



REPORTE LIFESTORE

Análisis de datos de la tienda virtual LIFESTORE

Judy Esperanza Montero Zapata

Path: Python for data science

Grupo 2

Septiembre 2020

EMTECH

Emerging Technologies Institute

Tabla de contenido

Introducción:	2
Información preliminar	2
Obtención de datos	2
Depuración	2
Link de Github	2
Estructura del archivo	3
Código	3
Ejecución de código	3
Estructura del código	6
Inicio de sesión	6
Código base	10
Código base para algunos scripts	13
Códigos adicionales	14
Código adicional para algunos scripts	16
Análisis de resultados	16
Ventas	16
Ventas por producto	16
Ventas por categoría	18
Búsquedas	19
Búsqueda por producto	19
Reseñas	20
Reseñas por producto	20
Ingresos y Ventas totales al año	21
Ingresos y ventas mensuales	21
Conclusiones	23

Introducción:

El siguiente documento hace un análisis desde cero de las ventas e ingresos de la tienda LIFESTORE, empezando por el desarrollo del código para el procesamiento de los datos en bruto, hasta el análisis de los resultados obtenidos.

Este proyecto tiene como fin realizar una conclusión acerca del comportamiento de ventas y así en un futuro poder hacer un modelo de negocio o estrategia de venta para la tienda virtual LIFESTORE.

LIFESTORE es una tienda virtual, vende diversos artículos para computadoras y laptops. Inició sus ventas el 1 de enero del 2020, lanzando una aplicación para Android y IOS.

Esta tienda virtual tiene una importante acumulación de inventario y ha tenido reducciones en sus ventas en lo que va del año 2020.

Información preliminar

Obtención de datos

Los datos en bruto fueron obtenidos desde la siguiente liga:

<https://github.com/emtechinstitute/proyecto>

Contiene 3 listas:

- 1: Lista de productos con su ID, nombre, precio, categoría y precio
- 2: Lista de ventas con su ID, producto, calificación, fecha de venta y reembolso
- 3: Lista de búsqueda con su ID y producto.

Depuración

Únicamente dos datos estaban erróneos, la fecha de venta fue cambia, ya que ambos tenían año venta 2019 y 2002. Estos datos se pusieron en año de venta 2020. Por lo que no fue necesaria la depuración de datos, dado que todos estaban ordenados y no hubieron datos perdidos.

El desarrollo del código fue hecho en Python en el entorno de desarrollo de Repl.it

Link de Github

<https://github.com/JudyEMZ/PROYECTO-01-/blob/master/PROYECTO-01-MONTERO-ZAPATA-JUDY-ESPERANZA.zip>

Estructura del archivo

El nombre del archivo se llama PROYECTO-01-MONTERO-JUDY que contiene quince scripts.

El script main.py es donde se ejecutan todos los códigos, es decir todas las consignas. Este muestra un menú donde posteriormente se llama al script que se quiera ejecutar en ese momento.

Scripts del archivo:

Nombre del Script	Función
<i>main.py</i>	Manda a ejecutar todos los demás scripts conforme el usuario lo pida
<i>PMaV.py</i>	Ejecuta el listado de productos con más ventas (consigna 1)
<i>PMaB.py</i>	Ejecuta el listado de productos con más búsquedas (consigna 1)
<i>PMeV.py</i>	Ejecuta el listado de productos con menos ventas
<i>PMeB.py</i>	Ejecuta el listado de productos con menos búsquedas
<i>CMeV.py</i>	Ejecuta las Categorías con menos ventas (consigna 1)
<i>CMeB.py</i>	Ejecuta las Categorías con menos búsquedas (consigna 1)
<i>LMjsR.py</i>	Lista de 20 productos con mejores reseñas (consigna 2)
<i>LPrsR.py</i>	Lista de 20 productos con peores reseñas (consigna 2)
<i>lyVM.py</i>	Total de ingresos y ventas mensuales (consigna 3)
<i>lyVP.py</i>	Ingreso y venta promedio mensual (consigna 3)
<i>TopV.py</i>	Top meses con más ventas (consigna 3)
<i>TopI.py</i>	Top de meses con más ingreso (consigna 3)
<i>Lifestore_file.py</i>	Actúa como almacenamiento de datos, contiene las tres listas

Nota: El nombre de los scripts son abreviaturas del resultado que arrojan

Código

Ejecución de código

A continuación, se mostrará las instrucciones de cómo utilizar el código.

```
Bienvenido a la plataforma de lifestore,  
su tienda virtual favorita  
  
Inicie sesión para comenzar  
  
Nombre de usuario: Judy  
Password: 1111
```

El primer paso es ejecutar el código. Después le pedirá que inicie sesión para entrar al programa. Ingrese nombre de usuario y contraseña, en caso de ser correctos se le permitirá el acceso.

```
Bienvenido a la plataforma de lifestore,  
su tienda virtual favorita
```

```
    Inicie sesión para comenzar
```

```
Nombre de usuario: Sapito
```

```
Password: tururu
```

```
Usuario no registrado
```

```
¿Desea agregar un usuario nuevo?(si/no) : █
```

Si el nombre de usuario o la contraseña son incorrectos se mostrará un mensaje si quiere registrar un usuario nuevo.

```
Nombre de usuario: Sapito
```

```
Password: tururu
```

```
Usuario no registrado
```

```
¿Desea agregar un usuario nuevo?(si/no) : no
```

```
    Saliendo del programa...
```

```
        Adiós
```

En caso de que no quiera, el programa se cerrará por completo.

```
¿Desea agregar un usuario nuevo?(si/no) : si
```

```
Nombre de usuario nuevo : Sapito
```

```
Password: tururu
```

```
    !Has sido resgistrado exitosamente!
```

```
    Por favor vuelva ingresar con su nuevo usuario:
```

```
Nombre de usuario: █
```

Si la respuesta es afirmativa, el programa le pedirá el registro. Posteriormente le pedirá que vuelva a ingresar sus datos.

```
Hola Administrador. . .
```

```
    Eliga la opción que desee ver:
```

```
    Selecciona 1 , 2 o 3 y presiona enter:
```

```
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:
```

```
    Escriba "salir" para cerrar el programa.
```

```
Opción: █
```

¡Bienvenido! El programa te ha reconocido como un administrador.

