

Proyecto Final 1

Fundamentos de Programación con Python

Saúl Yael Puente Ruiz
16 de septiembre de 2021

Índice

Índice	1
Introducción	2
Definición del código	2
Solución al problema	13
Conclusión	17

Introducción

El presente reporte contiene el análisis realizado al caso propuesto en la parte 1 del curso *Fundamentos de Programación con Python*. El caso para analizar trata sobre una tienda virtual y los datos que ha podido recolectar mientras se ha encontrado en funcionamiento. La Gerencia de ventas de esta tienda ha solicitado que se haga un análisis sobre los productos en venta debido a una gran acumulación de inventario y a una disminución de búsquedas en algunos productos.

Para conseguir una solución se utilizarán diferentes métricas como la cantidad de ventas por producto y el número de veces que éste ha aparecido en alguna búsqueda. Además, se considerará que tan bien calificado es cada producto por sus compradores, así como si realiza alguna devolución o no. De manera más general se considerarán las ventas, reembolsos e ingresos en por mes y año, aunque en este caso sólo se considera el 2020.

Definición del código

A continuación, se explica el código en Python para realizar el análisis del caso. Este código se puede encontrar en el siguiente repositorio en GitHub:

<https://github.com/SaulPuente/IntroduccionAPython-Proyecto1.git>

Antes de comenzar con el análisis de los datos el programa solicita sus datos de acceso al usuario, y si es necesario registrarse. Se guardaron varios usuarios previamente, con diferentes niveles de acceso, pero no hay diferencia entre ellos.

```
print("\nSea bienvenido a mi proyecto 1 del curso Fundamentos de Programación con Python")

print("\nPara comenzar debe iniciar sesión")

print("\nSi desea salir del programa escriba \'Salir\' , si no es así escriba \'continuar\' \n")

continuar = input()

login = False

while continuar == "continuar" or continuar == "Si" or continuar == "si":

    new_user = input("\n¿Es un nuevo usuario? Si es un nuevo usuario puede registrarse escribiendo \'Si\' ")

    if new_user == "Si" or new_user == "si":

        print("\nIngrese su nuevo usuario y contraseña")

    else:

        print("\nMuy bien, acceda con su usuario y contraseña")
```

```

login = False
user = input("Usuario: ")
password = input("Contraseña: ")
if new_user == "Si" or new_user == "si":
    print("\nConfirme su contraseña: ")
    if password == input():
        users[user] = {"Contraseña": password, "Nivel": "Lector"}
        continuar = "No"
        login = True
    else:
        print("\nError, vuelva a introducir sus datos.")
elif users[user]["Contraseña"] == password:
    print("\nDatos ingresados correctamente, su nivel de permisos es \" +
users[user]["Nivel"] + "\"")
    nivel = users[user]["Nivel"]
    continuar = "No"
    login = True
else:
    print("\nIngrese sus datos nuevamente")

```

El primer paso es importar los datos necesarios para el análisis del caso, los cuales se encuentran en el archivo "lifestore_file.py", posteriormente se guardan los datos en variables. Además, se requiere importar las librerías necesarias.

```

import lifestore_file as LSF
import pandas as pd
products = LSF.lifestore_products
sales = LSF.lifestore_sales
searches = LSF.lifestore_searches

```

En el código las partes de éste se encuentran agrupadas en funciones. En el siguiente bloque de código se hace limpieza de los datos introducidos, seleccionando sólo las ventas correspondientes al año indicado. Para conseguir esto debemos descomponer la fecha dada en cada una de las ventas registradas. Separamos la fecha usando el carácter "/" en

día, mes y año. Mediante un ciclo analizamos cada elemento de la lista, cuando aparezca una venta diferente al año seleccionado, éste se elimina de la lista.

```
def limpieza(año):
    global sales
    año = str(año)
    ventas = sales.copy()
    for venta in ventas:
        venta[3] = venta[3].split("/")
    i = 0
    n = len(sales)
    while i < n:
        if ventas[i][3][2] != año:
            ventas.pop(i)
            sales.pop(i)
            i -= 1
            n = len(sales)
        i += 1
    return ventas
```

