PROYECTO 1

Introducción a Python

Arturo Ledezma Mejia

Contenido

Introducción

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.

Consignas:

Derivado de la situación, la Gerencia de Ventas te solicita que realices un análisis de la rotación de productos identificando los siguientes elementos.

1. Categorías con menores ventas y categorías con menores búsquedas.
2. Categorías con mayores ventas y categorías con mayores búsquedas.
3. Sugerir una estrategia de productos a retirar del mercado así como sugerencia de cómo reducir la acumulación de inventario considerando los datos mensuales.

Definición del Código

Login de Usuarios

Utilizamos la librería gestpass para ocultar el pass al momento de escribirlo.

```
# Script login
print("Iniciar Sesión")

#Utilizamos librería getas para ocultar el pass
import getpass

#Configuramos usuario y pass
Usr_Name = "Arturo"
Usr_Password = "1234"

#creamos variable login que desencadenara el while
login = 'true'
while (login == 'true'):

#solicitamos el nombre de usuario
    username = input("Usuario: ")
    #sí el usuario es correcto continuamos, de lo contrario mandamos mensaje
de usuario incorrecto e intentar nuevamente
    if (username == Usr_Name):
        login1 = 'true'
        #solicitamos el password y comparamos con el permitido siempre y
cuando el usuario sea correcto
        while (login1 == 'true'):
            password = getpass.getpass("Password: ")
            if (password == Usr_Password):
                print("Bienvenido " + username)
                login = 'false'
                login1 = 'false'
                acceso="autorizado"
                # en caso de que sean correctas las credenciales enviamos
confirmación de autorizado y la variable login permanecerá en false para
detener el while
            else:
              #en caso de que las credenciales sean incorrectas enviamos
mensaje de error y de volver a intentar
                print("Password incorrecto!, intente de nuevo")
    else:
        print("Usuario incorrecto!, intente de nuevo")

#la variable acceso determina el mensaje de bienvenida y continuidad al menú
if(acceso=="autorizado"):
    print("""
           Comencemos con el caso
           """)
```

Menú

Mediante un while se implementa un menú que solo se mostrara sí las credenciales del usuario son correctas y la opción de ver reportes se mantenga confirmada con una variable opcion

```
#mientras se cumplan las condiciones de que sea un usuario autorizado quien selecciona ver el menu  el bucle while se mostrara
   while opcion=="s" and login=='false':
      print("Selecciona el número del reporte que deseas: ")
      print("""
          """)
      print("Top 50 Ventas Productos    [1]")
      print("Bottom 50 Ventas productos [2]")
      print("Los 100 MAS buscados       [3]")
      print("Los 100 MENOS buscados     [4]")
      print("Las mejores reseñas =)     [5]")
      print("Las peores reseñas  =(     [6]")
      print("Ingresos al mes y Promedio [7]")
      print("Ingreso Anual y mejor mes  [8]")
      print("Salir                      [9]")

      print("""
          """)
          #dependiendo de la opción seleccionada se mostrara o ejecutara el reporte
      seleccion=input("ingresa una opción: ")
      print("""
          """)
```

