



EMTECH

GRUPO DATATON

PROYECTO 1

***Introducción a
Python***

**Luis Manuel
García Guerrero**

Febrero 2022

Índice

0.1	Introducción	1
0.2	Definición del código	2
0.2.1	Importación de datos de la lista y datos de acceso	2
0.2.2	Listas de búsquedas y reseñas	2
0.2.3	Listas con y sin búsquedas	3
0.2.4	Listas mayor y menor búsquedas	4
0.2.5	Listas mayores ventas, reseñas mayores y menores	5
0.2.6	Listas cálculo de ventas por mes y anual, mayor a menor	7
0.2.7	Login	8
0.2.8	Menú y submenú para imprimir resultados	8
0.3	Solución al problema	10
0.4	Conclusión	11

1 Introducción

La primera parte de curso, se ha desarrollado un primer análisis sobre un caso particular donde se han puesto en práctica métodos principales sobre programación en Python.

El nombre de la tienda física y virtual es LifeStore esta tiene una gran cantidad de artículos. El departamento de gerencia de ventas solicito al departamento de Ciencia de datos un análisis básico de los datos de los productos basado en sus ventas y búsquedas para evitar acumulación de inventario.

Mostraremos los resultados obtenidos y las propuestas dadas para su revisión por el departamento de ventas y el departamento de Marketing.

2 Definición del código

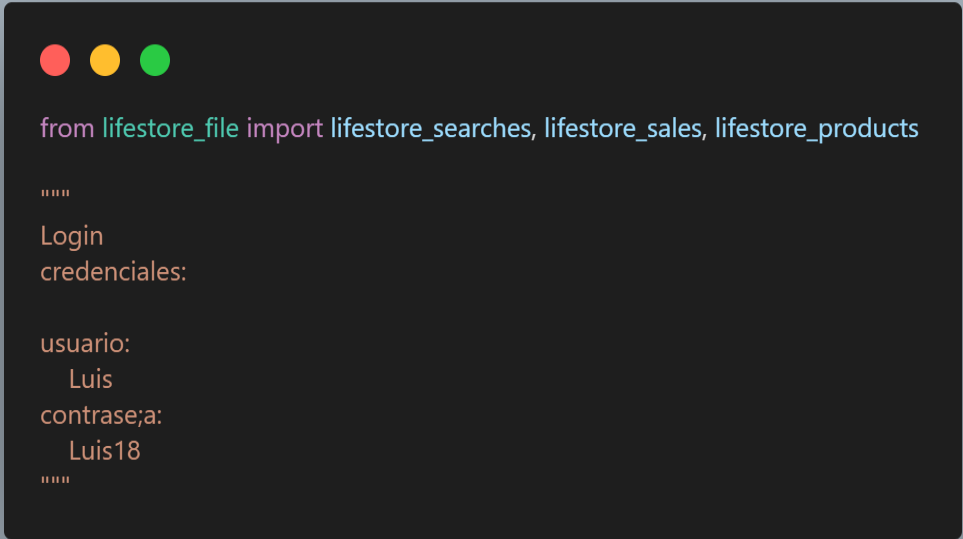
2.1.1 Importación de listas y datos de acceso

Se importan las listas:

- lifestore_searches
- lifestore_sales
- lifestore_products

Estos se importan del archivo lifestore_file.py

Se dan los accesos para su correcto ingreso al sistema de reporte de ventas.




```
from lifestore_file import lifestore_searches, lifestore_sales, lifestore_products

"""
Login
credenciales:

usuario:
    Luis
contrase;a:
    Luis18
"""
```

2.1.2 Crear listas para contar las búsquedas y reseñas.




```

contador_busquedas = []
contador_reseñas = []
for n in range(total_productos):
    contador_busquedas.append([n+1,0,lifestore_products[n][1]])
    contador_reseñas.append([n+1,0,0,lifestore_products[n][1]])
    for busqueda in lifestore_searches:
        if busqueda[1] == contador_busquedas[n][0]:
            contador_busquedas[n][1] += 1
    for busqueda in lifestore_sales:
        if busqueda[1] == contador_reseñas[n][0]:
            contador_reseñas[n][1] += busqueda[2]
            contador_reseñas[n][2] += 1