Una vez realizada la limpieza podemos comenzar a analizar los datos. Primero obtendremos los productos más y menos vendidos. Para comenzar contamos las veces que ha sido vendido cada producto y lo guardamos en una lista. Al finalizar con el conteo ordenamos la lista según la cantidad de ventas realizadas en cada producto.

```
def ventasPorProducto(products, sales):
    # Creamos una lista con las ventas por producto
    ventas_por_producto = [[i+1,0,0] for i in range(len(products))]
    # Contamos las ventas que obtuvo cada producto
    i = 1
    for sale in sales:
        while sale[1] != i:
            i += 1
        ventas_por_producto[i-1][1] += 1
        if sale[3] == 1:
            ventas_por_producto[i-1][2] += 1
    # Ordenamos la lista usando la función previamente definida
    ventas_por_producto.sort(key = lambda x: x[1], reverse = True)
    return ventas_por_producto
```

Con la lista preparada lo siguiente es mostrar los resultados en pantalla. En el siguiente bloque de código generamos un dataframe con los productos más vendidos y los menos vendidos, posteriormente los imprimimos en pantalla y los guardamos en archivos csv.

```
def imprimirProductosPorVentas(ventas_por_producto):
    pd.options.display.max_colwidth = 70 #-----
    print("\n Productos más vendidos\n")
    # Obtenemos el nombre de cada uno de los primeros productos y los guardamos en
    una lista
    productosMasVendidos = [[ventas_por_producto[i][0],
    products[ventas_por_producto[i][0]-1][1], ventas_por_producto[i][1]] for i in range(10)]
    # Le damos formato de tabla a la lista
    productosMasVendidos = pd.DataFrame(productosMasVendidos)
    productosMasVendidos.columns = ["ID","Productos","Ventas"]
    productosMasVendidos = productosMasVendidos.set_index("ID")
    # Imprimimos
    print(productosMasVendidos)
    #Guardamos el dataframe en un archivo csv
    productosMasVendidos.to_csv("productosMasVendidos.csv")

    print("\n Productos menos vendidos\n")
    # Obtenemos el nombre de cada uno de los últimos productos y los guardamos en una
    lista
    productosMenosVendidos = [[ventas_por_producto[i][0],
    products[ventas_por_producto[i][0]-1][1], ventas_por_producto[i][1]] for i in range(-1,-11,-1)]
    # Le damos formato de tabla a la lista
    productosMenosVendidos = pd.DataFrame(productosMenosVendidos)
    productosMenosVendidos.columns = ["ID","Productos","Ventas"]
    productosMenosVendidos = productosMenosVendidos.set_index("ID")
    # Imprimimos
    print(productosMenosVendidos)
    #Guardamos el dataframe en un archivo csv
    productosMenosVendidos.to_csv("productosMenosVendidos.csv")

Un proceso similar se utiliza para analizar los productos más buscados. Primero
ordenamos los productos según las búsquedas.

def busquedasPorProducto(products,searches):
    # Creamos una lista con las búsquedas por producto
    busquedas_por_producto = [[i+1,0] for i in range(len(products))]
```

```

# Contamos las búsquedas que obtuvo cada producto
i = 1
for search in searches:
    while search[1] != i:
        i += 1
    busquedas_por_producto[i-1][1] += 1
# Ordenamos la lista usando la función previamente definida
busquedas_por_producto.sort(key = lambda x: x[1], reverse = True)
return busquedas_por_producto
Posteriormente exportamos los resultados.

def imprimirProductosPorBusquedas(busquedas_por_producto):
    pd.options.display.max_colwidth = 70 #-----

    print("\n Productos más buscados\n")
    # Obtenemos el nombre de cada uno de los últimos productos y los guardamos en una
    lista
    productosMasBuscados = [[busquedas_por_producto[i][0],
    products[busquedas_por_producto[i][0]-1][1], busquedas_por_producto[i][1]] for i in
    range(10)]
    # Le damos formato de tabla a la lista
    productosMasBuscados= pd.DataFrame(productosMasBuscados)
    productosMasBuscados.columns = ["ID","Producto","Busquedas"]
    productosMasBuscados = productosMasBuscados.set_index("ID")
    # Imprimimos
    print(productosMasBuscados)
    # Guardamos el dataframe en un archivo csv
    productosMasBuscados.to_csv("productosMasBuscados.csv")

    print("\n Productos menos buscados\n")
    # Obtenemos el nombre de cada uno de los últimos productos y los guardamos en una
    lista
    productosMenosBuscados = [[busquedas_por_producto[i][0],
    products[busquedas_por_producto[i][0]-1][1], busquedas_por_producto[i][1]] for i in range(-
    1,-11,-1)]
    # Le damos formato de tabla a la lista
    productosMenosBuscados = pd.DataFrame(productosMenosBuscados)