Este es el menú principal donde podrá elegir 3 opciones. Escriba el número que desee ver y presione enter, caso contrario escriba "salir" y el programa se cerrará.

Se pone como ejemplo escoger la opción 2. Enseguida le mostrará el menú de la opción seleccionada.

```

Opción: 2

  Productos por reseña en el servicio

  Selecciona 1 , 2 o 3 y presiona enter:

  1: Productos con mejores reseñas:
  2: Productos con peores reseñas:

  Escriba "salir" para cerrar el programa.

Opción: █

```

De nuevo elija cualquier opción y presione enter, caso contrario escriba "salir" y el programa se cerrará.

Se pone como ejemplo escoger de nuevo la opción 2.

```

Opción: 2

  LISTA DE PRODUCTOS CON PEORES RESEÑAS

  -----
  ( 1 )
  ID de producto: 17
  Nombre del producto: Tarjeta de Video Gigabyte
  OC, 2GB 256-bit GDDR5, PCI Express 3.0
  Reseña promedio: 1.0
  Devoluciones Totales: 1 de 1 venta(s)

  -----
  ( 2 )
  ID de producto: 45
  Nombre del producto: Tarjeta Madre ASRock ATX B
  151, Intel H110, 32GB DDR4, para Intel
  Reseña promedio: 1.0
  Devoluciones Totales: 1 de 1 venta(s)

```

Se despliega toda la información de esa consigna.

Por cada producto se muestra el nombre de producto y la posición en la que está.

*Cada opción le desplegará la información que ha escogido

```

¿Desea volver la menú?:
1: si:
2: no:
Opción: █

```

Al final de cada lista el programa le preguntará si quiere volver al menú.

El menú al que regresa es el segundo, si quiere volver al menú principal tendrá que "salir" del programa y volver a iniciar sesión.

Si escoge la opción 2: el programa se cerrará.

Estructura del código

Inicio de sesión

Antes de siguientes líneas de código se definieron dos mensajes: Uno de Bienvenida al programa y otro de registro.

La lista *Usuarios* contiene el nombre de los usuarios que pueden entrar a la plataforma.

```
main.py
18
19 Usuarios=[["Javier", "1234"],["Intruso", "3456"],["Judy", "1111"],["1", "1"]]
20
21 #Se valida el input con un status, status =0 (inicio)
22 status=0
23 while status==0: # El bucle while sirve para verificar el status del usuario y ver si se ha registrado alguien
nuevo
24     admin=input("Nombre de usuario: ") #Se le pide al usuario que ingrese sus datos
25     contraseña=input("Password: ")
26
27     for usuario in Usuarios:
28         if admin==usuario[0] and contraseña==usuario[1]:
29             status=1 #Si los datos ingresados son correctos el status es 1, se ha verificado que el usuario existe
30             break
31
32     if status==0:
33         print("Usuario no registrado") #Si el status no cambia, pide un registro
34         respuesta=input("¿Desea agregar un usuario nuevo?(si/no) : ")
35         if respuesta=="si": #Pasos cuando el usuario incia un registro
36             admin=input("Nombre de usuario nuevo : ")
37             contraseña=input("Password: ")
38             Usuarios.append([admin,contraseña]) # Se añade a la lista de usuarios
39             print(msg2)
40     elif respuesta=="no": #Pasos cuando el usuario no quiere registrarse
41         msg2=""
42         Saliendo del programa...
43         |
44         | | | | | Adiós""
45         print(msg2)
46         status=2
47     else:
48         print("Error al ingresar datos") #Si hay un ingreso al input distinto de si o no, arroja un error.
no
```

```

50 #Si el usuario existe y se le muestra el menu de opciones
51
52 if status==1:
53     #menu principal
54     msg="""
55     Hola Administrador. . .
56
57     | | | Eliga la opción que desee ver:
58
59     Seleccione 1 , 2 o 3 y presiona enter:
60
61     1: Productos más vendidos y productos rezagados:
62     2: Productos por reseña en el servicio:
63     3: Total de ingresos y ventas promedio mensuales,
64     | total anual y meses con más ventas al año:
65
66     Escriba "salir" para cerrar el programa.
67     """
68
69     #se verifican las opciones del usuario
70     opcion=0 #Se crea la variable opcion para verificar el input del usuario
71
72     while opcion!="salir": #El menú se mostrará hasta que el usuario escriba salir
73         print(msg)
74         opcion=input("Opción: ") #input para escoger el segundo menú de opciones
75         #Se verifican las opciones
76         if opcion=="1":
77             #bucle 2, verifica opciones del segundo menú
78             opcion2=0
79             while opcion2!="salir":
80                 msg="""
81                 Productos más vendidos y productos rezagados
82
83                 Seleccione 1, 2, 3 o 4 y presiona enter:
84
85                 1: Productos con mayores ventas:
86                 2: Productos con mayores búsquedas:
87                 3: Productos con menores ventas:
88                 4: Productos con menores búsquedas:
89                 5: Categoría con menos ventas:
90                 6: Categoría con menos búsquedas:
91
92                 Escriba "salir" para cerrar el programa.
93                 """
94                 print(msg)
95                 opcion2=input("Opción: ")
96                 #Aquí se mostrarán los resultados dependiendo del menú que el usuario elija
97                 if opcion2=="1":
98                     import PMAV #Listado productos más vendidos
99

