



---

# CASO LIFESTORE: IMPORTANCIA DEL DATA SCIENCE

---

Miroslava Silva Oviedo



14 DE SEPTIEMBRE DE 2021

Facilitador: Jaime Saúl Alonso Sánchez

Grupo 1: “Pythonistas”

## **Tabla de contenido**

<b>INTRODUCCIÓN .....</b>	<b>2</b>
<b>DEFINICIÓN DEL CÓDIGO .....</b>	<b>2</b>
<b>SOLUCIÓN AL PROBLEMA.....</b>	<b>6</b>
<b>CONCLUSIÓN .....</b>	<b>7</b>

# CASO LIFESTORE: IMPORTANCIA DEL DATASCIENCE

Miroslava Silva Oviedo

## INTRODUCCIÓN

Por medio de la presente se busca implementar los conocimientos adquiridos en la unidad de introducción a Python mediante un caso de estudio en la cual se presentará su realización mediante el uso de operadores lógicos y condicionales para la clasificación de la información con la finalidad de presentar la importancia del uso de Data Science.

### Caso de estudio

LifeStore es una tienda virtual que maneja una amplia gama de artículos, recientemente, la Gerencia de ventas, se percató que la empresa tiene una importante acumulación de inventario. Asimismo, se ha identificado una reducción en las búsquedas de un grupo importante de productos, lo que ha redundado en una disminución sustancial de sus ventas del último trimestre.

Se busca el realizar un análisis de la rotación de productos identificando los siguientes elementos:

- 1) Productos más vendidos y productos rezagados a partir del análisis de las categorías con menores ventas y categorías con menores búsquedas.
- 2) Productos por reseña en el servicio a partir del análisis de categorías con mayores ventas y categorías con mayores búsquedas.
- 3) Sugerir una estrategia de productos a retirar del mercado, así como sugerencia de como reducir la acumulación de inventario considerando los datos de ingresos y ventas mensuales.

## DEFINICIÓN DEL CÓDIGO

En principio, se procede a introducir los datos que se buscan analizar, del cual se definieron previamente tres variables con la información requerida a modo de lista, como se muestra a continuación.

```
1. """
2. This is the LifeStore_SalesList data:
3. lifestore_searches = [id_search, id product]
4. lifestore_sales = [id_sale, id_product, score (from 1 to 5), date, refund (1 for true or 0 to
   false)]
5. lifestore_products = [id_product, name, price, category, stock]
6. """
```

Se definen las variables que servirán para dar formato a la entrada de las listas.

```
1. salto=' '*35
2. salto1=' '*20
3. vineta='✧'
4. bienvenida = """Bienvenido
5.
```

```

6. Actualmente estas en el listado de LifeStore. \n """
7. linea=("◀..."+ "-"*80 + "▶")
8. print(linea)
9. print(salto+bienvenida)
10. print(linea)
11. usuario1="Gerente"
12. contraseña2="LifeStore"

```

Después se procede con el código para crear el login de usuario, al no ser especificado por el contratista tanto usuario como contraseña fueron supuestos. Se comienza con un ciclo while para introducir el nombre de usuario, del cual se procede con un if, del cual seguirá pidiendo el nombre de usuario hasta que sea correcto.

Al momento de introducir el nombre de usuario correcto pasará nuevamente a un ciclo while donde pedirá al usuario introducir la contraseña, de la cual tendrá tres intentos de los cuales se irán almacenando en la variable m.

```

1. n=0
2. while n==0:
3.     usuario=input("Ingrese nombre de usuario: ")
4.     print(linea)
5.     if usuario==usuario1:
6.         n=1 #Para poder cortar el ciclo while de usuario1
7.         m=1 #Para contar # de veces que ingreso contraseña
8.         while n==1:
9.             contraseña=input("Contraseña: ")
10.            if contraseña==contraseña2:
11.                print(linea)
12.                n=2
13.                '\n'
14.                print("""La contraseña es correcta.
15.                Bienvenido de nuevo """+ usuario1)
16.                print("Ingrese el número correspondiente a la categoría de interés.")
17.            elif contraseña != contraseña2 and m<3:
18.                '\n'
19.                print("La contraseña es incorrecta. Inténtelo de nuevo")
20.                n=1
21.                m+=1
22.            else:
23.                "\n"
24.                print("Ha superado el número de intentos, no puede ingresar.")
25.                n=2
26.        else:
27.            '\n'
28.            print("El nombre de usuario no existe. Inténtelo de nuevo")
29.            n=0 #Cuando n=0 volverá a pedir ingresar nombre de usuario

```

### 30. `print(linea)`

Se prosigue con la creación del menú, del cual se presentará la numeración de cada una de las listas y se le pedirá al usuario ingresar el número de la lista de interés.

```
1. print("""Ir a:
2. 1. Lista de productos con mayores ventas
3. 2. Lista de productos con menores ventas
4. 3. Lista de productos más buscados
5. 4. Lista de productos menos buscados
6. 5. Lista de productos con mejores reseñas
7. 6. Lista de productos con peores reseñas
8. """)
9. print(linea)
10. menu=int(input("Ingrese número: ")) #Escribir número del menú
```

Se procede a realizar el cálculo de ventas por producto, del cual se realizará fuera de un ciclo, siendo que será repetido para el calculo tanto de mayor como menores ventas de producto.

```
1. #Mayores ventas
2. producto_ventas = []
3. for producto in lifestore_products:
4.     lista_interna = [producto[0], 0]
5.     producto_ventas.append(lista_interna)
6. for venta in lifestore_sales:
7.     producto_vendido = venta[1]
8.     devuelto = venta[4]
9.     if devuelto == 0:
10.         validez = True
11.     else:
12.         validez = False
13.     if validez == True:
14.         producto_ventas[producto_vendido - 1][1] += 1
15. def segundo_elemento(lista):
16.     return lista[1]
```