```

```

productosMenosBuscados.columns = ["ID","Producto","Busquedas"]
productosMenosBuscados = productosMenosBuscados.set_index("ID")
# Imprimimos
print(productosMenosBuscados)
# Guardamos el dataframe en un archivo csv
productosMenosBuscados.to_csv("productosMenosBuscados.csv")

```

Ahora analizaremos la calificación que recibió cada uno de los productos en sus ventas. Para esto sumaremos cada una de las reseñas que un producto haya recibido, y después obtendremos el promedio de éste.

```

def reseñasPorProducto(products, sales):
    # Creamos una lista con las ventas por producto
    reseñas_por_producto = [[i+1,0] for i in range(len(products))]

    # Contamos las ventas que obtuvo cada producto y obtenemos el promedio de
    calificación que recibió cada uno
    i = 1
    n = 0
    for sale in sales:
        while sale[1] != i:
            if n != 0:
                reseñas_por_producto[i-1][1] = round(reseñas_por_producto[i-1][1]/n,2)
            n = 0
            i += 1
        reseñas_por_producto[i-1][1] += sale[2]
        n += 1

    # Ordenamos la lista usando la función previamente definida
    reseñas_por_producto.sort(key = lambda x: x[1], reverse = True)
    return reseñas_por_producto

```

Luego, imprimimos estos datos y los exportamos a un archivo csv, como hicimos anteriormente.

```

def imprimirProductosPorReseñas(reseñas_por_producto):
    pd.options.display.max_colwidth = 70 #-----

    print("\n Productos mejor calificados\n")
    # Obtenemos el nombre de cada uno de los primeros productos y los guardamos en
    una lista

```



```

    productosMejorCalificados = [[reseñas_por_producto[i][0],
products[reseñas_por_producto[i][0]-1][1], reseñas_por_producto[i][1]] for i in range(20)]
    # Le damos formato de tabla a la lista
    productosMejorCalificados = pd.DataFrame(productosMejorCalificados)
    productosMejorCalificados.columns = ["ID","Productos","Calificación"]
    productosMejorCalificados = productosMejorCalificados.set_index("ID")
    # Imprimimos
    print(productosMejorCalificados)
    # Guardamos el dataframe en un archivo csv
    productosMejorCalificados.to_csv("productosMejorCalificados.csv")

    print("\n Productos peor calificados\n")
    # Obtenemos el nombre de cada uno de los últimos productos y los guardamos en una
    lista
    productosPeorCalificados = [[reseñas_por_producto[i][0],
products[reseñas_por_producto[i][0]-1][1], reseñas_por_producto[i][1]] for i in range(-1,-
21,-1)]
    # Le damos formato de tabla a la lista
    productosPeorCalificados = pd.DataFrame(productosPeorCalificados)
    productosPeorCalificados.columns = ["ID","Productos","Calificación"]
    productosPeorCalificados = productosPeorCalificados.set_index("ID")
    # Imprimimos
    print(productosPeorCalificados)
    # Guardamos el dataframe en un archivo csv
    productosPeorCalificados.to_csv("productosPeorCalificados.csv")