```



```

295         opcion="salir"
296         break
297
298     elif opcion2=="3":
299         from TopV import lista11 #Top meses con mayores ventas
300         print("\nTOP MESES CON MÁS VENTAS\n")
301         for j in lista11:
302             print("\n Mes: ",j[0],"\nVentas totales: ", j[1])
303         print("\n ¿Desea volver la menú?: \n1: si:\n2: no:")
304         opcion3=input("Opción: ")
305
306         if opcion3!="1":
307             opcion="salir"
308             break
309
310     elif opcion2=="4":
311         from TopI import lista12 #Top meses con mayores ingresos
312         print("\nTOP MESES CON MÁS INGRESO\n")
313         for j in lista12:
314             print("\n Mes: ",j[0],"\nIngresos totales: $", j[1])
315
316         print("\n ¿Desea volver la menú?: \n1: si:\n2: no:")
317         opcion3=input("Opción: ")
318         if opcion3!="1":
319             opcion="salir"
320             break
321
322     elif opcion2!="salir":
323         print("Algo no ha salido bien")
324
325
326     elif opcion!="salir":
327         print("OPCION NO ENCONTRADA, vuelve a intentarlo") #si no se verifica ninguna opción repite el menú 1
328
329     msg2=""
330     Saliendo del programa...
331
332     | | | | | Adiós""
333
334     print(msg2) #Mensaje cuando se sale completamente del bucle

```

Resumen:

Lo que hace este código es ejecutar un menú hasta que el usuario elija una opción, ya sea 1,2 o 3 etc. Para hacer esto se usa un bucle `while`. Si el usuario escoge una opción distinta como: kdjfk ó fjej se mostrará un mensaje de error y se volverá a mostrar el menú. La única forma de salir completamente del bucle es escribir "salir" en el input de opciones.

Dentro del primer bucle `while` existe otro bucle `while` para ejecutar tres menús, estos se ejecutan cuando el usuario elige una opción. De igual manera la única forma de salir completamente del bucle es escribir "salir" en el input de opciones.

Cuando el usuario elige una opción, se ejecutan los resultados que se muestran en cada menú. Los resultados son llamados mediante `from` e `import`.

Código base

Mi código base 1 está compuesto por tres fases.

Primero llama al script `Lifestore_file.py` donde se almacena toda la información (todas las listas).

FASE 1:

Primer código para fase 1

```
8 from lifestore_file import lifestore_products
9 from lifestore_file import lifestore_sales
10 from lifestore_file import lifestore_searches
11
12 # FASE 1: Este Paso cuenta cuántas veces se vendió un determinado producto. Se almacena en la lista
   "lista"
13
14 contador=0
15 lista=[] #Lista de ventas por producto
16
17 for sale in range(len(lifestore_sales)):
18     contador=contador+1
19
20     if lifestore_sales[sale][1]!=lifestore_sales[sale-1][1]:
21         lista.append([lifestore_sales[sale-1][1],contador])
22         contador=0
23
24     if lifestore_sales[sale]==lifestore_sales[-1]:
25         lista.append([lifestore_sales[sale][1],contador+1])
26         lista.remove(lista[0])
27
```

Imagen código (1)

Este bucle recorre la longitud de la lista `lifestore_sales` es decir empieza desde 0 hasta 282.

Se crea una `lista` nueva para almacenar los datos que se recuperan del bucle. La lista contiene `[ID de producto, número de ventas de ese producto]`. Se crea una variable `contador` con valor inicial 0, para almacenar las repeticiones de un producto (segunda entrada de la lista). Nota: Cada que se repite un producto es una venta de ese producto.

El bucle empieza recorriendo cada elemento de `lifestore_sales` y almacena el conteo de cada producto, y verifica si la segunda entrada de la lista `lifestore_sales` (ID producto) es igual al elemento anterior, si es igual, el contador sigue su curso, pero si no es igual, el contador se detiene en el último elemento antes de el elemento distinto o elemento actual. Este último elemento se almacena en la lista `lista` junto con su contador. Este proceso termina cuando `sale` ha recorrido toda la lista `lifestore_sales`. La segunda sentencia es para que el último elemento se tome en cuenta.

Segundo código para fase 1

```
12
13 lista=[]
14
15 for ID in lifestore_products:
16     cont=0
17
18     for sale in lifestore_sales:
19         if sale[1]==ID[0]:
20             cont=cont+1
21
22
23 lista.append([ID[0],cont])
24
```

Imagen código (2)

Esta parte hace lo mismo que la *imagen código(1)*. Se toma `contador=0` para almacenar las repeticiones del ID. Esta toma como referencia el ID de producto de la lista `lifestore_products` y este es comparado con el ID de la lista `lifestore_sales`, cuando ambos elementos son iguales, se suma 1 a `contador` y así para cada elemento existente en `lifestore_products`. Cada que termina una iteración, se almacena el elemento iterado en lista.

Nota: he tomado `contador= contador +1` porque es más cómodo para mí que `contador+=1`.

FASE 2:

Primer código para fase 2

```
28 #Este paso ordena la lista de ventas de mayor a menor.Se almacena en la lista "lista_ordenada"
29
30 lista_ordenada=[]
31
32 while lista:
33
34     m=lista[0][1]
35     maximo=lista[0]
36
37     for elemento in range(len(lista)):
38
39         if lista[elemento][1] > m:
40             m=lista[elemento][1]
41             maximo=lista[elemento]
42         if lista[elemento][1] == m and lista[elemento]<maximo:
43             m=lista[elemento][1]
44             maximo=lista[elemento]
45
46     lista_ordenada.append(maximo)
47     lista.remove(maximo)
48
49 #Este paso te muestra el resultado final
50
```

Imagen código (3)

Se crea `lista_ordenada` para almacenar los elementos ya ordenados de mayor a menor.

Bucle `while`: Mientras exista algún elemento en `lista` el bucle se seguirá ejecutando. Se toman dos variables iniciales que contienen el primer elemento de la `lista`, primera entrada (`maximo`) y segunda entrada (`m`). La segunda entrada `m` es la cual tomaremos como objetivo para ordenar la lista. Estas dos variables se toman como máximo y son comparadas con el resto de los elementos en `lista`. Cuando un elemento es mayor a la variable `m` entonces ese elemento toma el lugar de `m` (se convierte en el nuevo máximo).

Si existe un elemento igual a `m`, se toma como referencia la primera entrada del elemento que se está comparando.

Cuando se encuentra el `máximo` dentro de una iteración, este se almacena en `lista_ordenada` y se elimina de lista. Al final de todas las iteraciones `lista_ordenada` tendrá los elementos que contenía `lista` pero ordenados de mayor a menor. `Lista` que dará vacía y el bucle se detendrá.

