Emtech Institute

Primer proyecto

Tema: Fundamentos de programación de Python

Creado por: Guzmán Pérez Victor Daniel

Correo: vdgp_12@hotmail.com

Contenido

2 introducción

Utilizar el conocimiento de Python sobre análisis y clasificación de datos aprendido en el curso de introducción para crear una solución para la empresa LifeStore la cual ha tenido problemas con la acumulación de su inventario al igual que con una reducción en las búsquedas de un grupo importante de productos

3 solución al problema

Se creó un programa que inicialmente tiene un log in para que solo pueda ingresar el personal autorizado. Existe un menú con 4 opciones.

1.- Productos más vendidos y productos rezagados.

Esta opción primero te muestra el top 5 de los productos más vendidos y el top 10 de los productos más buscados. En productos rezagados por cada categoría de producto existente te muestra el top 5 productos con menores ventas y el top 10 productos con menores búsquedas.

2.- Productos por reseña en el servicio.

Esta opción te muestra el top cinco productos con peores reseñas y el top 5 productos con mejores reseñas.

3.- Total de ingresos y ventas.

Esta opción te muestra todo lo relacionado con ventas, en primer lugar, te muestra por cada mes el total de ingresos, total de ventas y promedio de ingresos. Por año muestra el total de ventas y el top de los meses con más ventas.

4.- La opción numero 4 funciona para terminar el programa.

3.1 Log in

Usuario : user

Pass: user

Lo primero que se muestra iniciando el programa es un log in el cual pide que ingreses el nombre de usuario y contraseña. El log in manda un mensaje de error si la contraseña o el usuario son incorrectos, después de tres intentos fallidos se muestra un mensaje de despedida y se cierra el programa.

En caso de que las credenciales sean correctas se muestra un mensaje de bienvenida, el ciclo while se termina y se continua con el menú de opciones.

```python
def login():
    # usuario: user
    # contrasenia: user
    intentos = 0
    correcto = False
    while (not correcto):
        if intentos == 3:
            print("Intentos superados, ¡adios!")
            quit()
        usuario = input("\n Ingrese usuario: ")
        password = input("Ingrese Password: ")
        if usuario == "user" and password == "user":
            print("_____-")
            print("\t \t \t \t \t \t Bienvenido")
            print("_____-")
            correcto = True
        else:
            print("\n")
            if usuario != "user":
                print("¡Usuario incorrecto!")
            if password != "user":
                print ("Contraseña Incorrecta")
            intentos +=1
```

3.2 Menú de opciones

Esta función contiene un menú para cada una de las opciones mencionadas en la introducción, se intenta simular un for utilizando sentencias, dependiendo que opción se elija se manda a llamar su función correspondiente.

```python
def menu():
    opcion= 0
    while(opcion !=4):
        print("\n\n MENU ")
        print("[1] Productos más vendidos y productos rezagados")
        print("[2] Productos por reseña en el servicio")
        print("[3] Total de ingresos y ventas promedio mensuales, total anual y meses con más ventas al año")
        print("[4] Terminar")
        opcion= int(input("\nSelecciona una opcion: "))
        if opcion == 1:
            parte1()
        elif opcion == 2:
            parte2()
        elif opcion == 3:
            parte3()
        elif opcion == 4:
            quit()
        else:
            print("\n opcion incorrecta")
```

3.3 Productos más vendidos y productos rezagados.

Estas opciones se encuentran dentro de la función parte1()

Aquí se muestran las variables y listas utilizadas durante el proceso

Tanto la lista cantidad_ventas y cantidad_busquedas se crearon para tener almacenada la información necesaria de cada producto

```
    cantidad_ventas= []             # (nombre_producto, numero de ventas, categoria)
    productosMasVendidos=[]         # contiene los cinco productos con mas ventas
    cantidad_busquedas = []         #( nombre_producto, numero de busquedas, categoria)
    productosMasBuscados=[]         #contiene los diez productos mas buscados
    categorias=[]                   #contiene las distintas categorias de los productos
    ventasCategoria=[]              #contiene los productos de cada categoria
    busquedasCategoria=[]           #Contiene los productos de cada categoria
    count = 0
    print("_____________________________________________")
    print("\n\n \t \t Opción 1 Productos más vendidos y productos rezagados")
    print("_____________________________________________")
```

Los for anidados que se muestran en la imagen sirven para agregar valores a la lista cantidad_ventas, por cada producto que se encuentre en la lista lifestore_products se compara su id con todas las ventas almacenadas en lifestore_sales si los id´s coinciden significa que esa venta fue del producto que se está comparando así que se le suma uno a la variable count la cual lleva la cuenta de la cantidad de ventas por producto, al finalizar el segundo for se agrega a la lista cantidad_ventas el nombre del producto y la cantidad de ventas (variable count).

Cuando se terminan los dos for se hace un acomodo a la lista cantidad_ventas de menor a mayor basándose en la cantidad de ventas.