```

El último parte del código se basa en los ingresos recibidos. Lo primero que haremos será ordenar y agrupar las ventas según el año en el que se realizaron, para esto utilizaremos el siguiente bloque de código.

```

def obtenerVentasAnuales(ventas):
    # Ordenamos las ventas por año
    ventas.sort(key = lambda x: x[3][2])

    ventas_anuales = []
    venta_anual = []
    año = ventas[0][3][2]

    # Iteramos las ventas y agrupamos por año

```

```

for venta in ventas:
    if venta[3][2] != año:
        ventas_anuales.append(venta_anual)
        venta_anual = []
        año = venta[3][2]
    venta_anual.append(venta)
ventas_anuales.append(venta_anual)

del(venta_anual)
# Ordenamos las ventas en cada año por mes
for venta_anual in ventas_anuales:
    venta_anual.sort(key = lambda x: x[3][1])
return ventas_anuales

```

Realizamos el mismo procedimiento para obtener las ventas mensuales en cada año.

```

def obtenerVentasMensuales(ventas_anuales):
    ventas_mensuales = []
    venta_mensual = []
    año = ventas_anuales[0][0][3][2]
    mes = ventas_anuales[0][0][3][1]

    # Iteramos en cada año
    for venta_anual in ventas_anuales:
        # Iteramos en cada elemento de la lista anual para ordenar y agrupar por mes
        for venta in venta_anual:
            if venta[3][2] != año or venta[3][1] != mes:
                ventas_mensuales.append(venta_mensual)
                venta_mensual = []
                año = venta[3][2]
                mes = venta[3][1]
            venta_mensual.append(venta)
        ventas_mensuales.append(venta_mensual)

    return ventas_mensuales

```

Con las listas de ventas organizadas ahora procedemos a calcular las ventas, ingresos, reembolsos y las pérdidas provenientes de estos reembolsos en cada año a analizar, en este caso sólo el año 2020.

```

def obtenerIngresosAnuales(ventasAnuales):
    # Calculamos los ingresos obtenidos en cada año
    # Iteramos por cada año
    for i in range(len(ventas_anuales)):
        venta_anual = ventas_anuales[i]
        ventas = len(venta_anual)
        año = venta_anual[0][3][2]
        ingresos = 0
        reembolsos = 0
        perdidas = 0
        # Iteramos por cada venta en el año y sumamos ventas y reembolsos
        for venta in venta_anual:
            ingresos += products[venta[1]-1][2]
            if venta[-1] == 1:
                reembolsos += 1
                perdidas += products[venta[1]-1][2]
            ventas_anuales[i] = [año,ventas,ingresos,reembolsos,perdidas] + venta_anual

        return ventas_anuales

```

Realizamos un proceso parecido con los meses del año, sin embargo, esta vez incluiremos nuevas métricas como son los promedios de ingresos y reembolsos.

```

def obtenerIngresosMensuales(ventas_mensuales):
    # Variables para guardar promedios
    promedio_ingreso_mensual = 0
    promedio_venta_mensual = 0
    promedio_perdida_mensual = 0
    promedio_reembolso_mensual = 0

    # iteramos por cada mes
    for i in range(len(ventas_mensuales)):
        venta_mensual = ventas_mensuales[i]
        ventas = len(venta_mensual)
        año = venta_mensual[0][3][2]
        mes = venta_mensual[0][3][1]
        ingresos = 0