```

2.1.3 Se separan los productos de los que sí tienen búsquedas de los que no.



```

sin_busquedas = []
con_busquedas = []
con_busquedas_copy = []
for cuenta in contador_busquedas:
    if cuenta[1] != 0:
        con_busquedas.append(cuenta)
        con_busquedas_copy.append(cuenta)
    else:
        sin_busquedas.append(cuenta)
ord_busqueda_less = []
ord_busqueda_more = []

```

2.1.4 Se ordenan los valores de menor búsqueda a mayor búsqueda

```
while con_busquedas:
    minimo = con_busquedas[0][1]
    busquedaes_actual = con_busquedas[0]
    for busqueda in con_busquedas:
        if busqueda[1] < minimo:
            minimo = busqueda[1]
            busquedaes_actual = busqueda
    ord_busqueda_less.append(busquedaes_actual)
    con_busquedas.remove(busquedaes_actual)
#Se ordenan los valores de mayor búsqueda a menor
while con_busquedas_copy:
    maximo = con_busquedas_copy[0][1]
    busquedaes_actual = con_busquedas_copy[0]
    for busqueda in con_busquedas_copy:
        if busqueda[1] > maximo:
            maximo = busqueda[1]
            busquedaes_actual = busqueda
    ord_busqueda_more.append(busquedaes_actual)
    con_busquedas_copy.remove(busquedaes_actual)
vendidos = []
vendidos_copy = []
promedio = []
promedio_copy = []
sin_registro = []
for score in contador_reseñas:
    if score[2] != 0:
        vendidos.append([score[0],score[2],score[3]])
        vendidos_copy.append([score[0],score[2],score[3]])
        promedio.append([score[0], round(score[1]/score[2],2),score[3]])
        promedio_copy.append([score[0], round(score[1]/score[2],2),score[3]])
    else:
        sin_registro.append([score[0], "Sin registro de venta"])
```

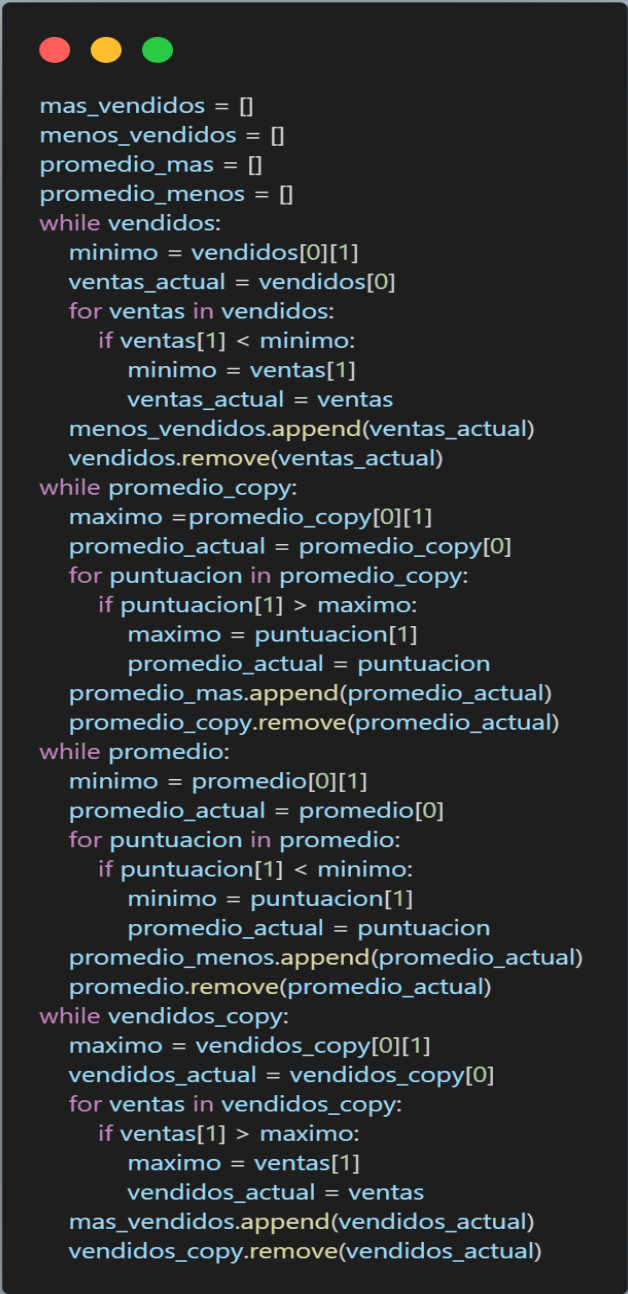
Para obtener los productos más buscados y productos menos buscados, se tomaron en cuenta los productos que aparecieron en la lista búsquedas y la cantidad de veces que fueron buscados, finalmente, se obtuvo la lista ordenada top búsquedas la cual contiene los nombres de los productos y número de veces que fueron buscados en la por los clientes. De esta lista, se tomaron los primeros 10 productos como los más buscados y los últimos 10 de menor búsqueda.