Para finalizar se guardan los últimos cinco valores de la lista cantidad_ventas, los cuales son los cinco con más ventas, en la lista productos más vendidos.

```
#Creando la lista de cantidad ventas (nombre_producto, numero de ventas, categoria)
#Se recorre primero la lista de productos para comparar cada producto con todas las ventas que hay
for producto in lifestore_products:
    count=0
    for ventas in lifestore_sales:
        #Se cuenta la cantidad de veces que el id de un producto aparece en la lista de ventas
        if producto[0] == ventas[1] :
            count +=1
    #Se agrega el valor a la lista nueva
    cantidad_ventas.append([producto[1],count,producto[3]])
# Se ordenan los elementos de menor a mayor basandonos en el numero de ventas posicion 1
cantidad_ventas = sorted(cantidad_ventas, key= lambda x: x[1] )
# se guarda en una lista nueva los ultimos cinco elementos de la lista cantidad_ventas, los cuales son los productos mas vendidos
productosMasVendidos= cantidad_ventas[len(cantidad_ventas)-5 : len(cantidad_ventas)]
```

Los for anidados que se muestran en la imagen sirven para agregar valores a la lista cantidad_busquedas, por cada producto que se encuentre en la lista lifestore_searches se compara su id con todas las busquedas almacenadas en lifestore_searches si los id´s coinciden significa que esa busqueda fue del producto que se está comparando así que se le suma uno a la

variable count la cual lleva la cuenta de la cantidad de ventas por producto, al finalizar el segundo for se agrega a la lista cantidad_busquedas el nombre del producto y la cantidad de búsquedas de ese producto (variable count).

Cuando se terminan los dos for se hace un acomodo a la lista cantidad_busquedas de menor a mayor basándose en la cantidad de busquedas.

Para finalizar se guardan los últimos cinco valores de la lista cantidad_busquedas, los cuales son los cinco más buscados, en la lista productosmasBuscados.

```
# Creando la lista de cantidad_busquedas (nombre_producto, numero de busquedas, categoria)
#Busquedas de productos
for producto in lifestore_products:
    count = 0
    for busquedas in lifestore_searches:
        if producto[0] == busquedas[1]:
            count +=1
    cantidad_busquedas.append([producto[1],count, producto[3]])
# Se ordenan los elementos de menor a mayor basandonos en el numero de busquedas posicion 1
cantidad_busquedas = sorted(cantidad_busquedas, key=lambda x: x[1])
# se guarda en una lista nueva los ultimos diez elementos de la lista cantidad_busquedas, los cuales son los productos mas buscados
productosMasBuscados = cantidad_busquedas[len(cantidad_busquedas) - 10: len(cantidad_busquedas)]
productosMasVendidos.reverse()
productosMasBuscados.reverse()
```

Después se imprimen ambas listas con los resultados obtenidos.

```
print("\n ----------Cinco Productos con mayores ventas----------\n")
for x in range(0,5):
    print(f"[{x+1}] Nombre: {productosMasVendidos[x][0][:45]}")
    print(f"Numero de ventas: {productosMasVendidos[x][1]}\n")


print("\n ----------Diez Productos con mayores Busquedas----------\n")
for x in range(0, 10):
    print(f"[{x+1}] Nombre: {productosMasBuscados[x][0][:45]}")
    print(f"Numero de busquedas: {productosMasBuscados[x][1]}\n")
```

Lo que sigue es mostrar el top mejores y peores ventas por cada categoría.

Primero obtengo las distintas categorías existentes en la lista lifeStore_products y estas se almacenan en la lista categorías.

```python
#obteniendo las distintas categorias
for categoria in lifestore_products:
    if categoria[3] not in categorias:
        categorias.append(categoria[3])
```

Con el primer for se iteran las categorías obtenidas anteriormente, se compara esa categoría con la categoría de cada venta en la lista cantidad_ventas, si la categoría coincide se almacena el producto en la lista ventas_categoria. Una vez que ya se obtuvieron todas las ventas de la categoría actual se ordena la lista ventas_categoria de menor a mayor guiándose por la cantidad de ventas y se imprimen los primeros cinco valores de la lista los cuales son los cinco productos con peores ventas de esa categoria

```python
#Obteniendo los productos menos vendidos por categoria
#por cada categoria se recorre la lista cantidad ventas, si coincide la categoria se almacena en
#una lista distinta
print("\n\t\tPor categoria:")
print("_____________________________")
print("\n--------- 5 productos con menores ventas---------\n ")
for cat in categorias:
    #limpiar la lista temporal
    ventasCategoria =[]
    print(f" Categoria: {cat}")
    for producto in cantidad_ventas:
        #Si las categorias coinciden se almacena en la lista temporal
        if producto[2] == cat:
            ventasCategoria.append(producto)
    #Acomodando la lista de menor a mayor ventas
    ventasCategoria = sorted(ventasCategoria, key=lambda x: x[1])
    #imprimir los productos
    for x in range(0,len(ventasCategoria)):
        if x == 5:
            break
        else:
            print(f"[{x + 1}] producto: {ventasCategoria[x][0][:45]}")
            print(f"cantidad de ventas: {ventasCategoria[x][1]}")
```