```

```

reembolsos = 0
perdidas = 0
# Iteramos por cada venta en el mes
for venta in venta_mensual:
    # Calculamos los ingresos y reembolsos
    ingresos += products[venta[1]-1][2]
    if venta[-1] == 1:
        reembolsos += 1
        perdidas += products[venta[1]-1][2]
# Calculamos los promedios por cada mes
promedio_ingreso_mensual += ingresos
promedio_venta_mensual += ventas
promedio_perdida_mensual += perdidas
promedio_reembolso_mensual += reembolsos
ventas_mensuales[i] = [mes,año,ventas,ingresos,reembolsos,perdidas] +
venta_mensual

# Terminamos de calcular los promedios e imprimos los resultados
print("")
print("Total de ingresos: $" + str(promedio_ingreso_mensual))
promedio_ingreso_mensual /= len(ventas_mensuales)
promedio_venta_mensual /= len(ventas_mensuales)
promedio_perdida_mensual /= len(ventas_mensuales)
promedio_reembolso_mensual /= len(ventas_mensuales)
print("Ingreso promedio mensual: $" + str(round(promedio_ingreso_mensual,2)))
print("Ventas promedio mensual: " + str(round(promedio_venta_mensual,2)))
print("Perdida promedio mensual: $" + str(round(promedio_perdida_mensual,2)))
print("Reembolso promedio mensual: " + str(round(promedio_reembolso_mensual,2)))

```

```

return ventas_mensuales

```

Imprimimos los resultados de los ingresos anuales con el formato adecuado.

```

def imprimirIngresosAnuales(ventas_anuales):

```

```

    for venta_anual in ventas_anuales:

```

```

        print("\nIngresos en " + str(venta_anual[0]) + ": $" + str(venta_anual[2]))

```

Por último, definimos una función para imprimir los datos que necesitamos de los ingresos mensuales y los guardamos en un archivo csv.

```

def imprimirIngresosMensuales(ventas_mensuales):
    ventas_mensuales.sort(key = lambda x: x[2]-x[4], reverse = True)
    def format(x):
        return "${:.2f}".format(x)

    print("\n Meses con más ventas al año\n")
    mesesMasVentas =
[[ventas_mensuales[i][0],ventas_mensuales[i][2],ventas_mensuales[i][3],
ventas_mensuales[i][4], ventas_mensuales[i][5]] for i in range(len(ventas_mensuales))]
    mesesMasVentas = pd.DataFrame(mesesMasVentas)
    mesesMasVentas.columns = ["Mes","Ventas","Ingresos","Reembolsos","Perdidas"]
    mesesMasVentas = mesesMasVentas.set_index("Mes")
    mesesMasVentas['Ingresos'] = mesesMasVentas['Ingresos'].apply(format)
    mesesMasVentas['Perdidas'] = mesesMasVentas['Perdidas'].apply(format)
    print(mesesMasVentas)
    mesesMasVentas.to_csv("mesesMasVentas.csv")

```

En la última parte del código ejecutamos todas estas funciones en caso de que el usuario haya iniciado sesión correctamente.

if login:

```

    print("\n")
    print("*****")
    print("\n")
    input()
    print("\nA continuación se muestran los resultados obtenidos de los datos
proporcionados en el año 2020.")
    ventas = limpieza(2020)
    ventas_por_producto = ventasPorProducto(products,sales)
    busquedas_por_producto = busquedasPorProducto(products,sales)
    imprimirProductosPorVentas(ventas_por_producto)
    input()
    imprimirProductosPorBusquedas(busquedas_por_producto)
    input()
    reseñas_por_producto = reseñasPorProducto(products,sales)
    imprimirProductosPorReseñas(reseñas_por_producto)
    input()
    ventas_anuales = obtenerVentasAnuales(ventas)

```

```

ventas_mensuales = obtenerVentasMensuales(ventas_anuales)
ingresos_anuales = obtenerIngresosAnuales(ventas_anuales)
imprimirIngresosAnuales(ingresos_anuales)
input()
ingresos_mensuales = obtenerIngresosMensuales(ventas_mensuales)
imprimirIngresosMensuales(ingresos_mensuales)
input()
print("\nEso fue todo, muchas gracias por usar este programa. :)")

```

Solución al problema

Primeramente, vamos a analizar los resultados del programa uno por uno. Tenemos que los 10 productos más vendidos son los siguientes.