Segundo código para fase 2:

```
27
28 lista_ordenada=[]
29
30 while lista:
31     m=lista[0][1]
32     minimo=lista[0]
33
34     for elemento in range(len(lista)):
35
36         if lista[elemento][1] < m:
37             m=lista[elemento][1]
38             minimo=lista[elemento]
39         if lista[elemento][1] == m and lista[elemento]<minimo:
40             m=lista[elemento][1]
41             minimo=lista[elemento]
42
43     lista_ordenada.append(minimo)
44     lista.remove(minimo)
45
46 #print(lista_ordenada)
47 #Este paso te muestra el resultado final
48
```

Imagen código (4)

Similar al anterior pero se toma como referencia el valor mínimo, así se puede ordenar una lista de menor a mayor.

FASE 3:

```
51 x=int(input("\n ¿Cuántos productos desea ver?: ")) #se le pide al usuario ingresar un número
52
53 if x>43 or x<=0:
54     x=42
55     print("NOTA: Se le mostrarán únicamente productos con ventas") #Dado que existen sólo 42 productos con
    ventas se pone esta restricción
56
57 for i in range(x):
58     contador=0
59     while lista_ordenada[i][0]!=lifestore_products[contador][0]:
60         contador=contador+1
61
62     msg=""
63     -----
64     ""
65     print(msg,"( ",i+1," )")
66     print("ID de producto: ", lista_ordenada[i][0])
67     print("Nombre del producto: ",lifestore_products[contador][1])
68     print("Ventas totales: ", lista_ordenada[i][1])
69
70
71
```

Imagen código (5)

El ciclo **for** sirve para que los productos sean mostrados en consola. El ciclo **while** dentro del **for** sirve para buscar el nombre del producto en la lista **lifestore_products**. Compara cada ID en ambas listas (**lifestore_products** y **lifestore_sales**) cuando es igual, el producto es mostrado en consola con ID, nombre de producto y ventas totales.

Código base para algunos scripts

Los siguientes scripts tienen implementadas las fases 1,2 y 3.

Los Scripts con asterisco indican que, en ese código la lista **lifestore_sales** fue sustituida por **lifestore_searches**. El código es el mismo sólo se hace ese cambio de referencia.

Para los Scripts que inician con “C”, el 1er código fase 1 toma como comparación las categorías en vez del producto. El código es el mismo sólo se hace ese cambio de comparación.

Scripts	Contenido
PMaV.py	[1er código fase 1][1er código fase 2][fase 3]
PMaB.py*	[1er código fase 1][1er código fase 2][fase 3]
PMeV.py	[2do código fase 1][2do código fase 2][fase 3]
PMeB.py*	[2do código fase 1][2do código fase 2][fase 3]
CMeB.py*	[1er código fase 1][2do código fase 2][fase 3]
CMeV.py	[1er código fase 1][2do código fase 2][fase 3]

Códigos adicionales

Código adicional 1:

Este código es similar a la 2do código para la fase 2.

```
12 lista=[]
13
14 for ID in lifestore_products:
15     suma=0
16     cont=0
17     devoluciones=0
18
19     for sale in lifestore_sales:
20         if sale[1]==ID[0]:
21             cont=cont+1
22             suma=sale[2]+suma
23             devoluciones+=sale[4]
24
25     if suma!=0:
26         prom=suma/cont
27     else:
28         prom=0
29
30     lista.append([ID[0],prom,devoluciones,cont])
31
32 #print(lista)
33
```

El `if` dentro del `for` nos sirve para contar las devoluciones que se han hecho en un determinado producto.

Aquí se toma como objetivo conocer el promedio de las reseñas. La variable `suma` almacena el score de reseñas de determinado producto, la variable `contador` almacena el número de veces que se calificó un producto. Así al sacar el promedio es dividir `suma/contador`, siempre y cuando suma es distinto de cero. Suma puede ser igual a cero cuando un producto no ha sido vendido.

Código adicional 2:

Se crea una lista vacía `Lista_separada`, esta almacenará el ID de producto, el precio del producto, mes en el que fue vendido, año en que fue vendido, ID de venta, devolución.

El mes y el año fueron separados usando índices.

```
16 Lista_separada=[]
17
18
19 for i in range(len(lifestore_sales)):
20     contador=0
21     while lifestore_sales[i][1]!=lifestore_products[contador][0]:
22         contador=contador+1
23
24     Lista_separada.append([lifestore_sales[i][1],lifestore_products[contador][2],int(lifestore_sales[i][3]
25     [3:5]),int(lifestore_sales[i][3][6:10]),lifestore_sales[i][0],lifestore_sales[i][4]])
26
27 #print(Lista_separada)
28
```

El código revisa que los ID de cada lista `lifestore_products` y `lifestore_sales` sean iguales, cuando son iguales agrega a la lista `lista_separada` el precio de esos productos. También se agrega la lista el mes y el año separados y convertidos en enteros.

Código adicional 3:

Se toman en cuenta las ventas que no tuvieron devoluciones `if i[5]==0`. Se suman los precios del producto de cada venta realizada, así se obtiene el ingreso total, que es almacenado en la variable `suma`. Similar con las ventas totales que son almacenadas en `cont`.

```
--
30 print( " INGRESOS Y VENTAS TOTALES POR MES EN EL 2020 ")
31
32 for j in range(1,12+1):
33     suma=0
34     cont=0
35     for i in Lista_separada:
36         if i[2]==j:
37             if i[5]==0:
38                 suma+=i[1]
39                 cont+=1
40
41     lista.append([j,suma,cont])
42
43     print("\n Mes: ",j,"\n Ingreso total: $",suma,"\n Ventas totales: ",cont)
```

Luego se le agrega a una nueva lista llamada `lista`. Tome en cuenta que se toma cada venta e ingreso por cada mes.