Con el primer for se iteran las categorías obtenidas anteriormente, se compara esa categoría con la categoría de cada producto en la lista cantidad_busquedas, si la categoría coincide se almacena el producto en la lista busquedascategoria. Una vez que ya se obtuvieron todas las busquedas de la categoría actual se ordena la lista busquedacategoria de menor a mayor guiándose por la cantidad de busquedas y se imprimen los primeros cinco valores de la lista los cuales son los cinco productos con menores búsquedas de esa categoria

```
# Obteniendo los productos con menos busquedas
# por cada categoria se recorre la lista cantidad_busquedas, si coincide la categoria se almacena en
# una lista distinta

print("\n----------  10 productos con menores busquedas ---------\n ")
for cat in categorias:
    # limpiar la lista temporal
    busquedasCategoria = []
    print(f"Categoria: {cat}\n ")
    for producto in cantidad_busquedas:
        # Si las categorias coinciden se almacena en la lista temporal
        if producto[2] == cat:
            busquedasCategoria.append(producto)
    # Acomodando la lista de menor a mayor busquedas
    busquedasCategoria = sorted(busquedasCategoria, key=lambda x: x[1])
    # imprimir los productos
    for x in range(0, len(busquedasCategoria)):
        if x == 10:
            break
        else:
            print(f"[{x + 1}] producto: {busquedasCategoria[x][0][:45]}")
            print(f"cantidad de ventas: {busquedasCategoria[x][1]}")
int("                                                                    ")
```

3.4 Productos por reseña en el servicio.

Estas opciones se encuentran dentro de la función parte2()

Variables y listas utilizadas en esta función.

```
resenias = [] # (nombre prodcuto, promedio de reseñas) solo si existen reseñas
productosMejoresResenias = [] #Guarda los cinco productos con mejores reseñas
productosPeoresResenias = [] #guarda los cinco productos con peores reseñas

sumaScore=0                 #suma las reseñas de cada prodcuto
count=0                     #cantidad de reseñas que tiene cada producto
promedio=0                  #Se promedian las reseñas por producto
```

Existen dos for anidados el primero recorre la lista llifestore_products y el segundo recorre lifestore_sales cada id de los productos se compara con el id_ producto de la lista ventas si son iguales se le suma el valor de score de esta venta especifica a la variable sumaScore y se le aumenta uno a count, una vez que ya se comparo el producto con todas las ventas se promedia el score y se almacena el nombre de producto y promedio de reseñas en una lista.

```
#Se recorre cada uno de los productos existentes
for producto in lifestore_products:
    sumaScore=0
    count=0
    #Si existe el id de produto en la lista de ventas se suma su score
    for ventas in lifestore_sales:
        if producto[0] == ventas[1]:
            sumaScore+=ventas[2]
            count+=1
    #Se promedia el score de cada producto y se almacena en la lista resenia siempre y cuando existan reseñas
    if count > 0:
        promedio = float(sumaScore / count)
        resenias.append([producto[1],promedio])
```

Una vez que ya se almacenaron todos los productos con el promedio de reseñas, se acomoda la lista resenias de menor a mayor basandose en el promedio de reseñas y se copian los datos de las mejores y peores reseñas en sus respectivas listas.

```
 # Se ordenan los elementos de menor a mayor basandonos en el promedio de las reseñas
resenias = sorted(resenias, key=lambda x: x[1])
# se guarda en una lista nueva los ultimos diez elementos de la lista cantidad_busquedas, los cuales son los productos mas bu
productosMejoresResenias = resenias[len(resenias) - 5: len(resenias)]
# se guarda en una lista nueva los ultimos diez elementos de la lista cantidad_busquedas, los cuales son los productos mas bu
productosPeoresResenias = resenias[0: 5]
```

Por ultimo se imprimen los resultados esperados.

```
print("_____________________________________________________________-")
print("\t\t\t Top productos ")
print("_____________________________________________________________-")
print("5 Productos con mejores reseñas: \n")
for x in range(0,len(productosMejoresResenias)):
    print(f"\nProducto{x+1}: {productosMejoresResenias[x][0]}")
    print(f" Promedio de reseñas: {productosMejoresResenias[x][1]}")

print("\n\n5 Productos con peores reseñas: ")
print("\n")
for x in range(0, len(productosPeoresResenias)):
    print(f"\nProducto{x + 1}: {productosPeoresResenias[x][0]}")
    print(f" Promedio de reseñas: {productosPeoresResenias[x][1]}")

print("_____________________________________________________________")
```