ID	Productos	Ventas
54	SSD Kingston A400, 120GB, SATA III, 2.5", 7mm	49
3	Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth	42
5	Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)	20
42	Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD	18
57	SSD Adata Ultimate SU800, 256GB, SATA III, 2.5", 7mm	15
29	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD	14
4	Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire	13
2	Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth	12
47	SSD XPG SX8200 Pro, 256GB, PCI Express, M.2	11
12	Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC, 6GB 192-bit GDDR6, PCI Express x16 3.0	9

A continuación, tenemos los productos menos, los cuales tienen cero ventas cada uno. Aquí sólo 10 productos menos vendidos, pero en realidad aún hay más con cero ventas.

ID	Productos	Ventas
96	Klip Xtreme Audífonos Blast, Bluetooth, Inalámbrico, Negro/Verde	0
95	logear Audífonos Gamer GHG601, Alámbrico, 1.2 Metros, 3.5mm, Negro	0
93	Ginga Audífonos con Micrófono GI18ADJ01BT-RO, Bluetooth, Alámbrico/Inalámbrico, 3.5mm, Rojo	0
92	Getttech Audífonos con Micrófono Sonority, Alámbrico, 1.2 Metros, 3.5mm, Negro/Rosa	0

91	Genius GHP-400S Audífonos, Alámbrico, 1.5 Metros, Rosa	0
90	Energy Sistem Audífonos con Micrófono Headphones 1, Bluetooth, Inalámbrico, Negro/Grafito	0
88	Audífonos Gamer Balam Rush Orphix RGB 7.1, Alámbrico, USB, Negro	0
87	Acer Audífonos Gamer Galea 300, Alámbrico, 3.5mm, Negro	0
86	ASUS Audífonos Gamer ROG Theta 7.1, Alámbrico, USB C, Negro	0
83	Ghia Bocina Portátil BX500, Bluetooth, Inalámbrico, 10W RMS, USB, Gris	0

Los siguientes son los productos más buscados por los cibernautas que ingresan a la página. Podemos notar que varios de ellos coinciden con los productos más vendidos, lo cual nos da una razón más para conservar estos productos.

ID	Producto	Búsquedas
54	SSD Kingston A400, 120GB, SATA III, 2.5", 7mm	263
57	SSD Adata Ultimate SU800, 256GB, SATA III, 2.5", 7mm	107
29	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD	60
3	Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth	55
4	Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire	41
85	Logitech Audífonos Gamer G635 7.1, Alámbrico, 1.5 Metros, 3.5mm, Negro/Azul	35
67	TV Monitor LED 24TL520S-PU 24, HD, Widescreen, HDMI, Negro	32
7	Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)	31
5	Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)	30
47	SSD XPG SX8200 Pro, 256GB, PCI Express, M.2	30

En la tabla se muestran los productos menos buscados. Como es de esperarse, algunos de éstos también se encuentran entre los productos menos vendidos, teniendo cero ventas como cero búsquedas. Esto es un fuerte indicador de que debe cambiarse la estrategia sobre estos productos, ya sea eliminarlos del inventario o promocionarlos más.

ID	Producto	Búsquedas
96	Klip Xtreme Audífonos Blast, Bluetooth, Inalámbrico, Negro/Verde	0
92	Gettech Audífonos con Micrófono Sonority, Alámbrico, 1.2 Metros, 3.5mm, Negro/Rosa	0
90	Energy Sistem Audífonos con Micrófono Headphones 1, Bluetooth, Inalámbrico, Negro/Grafito	0
88	Audífonos Gamer Balam Rush Orphix RGB 7.1, Alámbrico, USB, Negro	0