Importación de datasets y creación de DataFrames

Utilizando la librería de pandas converti las listas a dataframes para facilitar la unión de los registros, los dataframes inician con df_ también asigne nombre a los encabezados de cada columna para llamar a cada variable por el nombre de encabezado

```
#LIBRERIAS DEL PROYECTO
import getpass
import pandas as pd



#IMPORTAR LISTAS
from lifestore_file import
lifestore_products,lifestore_sales,lifestore_searches

#asignando titulos de encabezado de columna

dflifestore_products = pd.DataFrame(lifestore_products,
columns=['id_producto', 'Descripcion', 'precio', 'categoria', 'stock'])
dflifestore_sales = pd.DataFrame(lifestore_sales, columns=['id_venta',
'id_producto', 'score', 'fecha', 'devolucion'])
dflifestore_searches = pd.DataFrame(lifestore_searches,
columns=['id_busqueda', 'id_producto'])

##unir dataframes

df_product_sales = pd.merge(dflifestore_sales, dflifestore_products,
on='id_producto', how='left')

#crear dataframe con el conteo de productos en venta
df2_product_sales=df_product_sales.groupby(['Descripcion'])[['id_producto']].
count()

#ordenar de forma descendente las ventas
df2_product_sales.sort_values('id_producto', ascending=False,inplace=True)

#cambiando el encabezado de la columna
df2_product_sales.rename(columns={ 'id_producto': 'Frecuencia'},
inplace=True)

#pasamos el top a un listado limitado a 50 productos con mas venta
```

```
top_50 = df2_product_sales[:50]

## Por categoría, generar un listado con los 50 productos con menores ventas
df3_product_sales=df_product_sales.groupby(['Descripcion','categoria'])[['id_
producto']].count()

#ordenar de forma descendente las ventas
df3_product_sales.sort_values('id_producto', ascending=True,inplace=True)
#cambiando el encabezado de la columna
df3_product_sales.rename(columns={ 'id_producto': 'Frecuencia'},
inplace=True)

#pasamos el bottom categorias a un listado limitado a 50 registros
bottom_50_cat_item = df3_product_sales[:50]

##para el listado con los 100 productos con mayor búsquedas.
##unir dataframes
df_product_searches = pd.merge(dflifestore_searches, dflifestore_products,
on='id_producto', how='left')

#crear dataframe con el conteo de productos en venta
df2_product_searches=df_product_searches.groupby(['Descripcion'])[['id_produc
to']].count()

#ordenar de forma descendente las ventas
df2_product_searches.sort_values('id_producto', ascending=False,inplace=True)
#cambiando el encabezado de la columna
df2_product_searches.rename(columns={ 'id_producto': 'Frecuencia'},
inplace=True)
#pasamos el top de los mas buscados a un listado limitado a 100
top_100busquedas = df2_product_searches[:100]



##para el listado con los 100 productos con menores búsquedas.
df3_product_searches=df_product_searches.groupby(['Descripcion'])[['id_produc
to']].count()

#ordenar de forma descendente las ventas
df3_product_searches.sort_values('id_producto', ascending=True,inplace=True)
#cambiando el encabezado de la columna
df3_product_searches.rename(columns={ 'id_producto': 'Frecuencia'},
inplace=True)
```

```
bottom_100busquedas = df3_product_searches[:100]

#Listado 20 productos con mejores reseñas.
#agrupamos los score por descripción e id
bestscore_product_sales=df_product_sales.groupby(['Descripcion','score'])[['i
d_producto']].count()

#ordenamos
bestscore_product_sales.sort_values(['score','id_producto'],
ascending=[False,False],inplace=True)

#cambiando el encabezado de la columna
bestscore_product_sales.rename(columns={'Descripcion': 'Descripcion','score':
'score', 'id_producto': 'Frecuencia'}, inplace=True)

#pasando el data frame  a otro limitandolo solo a 20 registros
top_score = bestscore_product_sales[:20]

#Listado 20 productos con peores reseñas.
#agrupamos los score por descripción e id
worstscore_product_sales=df_product_sales.groupby(['Descripcion','score'])[['
id_producto']].count()

#ordenamos
worstscore_product_sales.sort_values(['score','id_producto'],
ascending=[True,False],inplace=True)

#cambiando el encabezado de la columna
worstscore_product_sales.rename(columns={'Descripcion':
'Descripcion','score': 'score', 'id_producto': 'Frecuencia'}, inplace=True)

#pasando el data frame  a otro limitandolo solo a 20 registros
bottom_score = worstscore_product_sales[:20]
```