2.1.5 Se ordenan productos más vendidos, reseña de mayor a menor y al revés

Para obtener el listado de los productos más vendidos, primero se tuvo que contar todas las ventas, para posteriormente obtener la cantidad de veces un producto fue vendido. Para obtener los productos mejor calificados y productos peor calificados se tomaron en cuenta los productos vendidos de la lista sales, donde se tomaron las calificaciones que los usuarios le asignaron y se generó el diccionario.

to best rated el cual contiene como claves los valores del 1 al 5 (calificaciones) y cada uno de ellos contiene una lista de los productos con esa calificación y las veces que la recibieron. Por lo tanto, el top 20 de los productos con mejor reseña muestra el nombre de los productos con calificaciones más altas y por otro lado, los productos con peor reseña muestra los productos con calificaciones más bajas.

Adicionalmente, se obtuvieron también los productos que fueron reembolsados y la cantidad de veces que esto ocurrió.



```

mas_vendidos = []
menos_vendidos = []
promedio_mas = []
promedio_menos = []
while vendidos:
    minimo = vendidos[0][1]
    ventas_actual = vendidos[0]
    for ventas in vendidos:
        if ventas[1] < minimo:
            minimo = ventas[1]
            ventas_actual = ventas
    menos_vendidos.append(ventas_actual)
    vendidos.remove(ventas_actual)
while promedio_copy:
    maximo = promedio_copy[0][1]
    promedio_actual = promedio_copy[0]
    for puntuacion in promedio_copy:
        if puntuacion[1] > maximo:
            maximo = puntuacion[1]
            promedio_actual = puntuacion
    promedio_mas.append(promedio_actual)
    promedio_copy.remove(promedio_actual)
while promedio:
    minimo = promedio[0][1]
    promedio_actual = promedio[0]
    for puntuacion in promedio:
        if puntuacion[1] < minimo:
            minimo = puntuacion[1]
            promedio_actual = puntuacion
    promedio_menos.append(promedio_actual)
    promedio.remove(promedio_actual)
while vendidos_copy:
    maximo = vendidos_copy[0][1]
    vendidos_actual = vendidos_copy[0]
    for ventas in vendidos_copy:
        if ventas[1] > maximo:
            maximo = ventas[1]
            vendidos_actual = ventas
    mas_vendidos.append(vendidos_actual)
    vendidos_copy.remove(vendidos_actual)

```

2.1.6 Se crean listas para el cálculo de ventas por mes y anual, también se ordenan de mayor a menor.

```
por_dia = []
for sale in lifestore_sales:
    por_dia.append([lifestore_products[sale[1]-1][2],sale[3][3:],sale[4]])
lista_mes = []
lista_año = []
for mes in por_dia:
    if not mes[1] in lista_mes:
        lista_mes.append(mes[1])
    if not mes[1][-4:] in lista_año:
        lista_año.append(mes[1][-4:])
contador_ventas_mensual = []
for mes in lista_mes:
    contador_ventas_mensual.append([mes, 0])
for sale in por_dia:
    for mes in contador_ventas_mensual:
        if sale[1] == mes[0]:
            mes[1] += sale[0]
contador_ventas_anual = []
for año in lista_año:
    contador_ventas_anual.append([año,0,0])
for sale in contador_ventas_mensual:
    for año in contador_ventas_anual:
        if sale[0][-4:] == año[0]:
            año[1] += sale[1]
            año[2] += 1
meses_mayor_ventas = []
while contador_ventas_mensual:
    maximo = contador_ventas_mensual[0][1]
    vendidos_actual = contador_ventas_mensual[0]
    for ventas in contador_ventas_mensual:
        if ventas[1] > maximo:
            maximo = ventas[1]
            vendidos_actual = ventas
    meses_mayor_ventas.append(vendidos_actual)
    contador_ventas_mensual.remove(vendidos_actual)
meses = ["Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre"]
```