87	Acer Audífonos Gamer Galea 300, Alámbrico, 3.5mm, Negro	0
86	ASUS Audífonos Gamer ROG Theta 7.1, Alámbrico, USB C, Negro	0
83	Ghia Bocina Portátil BX500, Bluetooth, Inalámbrico, 10W RMS, USB, Gris	0
82	Ghia Bocina Portátil BX400, Bluetooth, Inalámbrico, 8W RMS, USB, Negro	0
81	Ghia Bocina Portátil BX900, Bluetooth, Inalámbrico, 2.1 Canales, 34W, USB, Negro - Resistente al Agua	0
79	Naceb Bocina Portátil NA-0301, Bluetooth, Inalámbrico, USB 2.0, Rojo	0

Entre los productos mejor calificados casi todos obtuvieron una puntuación perfecta, con excepción de algunos productos con una calificación casi igual de buena.

ID	Productos	Calificación
1	Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Quad-Core, 16MB L2 Cache	5
6	Procesador Intel Core i9-9900K, S-1151, 3.60GHz, 8-Core, 16MB Smart Cache (9na. Generación Coffee Lake)	5
7	Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)	5
8	Procesador Intel Core i5-9600K, S-1151, 3.70GHz, Six-Core, 9MB Smart Cache (9na. Generación - Coffee Lake)	5
11	Tarjeta de Video ASUS AMD Radeon RX 570, 4GB 256-bit GDDR5, PCI Express 3.0	5
21	Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC, 8GB 128-bit GDDR6, PCI Express 4.0	5
22	Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC, 4GB 128-bit GDDR5, PCI Express x16 3.0	5
25	Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500 XT Gaming, 8GB 128-bit GDDR6, PCI Express 4.0	5
28	Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti, 6GB 192-bit GDDR6, PCI Express x16 3.0	5
40	Tarjeta Madre Gigabyte XL-ATX TRX40 Designare, S-sTRX4, AMD TRX40, 256GB DDR4 para AMD	5
49	Kit SSD Kingston KC600, 1TB, SATA III, 2.5, 7mm	5
50	SSD Crucial MX500, 1TB, SATA III, M.2	5
52	SSD Western Digital WD Blue 3D NAND, 2TB, M.2	5
60	Kit Memoria RAM Corsair Dominator Platinum DDR4, 3200MHz, 16GB (2x 8GB), Non-ECC, CL16, XMP	5
66	TCL Smart TV LED 55S425 54.6, 4K Ultra HD, Widescreen, Negro	5
67	TV Monitor LED 24TL520S-PU 24, HD, Widescreen, HDMI, Negro	5
84	Logitech Audífonos Gamer G332, Alámbrico, 2 Metros, 3.5mm, Negro/Rojo	5
85	Logitech Audífonos Gamer G635 7.1, Alámbrico, 1.5 Metros, 3.5mm, Negro/Azul	5

57	SSD Adata Ultimate SU800, 256GB, SATA III, 2.5", 7mm	4.87
3	Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth	4.81

Las reseñas tampoco favorecen a los productos con los últimos ID, pues sin ventas no pueden recibir una calificación para comprobar su calidad.

