

Proyecto 1

ANALISIS DE DATASETS CON PYTHON

Bruno Yael Silva Morales | Data Science EMTECH | 15 / 09 / 2021

Índice de Contenido

Introducción	2
Descripción del caso.....	2
Definición del código	3
Solución al problema.....	9
Conclusión específica	10
Conclusión General.....	10

Introducción

Se requiere desarrollar un programa en para la clasificación y el análisis de los datos proporcionados en los “*DataSets*” de “LifeStore”. Este programa debe dar a conocer la información requerida.

Descripción del caso

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.

Ilustración 1 Texto de la descripción del caso, obtenido de las instrucciones del reporte final.

Definición del código

- En la ilustración 2 “Introducción al Script”, se muestra una la sección del código, la breve portada del script, se muestran los tres “DataSets” proporcionados para el ejercicio de los que extrae y analiza la información, además de la primera sentencia de control que es el “LogIn” para utilizar el sistema.

```

"""
Created on 03/09/2021

Proyecto 01, entrega 15 Septiembre 2021
@author: Bruno Yael Silva Morales / EMTECH / Data Science
"""

"""
El proyecto empieza en la línea 1946, las siguientes
contienen los DataSet proporcionados para su funcionamiento
"""

"DataSet de PRODUCTOS proporcionado"
lifestore_products = [

]

"DataSet de VENTAS proporcionado"
lifestore_sales = [

]

"DataSet de BUSQUEDAS proporcionado"
lifestore_searches = [

]

#Usuarios registrados para utilizar el sistema
usuarios=["BRUNO","SILVA","JAIME","JIMMY","ALONSO","LAPLACE","GAUSS","JORDAN","EMTECH"]

#Se solicita al usuario que introduzca su credencial de acceso
user = input("Ingresa tu nombre de usuario (Si eres JIMMY escribe: 'JIMMY'): ")

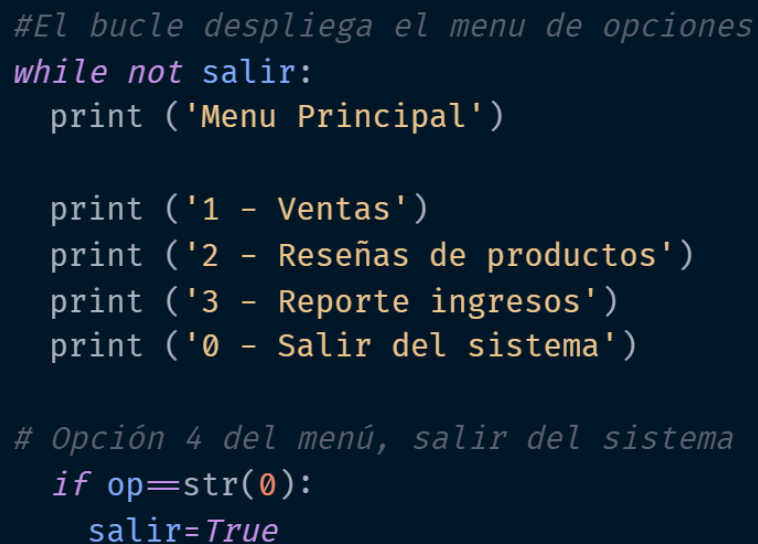
#Se busca lo que se ingresa a la terminal, si coincide con los usuarios guardados, entra al bucle
while
if user in usuarios:
    print(f'Usuario {user} encontrado, holi!')
    print(f'{user}, teclea la opcion que desees del menu:')
    salir = False

    #El bucle despliega el menu de opciones
    while not salir:
        print('Menu Principal')

#Si el usuario ingresado no está registrado nos saca del sistema
else:
    print("Usuario no encontrado, solicite al departamento de Sistemas que lo registre")
    print("Gracias")

```

- En la Ilustración 3 “Menú principal” se muestra el ciclo del menú principal, cuya función es mantenerte en el sistema y preguntarte que es lo que deseas visualizar.



```
#El bucle despliega el menu de opciones
while not salir:
    print ('Menu Principal')

    print ('1 - Ventas')
    print ('2 - Reseñas de productos')
    print ('3 - Reporte ingresos')
    print ('0 - Salir del sistema')

# Opción 4 del menú, salir del sistema
if op==str(0):
    salir=True
```