3.5 Total de ingresos y ventas

Variables y listas utilizadas.

```
meses =["01","02","03","04","05","06","07","08","09","10","11","12"]
years=[]                # almacena los distintos años encontrados en las fechas de las ventas
ventasProductos =[]      # (precio producto vendido, fecha, refund)
ingresoMensual=0        #Total de los ingresos mensuales
promedioMensual=0       #promedio de las vents de cada mes
countVentasMensuales = 0     #la cantidad de ventas que se hicieron por mes para lograr sacar el promedio
totalAnual=0
ventaMes=[]             #Guardara la venta total de cada mes y se alamcenará en mesesConMasVentas[]
```

Al igual que en los puntos anteriores es necesario juntar los datos de dos listas distintas en este caso lifestore_products y lifestores_sales.

```
#Se obtienen los valores para la lista ventasProductos
for producto in lifestore_products:
    for ventas in lifestore_sales:
        if producto[0] == ventas[1]:
            ventasProductos.append([producto[2],ventas[3],ventas[4]])
```

Como en esta parte del programa se mostraran datos distintos de cada año, decidí hacer el programa de una forma en la que funcione para cualquier cantidad de años y no solo con los datos que se nos dio, así que en un alista almaceno todos los distintos años encontrados en las ventas hechas, en este caso solo son dos años: 2019 y 2020.

Para lograr mi objetivo se recorre la lista creada anteriormente, la cual tiene almacenada la fecha de cada venta. Primero divide la fecha con ayuda de la función split la cual regresa una lista con tres posiciones. Día, mes y año. Toma el año y si no existe ese año en la lista years lo agrega.

Esto obtendrá los distintos años siempre y cuando se siga el mismo formato de fecha en proximos registros.

Por ultimo se ordenan los valores de la lista years de menor a mayor para tener los años en orden.

```
#Se obtienen los distintos años de las fechas
for ventas in ventasProductos:
    #Year es una variable temporal
    # se divide la fecha de cada venta y se almacenan en 'year' year = [dia,mes,año]
    year = str(ventas[1]).split('/')
    #si la fecha que se está comparando no se encuentre dentro de la lista years se agrega
    if year[2] not in years:
        years.append(year[2])
years.sort()
```

Una vez que ya se obtuvieron los distintos años se cran tres for anidados, uno para los distintos años, otro para los distintos meses y el ultimo recorre la lista ventasProducto. Se obtiene la fecha de cada venta y se separa con la función Split, si el año y el mes de la venta coinciden con el año y mes de la iteración actual de los for y no existe una devolución, se suma el valor de la venta actual a la variable ingresoMensual la cual nos ayudara a saber el total de las ventas que se obtuvieron en cada mes y también el total de ventas de cada año.

```
for anio in years:
    totalAnual=0
    ventaMes = []
    print("\n\t Año: ",anio)
    for mes in meses:
        ingresoMensual=0
        countVentasMensuales=0
        print("\n\t mes: ", mes)
        for venta in ventasProductos:
            temp = str(venta[1]).split('/')
            #Si el año y mes coinciden con los que se estan comparando actualmente
            # y además no tiene devolución se suma el costo del objeto
            if temp[1] == mes and temp[2] == anio and venta[2] != 1:
                ingresoMensual +=  venta[0]
                countVentasMensuales+=1
                #Si el mes tuvo ventas se saca el promedio de ventas mensuales
        if(countVentasMensuales >0):
            promedioMensual = float(ingresoMensual / countVentasMensuales)
        totalAnual +=ingresoMensual
```

Se imprimen los distintos valores obtenidos y se igualan a cero las variables ingresoMensual y promedioMensual ya que serán utilizadas para guardar la información del siguiente mes en la próxima iteración del for meses.

```
            print("Ingreso Mensual: $",ingresoMensual)
            print("Cantidad de ventas",countVentasMensuales)
            if(countVentasMensuales >0):
                print("Ventas promedio del mes: $", promedioMensual)
            else:
                print("Ventas promedio del mes: 0")
            ventaMes.append([mes,ingresoMensual])
```

La lista ventaMes contiene el mes y su cantidad de ventas total del año que se está iterando, se ordena de menor a mayor, se muestra el top tres meses con mejores ventas y el total de ventas de ese año.

```
    ventaMes.append([mes,ingresoMensual])

#se ordenan los valores de venta mes para que esten de menor a mayor cantidad de ventas
ventaMes = sorted(ventaMes, key=lambda x: x[1])
print("\n \t Top tres meses con mejores ventas del año: ")
print("[1] Mes: ",ventaMes[-1][0], " Ventas: $",ventaMes[-1][1])
print("[2] Mes: ",ventaMes[-2][0], " Ventas: $",ventaMes[-2][1])
print("[3] Mes: ",ventaMes[-3][0], " Ventas: $",ventaMes[-3][1])
print(f"Total de ventas del año {anio}: $",totalAnual)
print("_____________________________________________")
```

4 conclusión

Gracias a los resultados obtenidos en el programa sabemos cuáles son los productos que se están acumulando en el inventario y cuales son los que tienen una menor cantidad de búsqueda, esto nos ayudará a crear una solución con ayuda de machine learning para lograr que esos productos logren una mayor cantidad de vistas lo cual provocará que aumenten sus ventas y por ende se libere el inventario.