ID	Productos	Calificación
96	Klip Xtreme Audífonos Blast, Bluetooth, Inalámbrico, Negro/Verde	0
95	Iogear Audífonos Gamer GHG601, Alámbrico, 1.2 Metros, 3.5mm, Negro	0
93	Ginga Audífonos con Micrófono G18ADJ01BT-RO, Bluetooth, Alámbrico/Inalámbrico, 3.5mm, Rojo	0
92	Getttech Audífonos con Micrófono Sonority, Alámbrico, 1.2 Metros, 3.5mm, Negro/Rosa	0
91	Genius GHP-400S Audífonos, Alámbrico, 1.5 Metros, Rosa	0
90	Energy Sistem Audífonos con Micrófono Headphones 1, Bluetooth, Inalámbrico, Negro/Grafito	0
88	Audífonos Gamer Balam Rush Orphix RGB 7.1, Alámbrico, USB, Negro	0
87	Acer Audífonos Gamer Galea 300, Alámbrico, 3.5mm, Negro	0
86	ASUS Audífonos Gamer ROG Theta 7.1, Alámbrico, USB C, Negro	0
83	Ghia Bocina Portátil BX500, Bluetooth, Inalámbrico, 10W RMS, USB, Gris	0
82	Ghia Bocina Portátil BX400, Bluetooth, Inalámbrico, 8W RMS, USB, Negro	0
81	Ghia Bocina Portátil BX900, Bluetooth, Inalámbrico, 2.1 Canales, 34W, USB, Negro - Resistente al Agua	0
80	Ghia Bocina Portátil BX800, Bluetooth, Inalámbrico, 2.1 Canales, 31W, USB, Negro	0
79	Naceb Bocina Portátil NA-0301, Bluetooth, Inalámbrico, USB 2.0, Rojo	0
78	Ghia Bocina Portátil BX300, Bluetooth, Inalámbrico, 40W RMS, USB, Rojo - Resistente al Agua	0
77	Verbatim Bocina Portátil Mini, Bluetooth, Inalámbrico, 3W RMS, USB, Blanco	0
76	Acteck Bocina con Subwoofer AXF-290, Bluetooth, Inalámbrico, 2.1, 18W RMS, 180W PMPO, USB, Negro	0
75	Lenovo Barra de Sonido, Alámbrico, 2.5W, USB, Negro	0
73	Samsung Smart TV LED UN55TU7000FXZX 55, 4K Ultra HD, Widescreen, Negro/Gris	0
72	Hisense Smart TV LED 50H8F 49.5, 4K Ultra HD, Widescreen, Negro	0

Los siguientes son los meses con los datos de sus diferentes transacciones, ordenados por el número de ventas. Podemos notar algo curioso que es que no se tienen registradas

ventas durante el invierno, de hecho, las ventas también disminuyen para los otros meses conforme se acerca el frío. En la tabla los meses más aptos para vender resultan ser aquellos que se encuentran en primavera.

Mes	Ventas	Ingresos	Reembolsos	Perdidas
4	75	\$193,295.00	1	\$2,229.00
1	53	\$120,237.00	1	\$2,499.00
3	51	\$164,729.00	2	\$1,798.00
2	41	\$110,139.00	1	\$2,869.00
5	35	\$96,135.00	2	\$4,458.00
6	11	\$36,949.00	0	\$0.00
7	11	\$26,949.00	0	\$0.00
8	3	\$3,077.00	0	\$0.00
9	1	\$4,199.00	1	\$4,199.00

También contamos con los siguientes datos:

Ingresos en 2020: \$755709

Total de ingresos: \$755709

Ingreso promedio mensual: \$83967.67

Ventas promedio mensual: 31.22

Perdida promedio mensual: \$2005.78

Reembolso promedio mensual: 0.89

En este caso el ingreso total es el mismo que el anual al sólo analizarse el 2020. Estas métricas hablan muy bien de la empresa ya que se presentan grandes ingresos con pocas pérdidas, esto visto por la gran cantidad de ventas y pocos reembolsos.

Conclusión

Desde mi punto de vista la mejor decisión LifeStore tiene varias, que podrían ser el deshacerse de aquellos productos que no producen ventas y que no se ganan el interés del usuario. En caso de que la tienda decida que pueden venderse estos productos sería necesario promocionar mejor estos productos rezagados, sin dejar de lado a aquellos que ya demuestran ser exitosos. Sería prudente dar más énfasis a esto en aquellos meses en los que se realizan más ventas, que son esos que ocurren durante la primavera y meses más cálidos.

Por otro lado, considero que esta práctica fue muy instructiva me dio la oportunidad de aplicar mis conocimientos sobre y descubrir algunos nuevos. También me ayudó a utilizar mi capacidad para analizar y sacar conclusiones, habilidades necesarias para cualquiera que aspire a convertirse en un científico de datos.