La cantidad total (en pesos) de ingresos que la tienda registro en sus ventas fue de 760,177. Esto se obtuvo de sumar el precio de todos los productos vendidos de la lista sales.

Por otra parte, los ingresos anuales se obtuvieron tomando en cuenta los años de las fechas de los productos vendidos registrados en la lista sales, donde únicamente se obtuvieron 2 años: 2019 y 2020 con ingresos de 8,408 y 751,769 pesos respectivamente.

En el código, la lista mensual sales muestra el total de ingresos mensuales, esto se obtuvo tomando en cuenta las ventas por mes y el precio de cada producto, datos tomados de la lista sales; finalmente se obtuvo que el mes con mayores ingresos fue abril con un total de 193,295 pesos.

2.1.7 Se crea un sistema de login con varios intentos para ingresar con un ciclo while

[illegible]

2.1.8 Ingresar al submenú de listas de ventas de productos, se consulta la info de la lista de los 5 productos más vendidos y 10 más buscados. También se consulta la información de 5 productos menos vendidos y 10 con menos búsquedas. Los 5 mejores productos con mejores reseñas y 5 con peores reseñas. Y por ultimo el total de ingresos al año, las ventas totales mensuales y los meses con mayores ventas.

```

eleccion = 0
if opcion.isnumeric():
    if opcion == "1": # Ingresa al submenu Listoa ventas de productos.
        while eleccion != "3":
            print("<<<<< " + (usuario) + """, bienvenido al Menú del reporte de ventas >>>>>")
            [1] Lista de los productos más vendidos y más buscados.
            [2] Lista de los productos menos vendidos y menos buscados.
            [3] Regresar
            """)
            eleccion = input("Ingresa una opción: ")
            if eleccion == "1": # Se consulta la info de la lista de los 5 lproductos más vendidos y más buscados.
                # Productos más vendidos
                print("{} más vendidos:".format(10))
                for n in range(5):
                    print("{}>2s). {}>3s} piezas se vendieron de {}".format(str(n+1),str(mas_vendidos[n][1]),mas_vendidos[n][2].split(",")[0]))
                # Productos más buscados
                print("{} más buscados:".format(10))
                for n in range(10):
                    print("{}>2s). {}>3s} busquedas de {}".format(str(n+1),str(ord_busqueda_more[n][1]),ord_busqueda_more[n][2].split(",")[0]))
            elif eleccion == "2": #Se consulta la info de la lista de los 5 lproductos menos vendidos y menos buscados.
                # Productos menos vendidos
                print("{} menos vendidos:".format(5))
                for n in range(5):
                    print("{}>2s). {}>3s} piezas se vendieron de {}".format(str(n+1),str(menos_vendidos[n][1]),menos_vendidos[n][2].split(",")[0]))
                # Productos menos buscados
                print("{} menos buscados:".format(10))
                for n in range(10):
                    print("{}>2s). {}>3s} busquedas de {}".format(str(n+1),str(ord_busqueda_less[n][1]),ord_busqueda_less[n][2].split(",")[0]))
            elif eleccion == "3": # Con este se regresa al menu anterior.
                continue
            else:
                print("Ingresa una opción valida")
        elif opcion == "2":
            #Productos con mejores reseñas
            print("{} Mejores reseñas:".format(20))
            for n in range(5):
                print("{}>2s). Calificación {}>2s)/5 de {}".format(str(n+1),str(promedio_mas[n][1]),promedio_mas[n][2].split(",")[0]))

            # Productos con peores reseñas
            print("{} Peores reseñas:".format(5))
            for n in range(5):
                print("{}>2s). Calificación {}>2s)/5 de {}".format(str(n+1),str(promedio_menos[n][1]),promedio_menos[n][2].split(",")[0]))

    elif opcion == "3":
        #Ventas anuales y mensuales, promedio de ventas anuales y mensuales
        print("Ventas Anuales")
        for año in contador_ventas_anual:
            print("{}: $ {},{}.00".format(año[0],str(año[1])[-3:],str(año[1])[-3:]))
        print("Promedio mensual")
        for año in contador_ventas_anual:
            promedio_mensual = str(round(año[1]/año[2],2))
            print("{}: $ {}".format(año[0],promedio_mensual))
        print("Ventas por mes")
        indice = 0
        for mes in meses_mayor_ventas:
            indice += 1
            a = int(mes[0][:2])-1
            print("{}>2s). {}>10s {}: $ {},{}.00".format(str(indice),meses[a],mes[0][-4:],str(mes[1])[-3:],str(mes[1])[-3:]))

    elif opcion == "4": #Con esta te sales del menu y cierra programa.
        print("Sesion terminada. "+ (usuario) + " nos vemos. Esperamos su feedback para poder mejorar el reporte en futuras ocasiones")
        break

    else:
        print("Opción no valida.")