En las estadísticas y porcentajes obtenidos se puede observar que la disminución en la cantidad de búsquedas está directamente relacionada con la disminución de ganancias.

5. Resultados

Log in

```
 Ingrese usuario: qwe
Ingrese Password: qwe


¡Usuario incorrecto!
Contraseña Incorrecta

 Ingrese usuario: user
Ingrese Password: 122


Contraseña Incorrecta

 Ingrese usuario: user
Ingrese Password: user
------------------------------------------------------------------------
                         Bienvenido
------------------------------------------------------------------------
```

Menú

```
  MENU
 [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
 [4] Terminar

 Selecciona una opcion:
```

Productos más vendidos y productos rezagados

Cinco productos con mayores ventas.

```
 ----------Cinco Productos con mayores ventas----------

[1] Nombre: SSD Kingston A400, 120GB, SATA III, 2.5'', 7m
Numero de ventas: 50

[2] Nombre: Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz,
Numero de ventas: 42

[3] Nombre: Procesador Intel Core i3-9100F, S-1151, 3.60G
Numero de ventas: 20

[4] Nombre: Tarjeta Madre ASRock Micro ATX B450M Steel Le
Numero de ventas: 18

[5] Nombre: SSD Adata Ultimate SU800, 256GB, SATA III, 2.
Numero de ventas: 15
```

Diez productos con mayores búsquedas.

```
 ----------Diez Productos con mayores Busquedas----------

[1] Nombre: SSD Kingston A400, 120GB, SATA III, 2.5'', 7m
Numero de busquedas: 263

[2] Nombre: SSD Adata Ultimate SU800, 256GB, SATA III, 2.
Numero de busquedas: 107

[3] Nombre: Tarjeta Madre ASUS micro ATX TUF B450M-PLUS G
Numero de busquedas: 60

[4] Nombre: Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz,
Numero de busquedas: 55

[5] Nombre: Procesador AMD Ryzen 3 3200G con Gráficos Rad
Numero de busquedas: 41

[6] Nombre: Logitech Audífonos Gamer G635 7.1, Alámbrico,
Numero de busquedas: 35

[7] Nombre: TV Monitor LED 24TL520S-PU 24, HD, Widescreen
Numero de busquedas: 32

[8] Nombre: Procesador Intel Core i7-9700K, S-1151, 3.60G
Numero de busquedas: 31

[9] Nombre: SSD XPG SX8200 Pro, 256GB, PCI Express, M.2
```

Por categoria:

5 productos con menores ventas

```
 ---------------- Categoria: tarjetas de video--------------------

--------- 5 productos con menores ventas---------

[1] producto: Tarjeta de Video EVGA NVIDIA GeForce GT 710,
cantidad de ventas: 0
[2] producto: Tarjeta de Video EVGA NVIDIA GeForce GTX 1660
cantidad de ventas: 0
[3] producto: Tarjeta de Video EVGA NVIDIA GeForce RTX 2060
cantidad de ventas: 0
[4] producto: Tarjeta de Video Gigabyte NVIDIA GeForce GTX
cantidad de ventas: 0
[5] producto: Tarjeta de Video Gigabyte NVIDIA GeForce RTX
cantidad de ventas: 0
```

10 productos con menores búsquedas.

```
----------  10 productos con menores busquedas ---------

[1] producto: Tarjeta de Video EVGA NVIDIA GeForce GT 710,
cantidad de ventas: 0
[2] producto: Tarjeta de Video EVGA NVIDIA GeForce RTX 2060
cantidad de ventas: 0
[3] producto: Tarjeta de Video Gigabyte NVIDIA GeForce GTX
cantidad de ventas: 0
[4] producto: Tarjeta de Video Gigabyte NVIDIA GeForce RTX
cantidad de ventas: 0
[5] producto: Tarjeta de Video MSI Radeon X1550, 128MB 64 b
cantidad de ventas: 0
[6] producto: Tarjeta de Video PNY NVIDIA GeForce RTX 2080,
cantidad de ventas: 0
[7] producto: MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, P
cantidad de ventas: 1
[8] producto: Tarjeta de Video VisionTek AMD Radeon HD5450,
cantidad de ventas: 1
[9] producto: Tarjeta de Video Asus NVIDIA GeForce GTX 1050
cantidad de ventas: 2
[10] producto: Tarjeta de Video Gigabyte AMD Radeon R7 370 O
cantidad de ventas: 3
```