Se comienza con un if para indicar el número al cual corresponde esta lista y se realiza el ordenamiento necesario para imprimir en la pantalla los valores pedidos. Siendo el mismo caso para el de menores ventas.

```
1. if menu == 1:
2.     producto_ventas.sort(reverse=True, key=segundo_elemento)
3.     n = 15
4.     mas_vendidos = producto_ventas[:n]
5.     for mv in mas_vendidos:
6.         print(mv)
```

```

7. elif menu == 2:
8.     #Creación de lista productos menos vendidos
9.     producto_ventas.sort(reverse=False, key=segundo_elemento)
10.    m=5
11.    menos_vendidos = producto_ventas[:m]
12.    for mnv in menos_vendidos:
13.        print(mnv)

```

Se siguió el mismo procedimiento al anterior para el calculo de búsquedas. La diferencia en este caso es que no fue necesario el cálculo de las devoluciones.

```

1. producto_busquedas=[]
2. for producto1 in lifestore_products:
3.     listainterna2=[producto1[0],0]
4.     producto_busquedas.append(listainterna2)
5. #Calculo cantidad de busquedas
6. for busquedas in lifestore_searches:
7.     producto_buscado=busquedas[1]
8.     producto_busquedas[producto_buscado- 1][1] += 1
9. def segundo_elemento(listal):
10.    return listal[1]
11.
12. if menu == 3:
13.     #Creación de lista productos más buscados
14.     producto_busquedas.sort(reverse=True, key=segundo_elemento)
15.     o=20
16.     mas_buscados = producto_busquedas[:o]
17.     for mv in mas_buscados:
18.         print(mv)
19. elif menu == 4:
20.     #Lista productos menos buscados
21.     producto_busquedas.sort(key=segundo_elemento)
22.     o=20
23.     menos_buscados = producto_busquedas[:o]
24.     for mnv in menos_buscados:
25.         print(mnv)

```

Se realizo el mismo procedimiento para el cálculo de las reseñas

```

1. #reseñas
2. producto_reseña = []
3. for producto2 in lifestore_products:
4.     lista_interna2 = [producto2[0], 0]
5.     producto_reseña.append(lista_interna2)
6. for venta1 in lifestore_sales:
7.     producto_sale1 = venta1[1]

```

```

8.  num_reseña = venta1[2]
9.  producto_reseña[producto_sale1 - 1][1] += num_reseña
10. def segundo_elemento(lista):
11.     return lista[1]
12.
13. if menu == 5:
14.     #Creación de lista productos mejores reseñas
15.     producto_reseña.sort(reverse=True, key=segundo_elemento)
16.     o=20
17.     mejor_reseña = producto_reseña[:o]
18.     for mr in mejor_reseña:
19.         print(mr)
20. elif menu == 6:
21.     #Lista productos menos buscados
22.     producto_reseña.sort(key=segundo_elemento)
23.     o=20
24.     peor_reseña = producto_reseña[:o]
25.     for mn in peor_reseña:
26.         print(mn)

```

## SOLUCIÓN AL PROBLEMA

A partir del análisis de productos con mayores ventas presentadas a continuación podemos tener una estimación de los productos más interesantes para el publico y de los cuales se recomendaría el tener más en stock.

```

----->
Ingrese número: 1
[54, 49]
[3, 42]
[5, 20]
[42, 18]
[57, 15]
[4, 13]
[29, 13]
[2, 12]
[47, 11]
[12, 9]
[48, 9]
[7, 7]
[44, 6]
[18, 5]
[8, 4]

```

Para los productos menos buscados se tienen los siguientes, de los cuales lo primordial sería evitar tener más en stock hasta tener la venta como tal y realizar una difusión para evitar que se queden por más tiempo.

```

-----
Ingrese número: 4
[14, 0]
[16, 0]
[19, 0]
[20, 0]
[23, 0]
[24, 0]
[30, 0]
[32, 0]
[33, 0]
[34, 0]
[36, 0]
[37, 0]
[38, 0]
[41, 0]
[43, 0]
[53, 0]
[55, 0]
[58, 0]
[60, 0]
[61, 0]

```

A partir de las reseñas se puede observar los productos con mejor y peores reseñas, de las cuales para las primeras se debe buscar el no perder la calidad que ya se tiene; mientras que en estos últimos sería la evaluación de los mismos, buscando los defectos que tiene tomando las opiniones del consumidor.

<pre> ----- Ingrese número: 5 [54, 236] [3, 202] [5, 94] [42, 82] [57, 73] [4, 58] [29, 58] [2, 55] [47, 50] [12, 43] [48, 42] [7, 35] [44, 28] [18, 22] [8, 20] [6, 15] [11, 15] [49, 15] [51, 14] [31, 11] </pre>	<pre> ----- Ingrese número: 6 [9, 0] [14, 0] [15, 0] [16, 0] [19, 0] [20, 0] [23, 0] [24, 0] [26, 0] [27, 0] [30, 0] [32, 0] [34, 0] [35, 0] [36, 0] [37, 0] [38, 0] [39, 0] [41, 0] [43, 0] </pre>
---	---

## CONCLUSIÓN

Se ha podido observar el como el Data Science es esencial para el análisis de datos, del cual de manera sencilla se puede realizar un programa para facilitar la búsqueda de soluciones a cierto tipo de problemáticas conforme se presenten en una empresa y/o industria. Mediante este caso se pudo comprobar este hecho.

Link: <https://github.com/mirous09/Proyecto-final.git>