Código adicional 4:

En esta parte se cuenta las ventas que no tuvieron devoluciones. Se saca el ingreso total anual y ventas totales anuales. No se toma en cuenta el mes que se ha hecho la venta.

```
--
26 lista=[]
27
28 print( " INGRESO Y VENTA PROMEDIO MENSUAL EN EL 2020 \n")
29
30 suma=0
31 cont=0
32 for elemento in Lista_separada:
33     if elemento[5]==0:
34         suma+=elemento[1]
35         cont+=1
36
37 print("Ingreso promedio mensual: $",suma/12)
38 print("Venta promedio mensual: ",int(cont/12))
39
40 print( " INGRESOS Y VENTAS TOTALES EN EL 2020 \n")
41
42 print("Ingreso total anual: $",suma)
43 print("Venta total anual: ",cont)
```


Al final el resultado **suma** que almacena los ingresos y **cont** que almacena las ventas, son divididos entre doce para sacar el promedio mensual.

Código adicional para algunos scripts

Los siguientes scripts tienen implementados códigos adicionales a parte de las fases 1,2 y 3

Scripts	Contenido
LMjsR.py	[código adicional 1][1er código fase 2][fase3]
LPrsR.py	[código adicional 1][2do código fase 2][fase 3]
lyVM.py	[código adicional 2][código adicional 3]
lyVP.py	[código adicional 2][código adicional 4]
TopI.py	[código adicional 2][código adicional 3] [1er código fase 2]
TopIV.py	[código adicional 2][código adicional 3] [1er código fase 2]

Nota: Algunos códigos cambiarán por el objetivo de cada código, por ejemplo, tomar ventas en vez de ingresos.

Algo importante por mencionar es que se ha utilizado un **ciclo for** para presentar el resultado de cada uno de los scripts.

Análisis de resultados

Ventas

Ventas por producto

Los siguientes resultados fueron obtenidos con el programa de Python. Se muestran una serie de gráficos para hacer más visual el resultado.

Nota: El total de ventas de esta sección no considera devoluciones

En el gráfico 1 se puede observar el top 15 de productos más vendidos. El primer lugar es el producto con ID 54 (SSD Kingston A400, 120GB, SATA III, 2.5", 7mm,) de la categoría discos duros.



gráfico (1)

Se tiene que el total de ventas es 282 productos sin tomar en cuenta las devoluciones, lo que quiere decir que tan solo con las ventas de los primeros 10 productos se alcanza aproximadamente el 73% del total de ventas. Los productos que tienen mayores ventas se asocian con las categorías procesadores, discos duros y tarjetas madre.

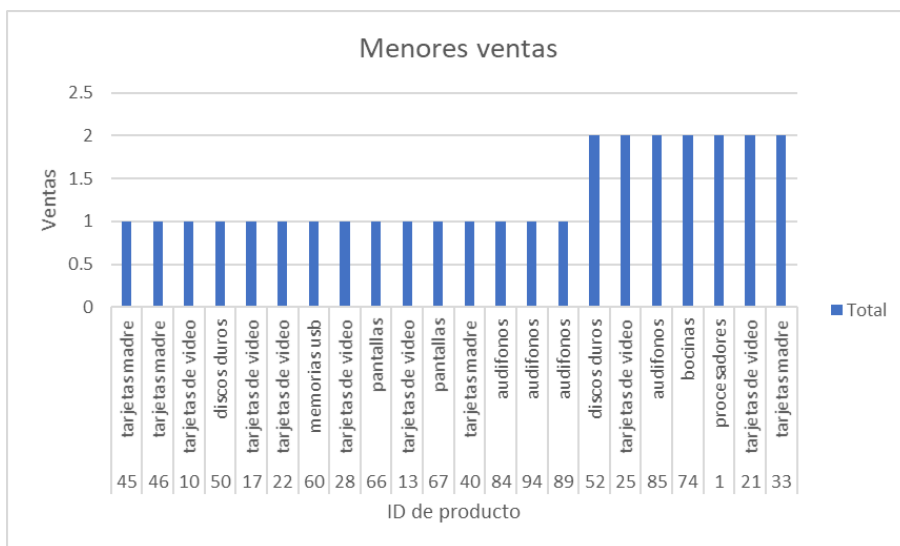


gráfico (2)

En el gráfico (2) muestra las menores ventas por producto. Con los 15 productos menos vendidos sólo se alcanza el 5% del total de ventas.

Tabla de producto sin ventas:

ID	Ventas	ID	Ventas	ID	Ventas
9	0	39	0	73	0
14	0	41	0	75	0
15	0	43	0	76	0
16	0	53	0	77	0
19	0	55	0	78	0
20	0	56	0	79	0
23	0	58	0	80	0
24	0	59	0	81	0
26	0	61	0	82	0
27	0	62	0	83	0
30	0	63	0	86	0
32	0	64	0	87	0
34	0	65	0	88	0
35	0	68	0	90	0
36	0	69	0	91	0
37	0	70	0	92	0
38	0	71	0	93	0
		72	0	95	0

Se tiene en la base de datos un total de 96 productos registrados. Sólo 42 productos tienen una venta o más, dejando 54 productos sin venta ¡ 56% de los productos que ofrece la tienda *lifestore* no se venden!

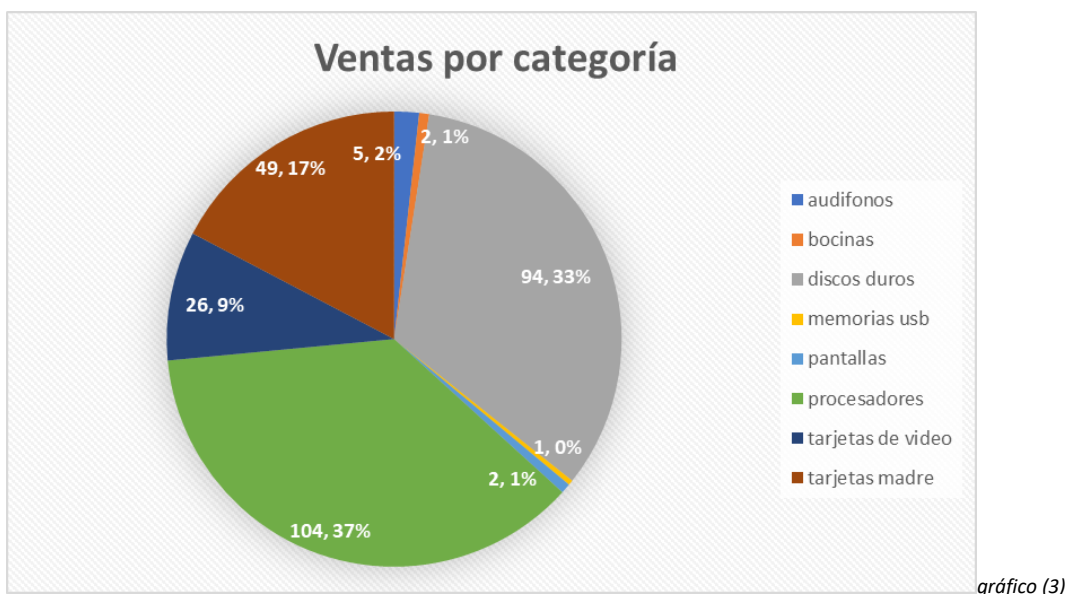
Este es un dato interesante porque juntos tiene un inventario de 2131 productos.