- En las ilustraciones 4 y 5 “Productos más vendidos y productos rezagados” se muestran los ciclos sentencias que generan las listas de los productos con mas y menores ventas, los productos con más y menos búsquedas.

```

op = input('Opcion: ') #Se lee la opcion ingresada

### Opción 1 del menu, muestra lo referente a ventas ###
if op == str(1):

    ventas = []
    search = []
    categoria = []
    categorias = []
    categoria1 = []

    #Ciclo for para iterar los productos
    for producto in lifestore_products:
        producto_venta = [producto[0], 0]
        producto_busca = [producto[0], 0]
        producto_categoria = [producto[0], 0, producto[3]]
        ventas.append(producto_venta)
        search.append(producto_busca)
        categorias.append(producto_categoria)
    #Ciclo for para iterar las ventas
    for venta in lifestore_sales:
        producto_vendido = venta[1]
        reembolso = venta[4]
        if reembolso == 0:
            devuelto = True
        else:
            devuelto = False

        if devuelto == True:
            ventas[producto_vendido-1][1] += 1
    #Ciclo for para iterar las busquedas
    for busca in lifestore_searches:
        producto_buscado = busca[1]
        search[producto_buscado-1][1] += 1

    #Funcion menos vendidos
    ordenados = sorted(ventas, key=lambda ordenados : ordenados[1])
    print(f'Los 5 articulos menos vendidos son: ')
    for m in range(57,62):
        print(m+1, ordenados[m])
        categoria.append(ordenados[m])

    print(f'Los 5 articulos categorizados menos vendidos son: ')
    a=0
    for cate in categoria:
        for catego in categorias:
            if cate[0]==catego[0]:
                if cate[0] not in categoria1:
                    a+=1
                    print(a, catego[2])
                    categoria1.append(catego[2])
    #Termina función menos vendidos

    #Función mas vendidos
    ordenadosr = sorted(ventas, key=lambda ordenadosr : ordenadosr[1], reverse=True)
    print(f'Los 15 articulos vendidos de mayor venta son: ')
    for n in range(15):
        print(n+1, ordenadosr[n])
    #Termina función mas vendidos

```

```

#Inicia funcion mas buscados
buscadosr = sorted(search, key=lambda buscadosr : buscadosr[1], reverse=True)
buscadosrm = sorted(search, key=lambda buscadosrm : buscadosrm[1])
print(f'Los 20 articulos de mayor busqueda son: ')
for t in range(20):
    print(t+1, buscadosr[t])
b=1
print(f'Los 20 articulos de menor busqueda son: ')
for t in range(60):
    if buscadosrm[t][1]≠0:
        print(b, buscadosrm[t])
        b+=1

for producto in lifestore_products:
    if producto[3] not in categorias:
        categorias.append(producto[3])

categoria_productos=[]
categoria_de_lista=[]

#Inicio de función productos en venta
print(f'Total de productos en venta :{len(ordenados)}')

#Inicio TOP producto en venta
maximo = ventas[0][1]
for i in range(len(ventas)):
    if ventas[i][1] > maximo:
        maximo = ventas[i][1]
print(f'El articulo más vendido se vendió:{maximo} veces',input('Enter para continuar'))
#Fin TOP ventas

```

- En la ilustración 6 “Productos por reseña en el servicio”, se muestran los ciclos y sentencias del código que generan las listas de los productos con las mejores reseñas y de los productos con las peores reseñas, considerando aquellos que se devolvieron.

```

### Opción 2 del menu, Productos por reseña en el servicio ###
if op == str(2):

    categorias = []
    for producto in lifestore_sales:
        if producto[2] not in categorias:
            categorias.append(producto[2])
    print(f'Las reseñas son: {categorias}')
    reseña_productos = []
    for reseña in categorias:
        res_prods = [reseña, []]
        reseña_productos.append(res_prods)

    lista_adicional=[]
    for lista in reseña_productos:#lifestore_products
        reseña_de_lista = lista[0] #id_products
        for producto in lifestore_sales:#ventas
            id_del_producto = producto[1]#id_products
            reseña_del_producto = producto[2]#score
            if reseña_del_producto == reseña_de_lista:
                lista[1].append(id_del_producto)
            if lista[0]==5: #Score más alto
                mejor= True
            else:
                mejor=False

            if mejor==True:
                for productos in lista[1]:
                    if productos not in lista_adicional:
                        lista_adicional.append(productos)

    print(f'Los productos con la mejor reseña {lista[0]} son: {lista_adicional}')
    #Reseñas mas bajas
    if lista[0]==3 or lista[0]==2 or lista[0]==1:
        peor= True
    else:
        peor=False
    if peor==True:
        print(f'Los productos con la peor reseña {lista[0]} son: {lista[1]}')

```


- En la ilustración 7 “Total de ingresos y ventas promedio mensuales, total anual y meses con más ventas al año” se muestran los ciclos y sentencias que dan un breve resumen financiero del total de meses registrados en los datasets, arrojando como resultado:

El total de ingresos, las ventas promedio mensuales, ingreso total anual y los meses con más ventas al año.

```

### Opción 3 del menu, resumen de ingresos ###
if op == str(3):

    ventas_totales=0
    ventas_totales_dev=0
    lista_p = []
    precio_productos = []
    for precio in lifestore_products:
        pre_prods = [precio[0], precio[2]]
        precio_productos.append(pre_prods)
    print(f'Lista de artículos [id producto] con su precio [precio]: {precio_productos}')#si,
    producto, precio

    lista_venta_fecha=[]
    lista_venta_fecha_dev=[]
    for producto_v in lifestore_sales:
        id_del_producto_p = producto_v[1]
        id_fecha_p = producto_v[3]
        reembolso_p=producto_v[4]
        for lista_p in precio_productos:
            precio_de_lista = lista_p[0]
            precio_del_producto = lista_p[1]
            if id_del_producto_p == precio_de_lista and reembolso_p==0:
                ventas_totales+=lista_p[1]
                lista_venta_fecha.append(id_fecha_p[3:5])
            if id_del_producto_p == precio_de_lista and reembolso_p==1:
                ventas_totales_dev+=lista_p[1]
                lista_venta_fecha_dev.append(id_fecha_p[3:5])

    ordenados = sorted(lista_venta_fecha, key=lambda mes : mes)

    meses_totales=[]
    meses = lista_venta_fecha
    maxx=meses[0]
    ene=0
    feb=0
    mar=0
    abr=0
    may=0
    jun=0
    jul=0
    ago=0
    sep=0
    oct=0
    nov=0
    dic=0
    total_meses=0
    año=[]
    mes_mas=[]
    for x in range(len(lista_venta_fecha)):
        if meses[x]=='01':
            ene+=1
        if meses[x]=='02':
            feb+=1
        if meses[x]=='03':
            mar+=1
        if meses[x]=='04':
            abr+=1
        if meses[x]=='05':
            may+=1
        if meses[x]=='06':
            jun+=1
        if meses[x]=='07':
            jul+=1
        if meses[x]=='08':
            ago+=1
        if meses[x]=='09':
            sep+=1
        if meses[x]=='10':
            oct+=1
        if meses[x]=='11':
            nov+=1
        if meses[x]=='12':
            dic+=1

        if meses[x]> maxx:
            maxx = meses[x]
        if meses[x] not in meses_totales:
            meses_totales.append(meses[x])
            total_meses+=1

    if meses[x]> maxx:
        maxx = meses[x]
    meses_totales.sort()
    promedio_mensual=(ventas_totales/total_meses)
    print(f'Ingreso Total: ${ventas_totales}')
    print(f'Total de devoluciones: ${ventas_totales_dev}')
    print(f'Total de meses con ventas: {total_meses}')
    print(f'Meses con ventas: {meses_totales}')
    print(f'Ingreso promedio mensual: ${promedio_mensual}')
    print(f'Los meses con mayores ventas son:')
    print(f'Enero:[ene], Febrero:[feb], Marzo:[mar], Abril:[abr], Mayo:[may]')
    print(f'Ingreso Total anual: ${promedio_mensual * total_meses}')

```

Solución al problema

Estrategia sugerida para la tienda “*LifeStore*”, de acuerdo con el análisis de datos, se tienen las siguientes listas:

```

Los 5 articulos menos vendidos son:
58 [58, 0]
59 [59, 0]
60 [60, 0]
61 [61, 0]
62 [62, 0]
Los 5 articulos categorizados menos vendidos son:
1 discos duros
2 discos duros
3 memorias usb
4 memorias usb
5 pantallas
Los 15 articulos vendidos de mayor venta son:
1 [94, 1]
2 [1, 0]
3 [2, 0]
4 [3, 0]
5 [4, 0]
6 [5, 0]
7 [6, 0]
8 [7, 0]
9 [8, 0]
10 [9, 0]
11 [10, 0]
12 [11, 0]
13 [12, 0]
14 [13, 0]
15 [14, 0]

```

```

coLos 20 articulos de mayor busqueda son:
1 [54, 263]
2 [57, 107]
3 [29, 60]
4 [3, 55]
5 [4, 41]
6 [85, 35]
7 [67, 32]
8 [7, 31]
9 [5, 30]
10 [47, 30]
11 [48, 27]
12 [44, 25]
13 [2, 24]
14 [42, 23]
15 [8, 20]
16 [12, 15]
17 [21, 15]
18 [66, 15]
19 [18, 11]
20 [51, 11]
Los 20 articulos de menor busqueda son:
1 [9, 1]
2 [10, 1]
3 [27, 1]
4 [35, 1]
5 [45, 1]
6 [59, 1]
7 [70, 1]
8 [80, 1]
9 [93, 1]
10 [13, 2]
11 [56, 2]
12 [76, 2]
13 [91, 2]
14 [17, 3]
15 [39, 3]
16 [95, 3]
17 [15, 4]
18 [46, 4]
19 [63, 4]
20 [73, 4]
Total de productos en venta :96

```

En ellas están enumerados los artículos, indicando [Numero de articulo, numero de eventos], también se muestran las categorías de artículos menos vendidos. El mismo formato se realizó par las búsquedas.

Dado lo anterior, se sugiere retirar los artículos que no tienen ventas registradas, los que tienen un bajo número de búsquedas.

Sugerencia, venderlos por lote.

Conclusión específica

Se sugiere que Lifestore siga las recomendaciones indicadas en la solución del problema.

Conclusión General

Este es un sistema que se puede emplear en cualquier tipo de negocios para evitar remanentes de productos además de tener breves estados financieros de los ingresos.