```

3 Solución del problema

Tomando en cuenta los puntos del reporte de ventas, podríamos llevar a cabo las siguientes acciones:

Primeramente, los gerentes de ventas deberían ver que productos son que tienen mayor y menor ranking en sus categorías tanto de ventas, búsquedas y reseñas. Entonces vemos que en el análisis podemos encontrar por lo menos 7 productos los cuales están los ID: 9, 10, 27, 35, 45, 59 y 70 que fueron los que menos se buscaron en los últimos meses. Estos productos tenemos dos alternativas quitarlos del inventario para que no generen perdidas o ofrecerlos con estrategias de descuentos o envíos gratis.

Para reponer las perdidas de los productos menos vendidos y buscados podemos aumentar el precio de los productos más vendidos ya que esto no será una diferencia muy grande ya que los clientes en estos productos no ven la diferencia al comprar.

En los meses con mejores ventas ofrecer descuentos en productos mejores vendidos y incentivar la compra de los productos menos vendidos con puntos o descuentos más grandes para incentivar su compra. Hacer venta cruzada y poder ofrecer artículos menos vendidos con descuentos atractivos en la venta de algún otro artículo.

Como en los supermercados podríamos poner los productos menos vendidos en la entrada o en las primeras secciones para que estén a la vista y los clientes puedan adquirirlos más. Esto lo podríamos hacer en un ecommerce para recomendar los productos de acuerdo a la segmentación de los clientes.

Además de hacer promociones en las que se ofrezca productos menos vendidos para incentivar su compra por lo clientes.

Con los productos con menos reseñas y con menos ventas podemos ofrecer envío gratis o con descuento en los meses donde menos se vende. Este es un ejercicio iterativo entonces se sugiere hacer este ejercicio mes con mes o por lo menos cada 3 meses para poder tener mayor rango de análisis y poder tener un margen de visión más amplio para proponer mejores estrategias. Así mejorarán sus inventarios y su flujo, sino también mejorarán sus expectativas de crecimiento que permitirán un mayor beneficio económico a largo plazo para la empresa. Por último, sería mejor si se trabaja en colaboración con el departamento de Data Science

para poder hacer segmentación de nuestros clientes y poder hacer sugerencia más personalizadas.

4 Conclusión

Como conclusión del análisis y como complemento de las propuestas sobre cómo dar solución al problema, se espera que la tienda aumente para los próximos 2 meses al menos un 25% sus ingresos.

Con base a lo propuesto en las propuestas se espera que la tienda aumente sus ventas un 20% por mes y 25% anual.

Se debe de hacer una colaboración de intercambio de información entre los departamentos para poder llevar al siguiente nivel el análisis con Ciencia de Datos. Y con esto lo más importante conocer mejor a sus clientes y sus necesidades para poder mantenerlos satisfechos con el servicio y con los productos.