Productos por reseña en el servicio

5 productos con mejores reseñas

```
5 Productos con mejores reseñas:

Producto1: Kit Memoria RAM Corsair Dominator Platinum DDR4, 3200MHz, 16GB (2x 8GB), Non-ECC, CL16, XMP
 Promedio de reseñas: 5.0

Producto2: TCL Smart TV LED 55S425 54.6, 4K Ultra HD, Widescreen, Negro
 Promedio de reseñas: 5.0

Producto3: TV Monitor LED 24TL520S-PU 24, HD, Widescreen, HDMI, Negro
 Promedio de reseñas: 5.0

Producto4: Logitech Audífonos Gamer G332, Alámbrico, 2 Metros, 3.5mm, Negro/Rojo
 Promedio de reseñas: 5.0

Producto5: Logitech Audífonos Gamer G635 7.1, Alámbrico, 1.5 Metros, 3.5mm, Negro/Azul
 Promedio de reseñas: 5.0
```

5 productos con peores reseñas.

```
5 Productos con peores reseñas:

Producto1: Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2GB 256-bit GDDR5, PCI Express 3.0
 Promedio de reseñas: 1.0

Producto2: Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, Intel H110, 32GB DDR4, para Intel
 Promedio de reseñas: 1.0

Producto3: Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
 Promedio de reseñas: 1.8333333333333333

Producto4: Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2, S-1151, Intel H110, 32GB DDR4 para Intel
 Promedio de reseñas: 2.0

Producto5: Cougar Audífonos Gamer Phontum Essential, Alámbrico, 1.9 Metros, 3.5mm, Negro.
 Promedio de reseñas: 3.0
------------------------------------------------------------
```

Total de ingresos y ventas

Ventas por mes.

```
     Año:  2020

     mes:  01
Ingreso Mensual: $ 117738
Cantidad de ventas 52
Ventas promedio del mes: $ 2264.1923076923076

     mes:  02
Ingreso Mensual: $ 107270
Cantidad de ventas 40
Ventas promedio del mes: $ 2681.75

     mes:  03
Ingreso Mensual: $ 162931
Cantidad de ventas 49
Ventas promedio del mes: $ 3325.122448979592

     mes:  04
Ingreso Mensual: $ 191066
Cantidad de ventas 74
Ventas promedio del mes: $ 2581.972972972973
```

Resultados anuales.

```
     Top tres meses con mejores ventas del año:
[1] Mes:  04  Ventas: $ 191066
[2] Mes:  03  Ventas: $ 162931
[3] Mes:  01  Ventas: $ 117738
Total de ventas del año 2020: $ 737916
------------------------------------------------
```