Durante un año estos productos no se han vendido ni una vez.

Pueden haber varias razones para este fenómeno, tales como un precio alto del producto o simplemente la gente no le interesa estos productos.

Ventas por categoría

Las categorías de procesadores y discos duros son las categorías que más acaparan las ventas, tal como lo vimos en mayores ventas por producto.



Dentro de nuestro del programa de Python se obtienen las ventas por categoría, estos datos se plasman en la siguiente tabla.

Categoría	Productos x categoría	Ventas	Stock
memorias usb	2	1	15
bocinas	10	2	146
pantallas	12	2	1190
audifonos	13	5	573
tarjetas de video	19	26	349
tarjetas madre	18	49	814
discos duros	13	94	433
procesadores	9	104	1821

En total existen 8 categorías.

Las categorías con menos ventas son memorias usb, bocinas y pantallas.

En caso de la categoría de memorias no es tan crítica como el de la categoría de pantallas y bocinas.

La categoría memorias tiene en inventario 15 productos y sólo vendió uno en todo lo que va del año, pero la categoría pantallas ha vendido dos productos y tiene 1190 en stock, menos de 1% de lo que tienen ahora.

En general ninguna categoría ha vendido más del 50% de sus productos en inventario por lo que sería recomendable disminuir el stock con ofertas o algún otro tipo de estrategia. La tecnología se vuelve obsoleta en cuestión nada, y si esos productos no se sacan a la venta serán pérdidas para la empresa.

Búsquedas

Búsqueda por producto

El producto con ID 54 tiene 263 de búsquedas totales. Este ID coincide con el ID de mayores ventas.

En el siguiente gráfico se muestran el top 15 productos con mayores búsquedas. Esto nos ayuda a averiguar que producto le interesa más a la gente que entra a la tienda.



gráfico (4)

Pero buscar un producto solo nos aporta el interés de las personas, pero ¿qué tanto lo compra la gente?

En el gráfico (5) podemos observar el contraste de búsqueda y ventas.

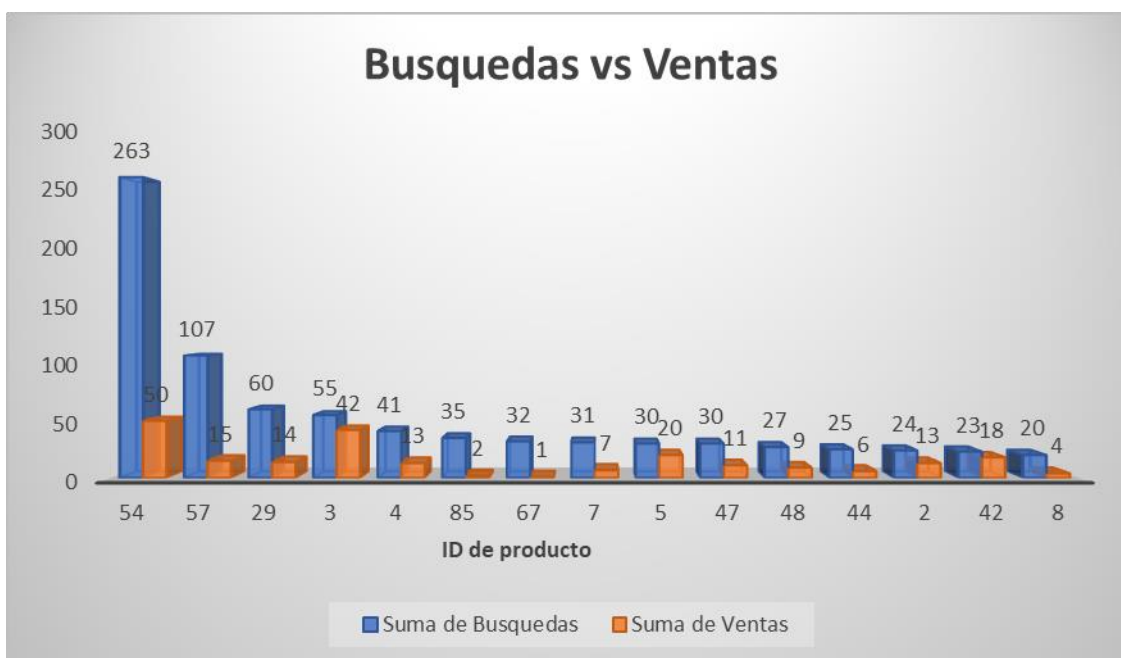


gráfico (5)

Aunque el ID 54 sea el más buscado y el más comprado. Su índice de venta no es muy alto a comparación del índice de búsqueda, similar a los ID 57, ID 59. Una de las razones que podría no venderse un producto es el precio de este. También como lifestore es una tienda virtual los precios de envío podrían no ser adecuados.

