Portátil BX500 BI	bocinas
Ghia Bocina Portátil BX800 BI	bocinas			ASUS Audifonos Gamer ROG Theta	audifonos
Ghia Bocina Portátil BX900 BI	bocinas			Acer Audifonos Gamer Galea 300	audifonos
Ghia Bocina Portátil BX400 BI	bocinas			Audifonos Gamer Balam Rush Orp	audifonos
Ghia Bocina Portátil BX500 BI	bocinas			Energy Sistem Audifonos con Mi	audifonos
ASUS Audifonos Gamer ROG Theta	audifonos			Getttech Audifonos con Micrófo	audifonos
Acer Audifonos Gamer Galea 300	audifonos			Ginga Audifonos con Micrófono	audifonos
Audifonos Gamer Balam Rush Orp	audifonos			logear Audifonos Gamer GHG601	audifonos
Energy Sistem Audifonos con Mi	audifonos			Klip Xtreme Audifonos Blast B	audifonos
Genius GHP-400S Audifonos Alá	audifonos				
Getttech Audifonos con Micrófo	audifonos				
Ginga Audifonos con Micrófono	audifonos				
logear Audifonos Gamer GHG601	audifonos				
Klip Xtreme Audifonos Blast B	audifonos				

Figura 3.2: Otros resultados de la opción 1.

Los resultados de la figura 3.3 muestran los productos mejor (tabla izquierda) y peor (tabla derecha) puntuados, estos valores pueden ser engañosos, debido a que la evaluación se basó solo en el puntaje promedio, y aunque hay productos con puntajes menores a 5, la cantidad de productos comprados es mayor.

3 – Resultados

20 PRODUCTOS MEJOR PUNTEADOS DESCENDENTEMENTE			
Producto	Puntaje	#productos	
Procesador AMD Ryzen	5	2	
Procesador Intel Cor	5	3	
Procesador Intel Cor	5	7	
Procesador Intel Cor	5	4	
Tarjeta de Video ASU	5	3	
Tarjeta de Video MSI	5	2	
Tarjeta de Video MSI	5	1	
Tarjeta de Video Sap	5	2	
Tarjeta de Video Zot	5	1	
Tarjeta Madre Gigaby	5	1	
Kit SSD Kingston KC6	5	3	
SSD Crucial MX500 1	5	1	
SSD Western Digital	5	2	
Kit Memoria RAM Cors	5	1	
TCL Smart TV LED 55S	5	1	
TV Monitor LED 24TL5	5	1	
Logitech Audifonos G	5	1	
Logitech Audifonos G	5	2	
SSD Adata Ultimate S	4.87	15	
Procesador AMD Ryzen	4.81	42	

20 PRODUCTOS PEOR PUNTEADOS ASCENDENTEMENTE			
Producto	Puntaje	#productos	
Tarjeta de Vid	1	1	
Tarjeta Madre	1	1	
Tarjeta Madre	1.83	6	
Tarjeta Madre	2	1	
Cougar Audifc	3	1	
MSI GeForce	4	1	
Tarjeta de Vid	4	1	
HyperX Audifc	4	1	
Tarjeta Madre	4.14	14	
Procesador Al	4.23	13	
Tarjeta de Vid	4.4	5	
Procesador Al	4.46	13	
Tarjeta Madre	4.5	2	
Logitech Boci	4.5	2	
SSD XPG SX82	4.55	11	
Tarjeta Madre	4.56	18	
Tarjeta Madre	4.67	6	
SSD Kingston	4.67	9	
SSD Kingston	4.67	3	
Procesador In	4.7	20	

Figura 3.3: Resultados de la opción 2.

Finalmente, los resultados mostrados en las tablas de la figura 3.4 reflejan inconsistencias, debido a que no hay datos de Octubre y Diciembre, regularmente la gente hace compras en esos meses. Por otro lado, se observa que como transcurrían los meses las ventas fueron disminuyendo, a tal punto de vender un solo producto o ninguno. La segunda tabla refleja lo mismo, lo cual indica bajas, debido a que septiembre es un mes de compras altas. Al observar la venta total anual, muestra que la mayor parte va de los meses de enero a mayo. A partir de ese último mes (mayo) las ventas fueron por con una proporción de 1/2.

INGRESOS MENSUALES		
Mes	Ingreso total mensual	Ingreso promedio mensual
Enero	120237	2268.622642
Febrero	110139	2686.317073
Marzo	164729	3229.980392
Abril	193295	2577.266667
Mayo	96394	2677.611111
Junio	36949	3359
Julio	26949	2449.909091
Agosto	3077	1025.666667
Septiembre	4199	4199
Noviembre	4209	4209
LA VENTA TOTAL ANUAL FUE DE:		
		760177

MESES CON MÁS VENTAS EN EL AÑO (DESCENDENTEMENTE)	
Mes	Número de ventas
Abril	75
Enero	53
Marzo	51
Febrero	41
Mayo	36
Julio	11
Junio	11
Agosto	3
Noviembre	1
Septiembre	1

Figura 3.4: Resultados de la opción 1

3.1 Solución al problema

Se pueden proponer algunas ideas para mejorar el análisis o solucionar el problema:

- El problema de decremento de ventas surge del mes de mayo hacia adelante, disminuyendo por mitad, por lo que se propone revisar específicamente ese mes.
- Al parecer la búsqueda de los productos depende fuertemente de las compras, se sugiere hacer un análisis de correlación para demostrar su dependencia. Partiendo de esto se sugiere hacer un estudio de mercado que analice sus compradores, ya que las mayores compras son de productos muy caros.
- Con respecto a la calificación de los productos, se debe hacer pruebas de significancia estadística, debido a que la comparación no es justa, de aquí se podrá inferir que productos son realmente vendibles y con calidad suficiente para el cliente.

4 — Conclusiones

Un análisis fundamental es presentado en este documentos, ya que la implementación o aplicación de métricas estadísticas reforzaría bastante el análisis. Por lo que, un visión muy general del análisis fue realizado, proponiendo unas soluciones en el capítulo anterior.

El uso de python mostró su relevancia en el ámbito del análisis de datos, debido a su facilidad de programación y su fluidez. Además de que existen muchas herramientas que complementarías un buen análisis.

Un análisis gráfico y con cruces de elementos de las tablas podría dar otra visión o mostrar información no percibida.

Bibliografía

- [1] *Definición de Información*. es-ES. URL: <https://www.definicionabc.com/tecnologia/informacion.php> (visitado 07-09-2020).
- [2] *¿Qué es la ciencia de datos?* | Oracle. es-MX. URL: <https://www.oracle.com/mx/data-science/what-is-data-science.html> (visitado 07-09-2020).