6. Repositorio

https://github.com/danielGuzmanPerez/Emtech.git

7. Código

```
from lifestore_file import lifestore_products, lifestore_sales,
lifestore_searches


def login():
    # usuario: user
    # contrasenia: user
    intentos = 0
    correcto = False
    while (not correcto):
        if intentos == 3:
            print("Intentos superados, ¡adios!")
            quit()
        usuario = input("\n Ingrese usuario: ")
        password = input("Ingrese Password: ")
        if usuario == "user" and password == "user":

print("_____________________________________________________________________
_-")
            print("\t \t \t \t \t \t Bienvenido")

print("_____________________________________________________________________
_-")
            correcto = True
        else:
            print("\n")
            if usuario != "user":
                print("¡Usuario incorrecto!")
            if password != "user":
                print ("Contraseña Incorrecta")
            intentos +=1

def parte1():
    cantidad_ventas= []             # (nombre_producto, numero de ventas,
categoria)
    productosMasVendidos=[]         # contiene los cinco productos con
mas ventas
    cantidad_busquedas = []         #( nombre_producto, numero de
busquedas, categoria)
    productosMasBuscados=[]         #contiene los diez productos mas
buscados
    categorias=[]                   #contiene las distintas categorias de
los productos
    ventasCategoria=[]              #contiene los productos de cada
categoria
    busquedasCategoria=[]           #Contiene los productos de cada
categoria
    count = 0
    print("_____________________________________________________")
```

```
    print("\n\n \t \t Opción 1 Productos más vendidos y productos
rezagados")
    print("_____________________________________________________")

    #Creando la lista de cantidad ventas (nombre_producto, numero de
ventas, categoria)
    #Se recorre primero la lista de productos para comparar cada producto
con todas las ventas que hay
    for producto in lifestore_products:
        count=0
        for ventas in lifestore_sales:
            #Se cuenta la cantidad de veces que el id de un producto
aparece en la lista de ventas
            if producto[0] == ventas[1] :
                count +=1
        #Se agrega el valor a la lista nueva
        cantidad_ventas.append([producto[1],count,producto[3]])
    # Se ordenan los elementos de menor a mayor basandonos en el numero
de ventas posicion 1
    cantidad_ventas = sorted(cantidad_ventas, key= lambda x: x[1] )
    # se guarda en una lista nueva los ultimos cinco elementos de la
lista cantidad_ventas, los cuales son los productos mas vendidos
    productosMasVendidos= cantidad_ventas[len(cantidad_ventas)-5 :
len(cantidad_ventas)]

    # Creando la lista de cantidad_busquedas (nombre_producto, numero de
busquedas, categoria)
    #Busquedas de productos
    for producto in lifestore_products:
        count = 0
        for busquedas in lifestore_searches:
            if producto[0] == busquedas[1]:
                count +=1
        cantidad_busquedas.append([producto[1],count, producto[3]])
    # Se ordenan los elementos de menor a mayor basandonos en el numero
de busquedas posicion 1
    cantidad_busquedas = sorted(cantidad_busquedas, key=lambda x: x[1])
    # se guarda en una lista nueva los ultimos diez elementos de la lista
cantidad_busquedas, los cuales son los productos mas buscados
    productosMasBuscados = cantidad_busquedas[len(cantidad_busquedas) -
10: len(cantidad_busquedas)]
    productosMasVendidos.reverse()
    productosMasBuscados.reverse()
    print("\n ----------Cinco Productos con mayores ventas----------\n")
    for x in range(0,5):
        print(f"[{x+1}] Nombre: {productosMasVendidos[x][0][:45]}")
        print(f"Numero de ventas: {productosMasVendidos[x][1]}\n")


    print("\n ----------Diez Productos con mayores Busquedas----------
\n")
    for x in range(0, 10):
        print(f"[{x+1}] Nombre: {productosMasBuscados[x][0][:45]}")
```

```
        print(f"Numero de busquedas: {productosMasBuscados[x][1]}\n")

    #obteniendo las distintas categorias
    for categoria in lifestore_products:
        if categoria[3] not in categorias:
            categorias.append(categoria[3])

    #Obteniendo los productos menos vendidos por categoria
    #por cada categoria se recorre la lista cantidad ventas, si coincide
la categoria se almacena en
    #una lista distinta
    print("\n\t\tPor categoria:\n\n")

    for cat in categorias:
        #limpiar la lista temporal
        ventasCategoria =[]
        print(f" ---------------- Categoria: {cat}--------------------")
        for producto in cantidad_ventas:
            #Si las categorias coinciden se almacena en la lista temporal
            if producto[2] == cat:
                ventasCategoria.append(producto)
        #Acomodando la lista de menor a mayor ventas
        ventasCategoria = sorted(ventasCategoria, key=lambda x: x[1])
        #imprimir los productos
        print("\n--------- 5 productos con menores ventas---------\n ")
        for x in range(0,len(ventasCategoria)):
            if x == 5:
                break
            else:
                print(f"[{x + 1}] producto:
{ventasCategoria[x][0][:45]}")
                print(f"cantidad de ventas: {ventasCategoria[x][1]}")


        # Obteniendo los productos con menos busquedas
        # por cada categoria se recorre la lista cantidad_busquedas, si
coincide la categoria se almacena en
        # una lista distinta

        print("\n----------  10 productos con menores busquedas ---------
\n ")
       ## for cat in categorias:
        # limpiar la lista temporal
        busquedasCategoria = []
        #print(f"\nCategoria: {cat}\n ")
        for producto in cantidad_busquedas:
             # Si las categorias coinciden se almacena en la lista
temporal
            if producto[2] == cat:
                busquedasCategoria.append(producto)
        # Acomodando la lista de menor a mayor busquedas
        busquedasCategoria = sorted(busquedasCategoria, key=lambda x:
x[1])
```

```
        # imprimir los productos
        for x in range(0, len(busquedasCategoria)):
            if x == 10:
                break
            else:
                print(f"[{x + 1}] producto:
{busquedasCategoria[x][0][:45]}")
                print(f"cantidad de ventas: {busquedasCategoria[x][1]}")