Reseñas

Reseñas por producto

Sabemos que cuando un individuo al momento de comprar en una tienda virtual se fija mucho en las reseñas. Este comportamiento podemos verlo en las siguientes gráficas:

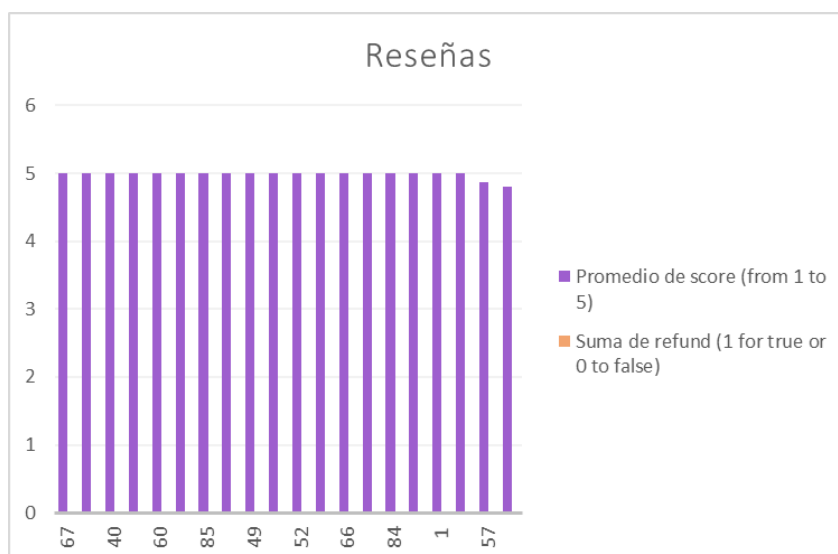
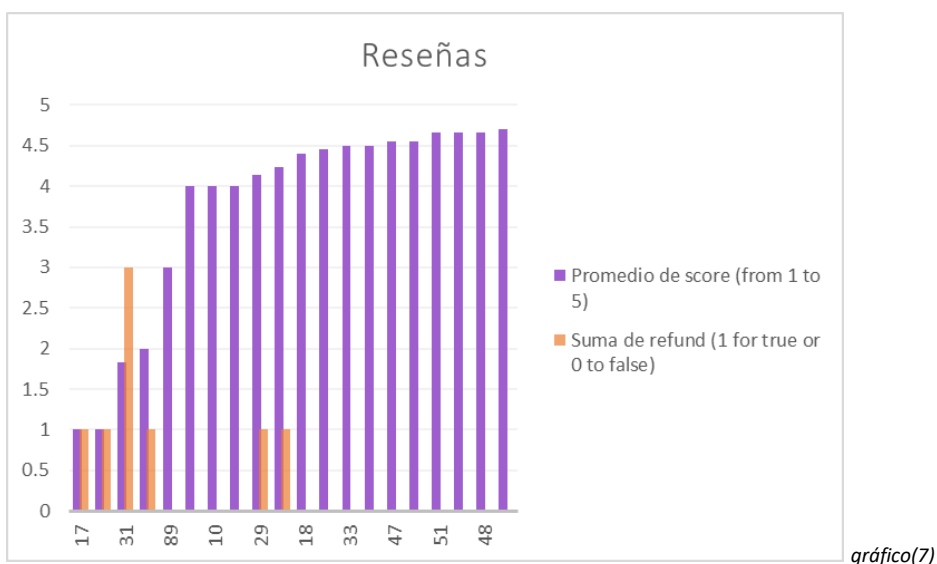


gráfico (6)

Los productos con mejores reseñas no tienen ningún reembolso.



Los productos con peores reseñas tienen devoluciones. EL ID 31 tiene 3 devoluciones y un puntaje menor a 2.

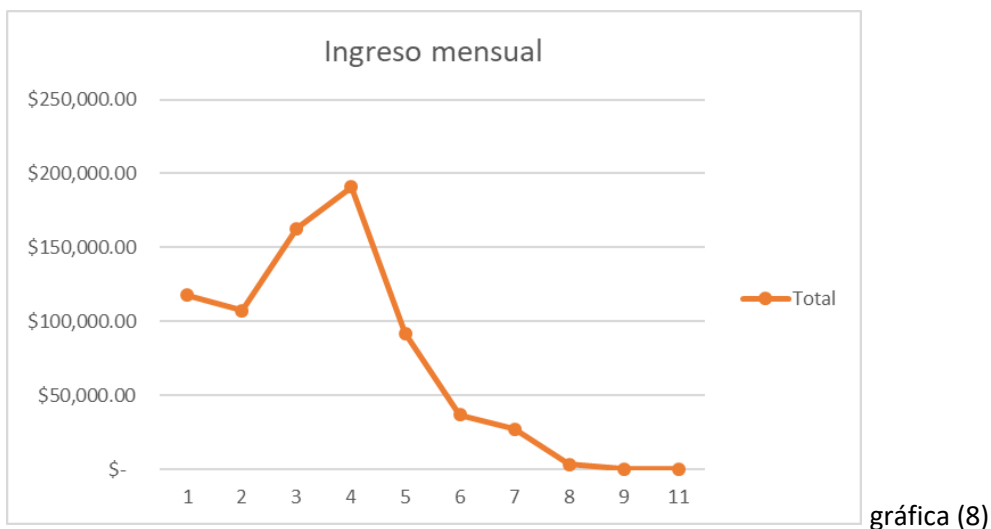
Con esto podemos concluir que la calidad de la mayoría de los productos que se venden es buena. Entonces esta no es una razón por el cual no se venden los productos.

Ingresos y Ventas totales al año

Ingresos y ventas mensuales

Se sacaron los ingresos netos mensuales para observar como se comportaron durante el año.

En la gráfica (8) se observa que de febrero a abril hubo un ingreso ascendente, de igual manera las ventas subieron, gráfica (9). Sin embargo, a partir del mes de abril los ingresos y las ventas fueron bajando hasta casi ser 0. Este no es un escenario favorable para cualquier empresa.