    print("_____________________________________________________________________")
```

```
def parte2():
    resenias = [] # (nombre prodcuto, promedio de reseñas) solo si
existen reseñas
    productosMejoresResenias = [] #Guarda los cinco productos con mejores
reseñas
    productosPeoresResenias = [] #guarda los cinco productos con peores
reseñas

    sumaScore=0                 #suma las reseñas de cada prodcuto
    count=0                     #cantidad de reseñas que tiene cada
producto
    promedio=0                  #Se promedian las reseñas por producto

    #Se recorre cada uno de los productos existentes
    for producto in lifestore_products:
        sumaScore=0
        count=0
        #Si existe el id de produto en la lista de ventas se suma su
score
        for ventas in lifestore_sales:
            if producto[0] == ventas[1]:
                sumaScore+=ventas[2]
                count+=1
        #Se promedia el score de cada producto y se almacena en la lista
resenia siempre y cuando existan reseñas
        if count > 0:
            promedio = float(sumaScore / count)
            resenias.append([producto[1],promedio])

     # Se ordenan los elementos de menor a mayor basandonos en el
promedio de las reseñas
    resenias = sorted(resenias, key=lambda x: x[1])
    # se guarda en una lista nueva los ultimos diez elementos de la lista
cantidad_busquedas, los cuales son los productos mas buscados
    productosMejoresResenias = resenias[len(resenias) - 5: len(resenias)]
```

```
    # se guarda en una lista nueva los ultimos diez elementos de la lista
cantidad_busquedas, los cuales son los productos mas buscados
    productosPeoresResenias = resenias[0: 5]
    print("_____-
")
    print("\t\t\t Top productos ")
    print("_____-
")
    print("5 Productos con mejores reseñas: ")
    for x in range(0,len(productosMejoresResenias)):
        print(f"\nProducto{x+1}: {productosMejoresResenias[x][0]}")
        print(f" Promedio de reseñas: {productosMejoresResenias[x][1]}")

    print("\n\n5 Productos con peores reseñas: ")
    for x in range(0, len(productosPeoresResenias)):
        print(f"\nProducto{x + 1}: {productosPeoresResenias[x][0]}")
        print(f" Promedio de reseñas: {productosPeoresResenias[x][1]}")

    print("_____")
```

```
def parte3():
    meses =["01","02","03","04","05","06","07","08","09","10","11","12"]
    years=[]                # almacena los distintos años encontrados en
las fechas de las ventas
    ventasProductos =[]      # (precio producto vendido, fecha, refund)
    ingresoMensual=0        #Total de los ingresos mensuales
    promedioMensual=0       #promedio de las vents de cada mes
    countVentasMensuales = 0     #la cantidad de ventas que se hicieron
por mes para lograr sacar el promedio
    totalAnual=0
    ventaMes=[]             #Guardara la venta total de cada mes y se
alamcenará en mesesConMasVentas[]
    #Se obtienen los valores para la lista ventasProductos
    for producto in lifestore_products:
        for ventas in lifestore_sales:
            if producto[0] == ventas[1]:
                ventasProductos.append([producto[2],ventas[3],ventas[4]])

    #Se obtienen los distintos años de las fechas
    for ventas in ventasProductos:
        #Year es una variable temporal
        # se divide la fecha de cada venta y se almacenan en 'year' year
= [dia,mes,año]
        year = str(ventas[1]).split('/')
        #si la fecha que se está comparando no se encuentre dentro de la
lista years se agrega
        if year[2] not in years:
            years.append(year[2])
```

```
    years.sort()

print("_____________________________________________________________________
_-")
    print("\t\t\t Ventas")

print("_____________________________________________________________________
_-")
        #comparo el mes de cada venta con cada mes de cada distinto año
para obtener las ventas mensuales
    for anio in years:
        totalAnual=0
        ventaMes = []
        print("\n\t Año: ",anio)
        for mes in meses:
            ingresoMensual=0
            countVentasMensuales=0
            print("\n\t mes: ", mes)
            for venta in ventasProductos:
                temp = str(venta[1]).split('/')
                #Si el año y mes coinciden con los que se estan
comparando actualmente
                # y además no tiene devolución se suma el costo del
objeto
                if temp[1] == mes and temp[2] == anio and venta[2] != 1:
                    ingresoMensual +=  venta[0]
                    countVentasMensuales+=1
                    #Si el mes tuvo ventas se saca el promedio de ventas
mensuales
            if(countVentasMensuales >0):
                promedioMensual = float(ingresoMensual /
countVentasMensuales)
            totalAnual +=ingresoMensual
            print("Ingreso Mensual: $",ingresoMensual)
            print("Cantidad de ventas",countVentasMensuales)
            if(countVentasMensuales >0):
                print("Ventas promedio del mes: $", promedioMensual)
            else:
                print("Ventas promedio del mes: 0")
            ventaMes.append([mes,ingresoMensual])

        #se ordenan los valores de venta mes para que esten de menor a
mayor cantidad de ventas
        ventaMes = sorted(ventaMes, key=lambda x: x[1])
        print("\n \t Top tres meses con mejores ventas del año: ")
        print("[1] Mes: ",ventaMes[-1][0], " Ventas: $",ventaMes[-1][1])
        print("[2] Mes: ",ventaMes[-2][0], " Ventas: $",ventaMes[-2][1])
        print("[3] Mes: ",ventaMes[-3][0], " Ventas: $",ventaMes[-3][1])
        print(f"Total de ventas del año {anio}: $",totalAnual)
        print("_____________________________________________________")
```

```
def menu():
    opcion= 0
    while(opcion !=4):
        print("\n\n MENU ")
        print("[1] Productos más vendidos y productos rezagados")
        print("[2] Productos por reseña en el servicio")
        print("[3] Total de ingresos y ventas promedio mensuales, total
anual y meses con más ventas al año")
        print("[4] Terminar")
        opcion= int(input("\nSelecciona una opcion: "))
        if opcion == 1:
            parte1()
        elif opcion == 2:
            parte2()
        elif opcion == 3:
            parte3()
        elif opcion == 4:
            quit()
        else:
            print("\n opcion incorrecta")



login()
menu()
```