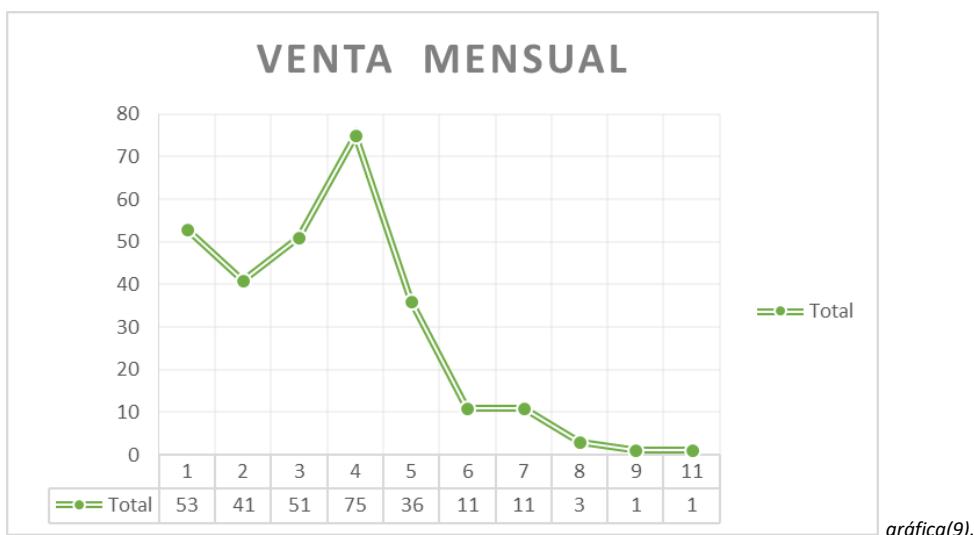
Mes	Ingreso	Ingreso promedio
Enero	\$ 117,738.00	\$ 92,239.50
Febrero	\$ 107,270.00	\$ 92,239.50
Marzo	\$ 162,931.00	\$ 92,239.50
Abril	\$ 191,066.00	\$ 92,239.50
Mayo	\$ 91,936.00	\$ 92,239.50
Junio	\$ 36,949.00	\$ 92,239.50
Julio	\$ 26,949.00	\$ 92,239.50
Agosto	\$ 3,077.00	\$ 92,239.50
Septiembre		
Octubre		
Noviembre		
Diciembre		

Es claro que la gente no siguió comprando como en los primeros meses.

A partir de septiembre se esperaría que las ventas mejoraran.

Pero, si la tendencia de compras y ventas siguen de esa manera, no hay forma que la empresa pueda recuperar la inversión inicial a la tienda virtual.

*El ingreso promedio mensual se sacó a partir de ingreso total anual entre el número de meses entre el número de meses transcurridos es total/8.



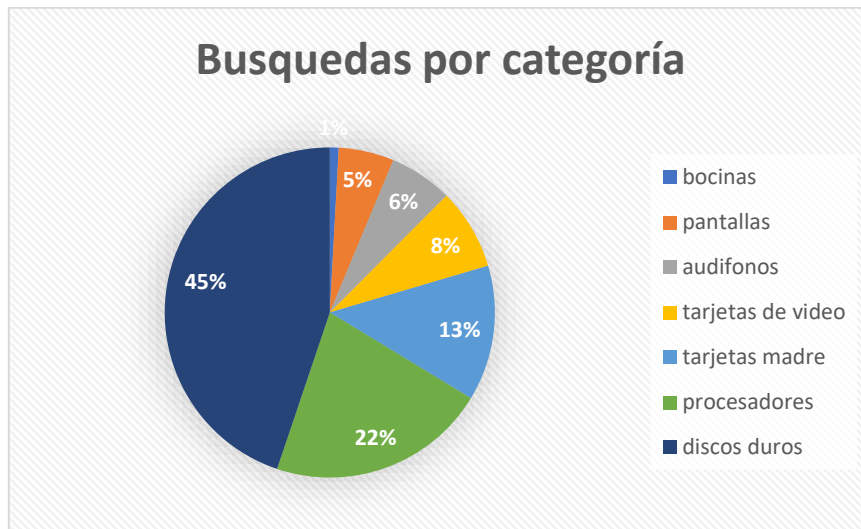
Para saber exactamente que sucede detrás de la baja de ventas se tendría que averiguar si las búsquedas también bajaron (asignarle fechas a las búsquedas).

Si un determinado producto conforme pasa el tiempo es menos buscado, significa que ya no es de interés, por tanto, la gente no lo comprará. Este producto habría que sacarlo del stock.

Si un producto sigue siendo buscado conforme pasa el tiempo y no es buscado, el problema podría venir de otra fuente como lo sería el precio.

Conclusiones

La sugerencia de productos a retirar de la tienda sería todos los de la categoría de pantallas y bocinas porque son los productos que menos se venden y menos se buscan.



No valdría la pena conservar alguno producto de estas categorías porque ni el 5% del stock se vende.

También sería recomendable disminuir el inventario de todos los productos. Por ejemplo, el producto con ID 3 con categoría procesador es el segundo producto con más ventas, pero se tienen en existencia 987. De enero a agosto sólo se han vendido 42 ejemplares y es posible que no se vendan más de 50 en lo que resta del año. Se podría pensar que al año siguiente se pueden vender más ejemplares, sin embargo, la tecnología cambia constantemente dejando a modelos anteriores completamente obsoletos. No conviene tenerlos tanto tiempo en inventario.

Para reducir el inventario se podrían implementar ofertas, como:

- En la compra de \$1000 en tienda el envío es gratis
- En la compra de dos audífonos el tercero va a mitad de precio.

Ofertas que persuadan al cliente a comprar más para obtener un “beneficio” extra. Sin perder las ganancias.

Pero que pasa si con las ofertas implementadas no mejoran las ventas y los ingresos se reducen.

Pueden existir muchos factores que hagan que los ingresos tengan un comportamiento a la baja.

Como antes mencionamos la calificación de la mayoría de los productos era buena, entonces el factor calidad no afecta considerablemente a las ventas. Los precios se han mantenido a lo largo del año, pero se podría averiguar si otras tiendas de la misma índole están dando mejores precios que lifestore.

También una parte importante para que una empresa tenga éxito en tiendas online es la calidad del servicio que ofrece. Como son:

- Tiempo de entrega
- Formas de pago y reembolso
- Costo de envío

El servicio se puede medir igual que la calidad del producto. El usuario puede asignarle una puntuación y así verificar si se tiene éxito con el servicio al cliente.

En caso de que el servicio sea malo, se puede deducir que las ventas no mejorarán, pero cuando el cliente queda satisfecho con el servicio, este lo recomienda con sus conocidos y la probabilidad de vender más aumenta.