Mostrando los reportes de cada consigna

Acontinuación el resto del codigo al seleccionar una opción de reporte. Para la consigna 3 dónde solicita ver los ingresos por mes, el mejor mes, promedios, etc. implemente las opciones de FOR e IF y recurri a las listas originales

```
if seleccion=="1":
          print("""
```

```
    """)
    print("Top 50 Ventas Productos    [1]")
   #muestra el reporte configurado fuera del bucle while
    print(top_50)

    print("""
    """)
    opcion=input("Ir a sección de reportes (s/n):  ")
    print("""
    """)
elif seleccion=="2":
    print("""
    """)
    print("Bottom 50 Ventas productos [2]")
    #muestra el reporte configurado fuera del bucle while
    print(bottom_50_cat_item)
    print("""
    """)
    opcion=input("Ir a sección de reportes (s/n):  ")
    print("""
    """)


elif seleccion=="3":
    print("""
    """)
    print("Los 100 MAS buscados       [3]")
    #muestra el reporte configurado fuera del bucle while
    print(top_100busquedas)
    print("""
    """)
    opcion=input("Ir a sección de reportes (s/n):  ")
    print("""
    """)


elif seleccion=="4":
    print("""
    """)
    print("Los 100 MENOS buscados     [4]")
    #muestra el reporte configurado fuera del bucle while
    print(bottom_100busquedas)
    print("""
```

```
        """)
        opcion=input("Ir a sección de reportes (s/n):  ")
        print("""
        """)


    elif seleccion=="5":
        print("""
        """)
        print("Las mejores reseñas =)     [5]")
        #muestra el reporte configurado fuera del bucle while
        print(top_score)
        print("""
        """)
        opcion=input("Ir a sección de reportes (s/n):  ")
        print("""
        """)


    elif seleccion=="6":
        print("""
        """)
        print("Las peores reseñas  =(     [6]")
        #muestra el reporte configurado fuera del bucle while
        print(bottom_score)
        print("""
        """)
        opcion=input("Ir a sección de reportes (s/n):  ")
        print("""
        """)


    elif seleccion=="7":
      #para el reporte quise hacerlo utilizando las listas originales y con
la libreria datetima y time para descomponer la fecha en mes y año
      import datetime
      import time
      #creo una lista vacia que me servida para almacenar los datos con las
fechas descompuestas
      ls_sales=[]
      for venta in lifestore_sales:
        for producto in lifestore_products:
          if venta[1]==producto[0]:
```

```
            fecha=venta[3]
            #doy formato a la fecha
            fecha2=datetime.datetime.strptime(fecha,'%d/%m/%Y')
            #separo mes y año y creo una columna donde solo me agrupe mes y
año
            mes=fecha2.month
            year=fecha2.year
            mes_year= fecha2.strftime('%Y.%m')
            #el siguiente if me servirá para determinar si existe una
devolución asignar como negativo el ingreso
            if venta[4]==1:
              precio=producto[2]*-1
            else:
              precio=producto[2]
          #agrego todos los registros a la lista

ls_sales.append([venta[0],venta[1],venta[2],fecha,mes,year,mes_year,year,vent
a[4],precio])
        #print(ls_sales)

        #ventas    columnas:id_venta,    id_producto,score,fecha,mes,    año,
devolucion, precio
        #Crearemos una lista de los periodos con registro unico año.mes
        lista_mes=[]
        lista_year=[]
        for periodo in ls_sales:
          lista_mes.append(periodo[6])
          lista_year.append(periodo[7])
        #las siguientes instrucciones me crean registros unicos de mes y años
para recorrerlas y sumar ingresos
        meses_unique=set(lista_mes)
        year_unique=set(lista_year)
        meses=sorted(meses_unique)
        years=sorted(year_unique)


        # Generamos la suma de ingresos por mes_año y el conteo de ventas
utilizando un recorrdido de las listas con resgistros unicos creada antes
        contador=0
        ingreso=0
        ingreso_mensual=[]
        for n in meses:
          for fechas in ls_sales:
```

```
            if fechas[6]==n:
              contador+=1
              ingreso+=fechas[9]
              avg=int(ingreso/contador)

          ingreso_mensual.append(['periodo:',n,'Total                ingreso:
',ingreso,'Promedio: ',avg])
          contador=0
          ingreso=0
        print("Ingresos al mes y Promedio [7]")
        print("El ingreso total y el ingreso promedio por mes es el siguiente:
")
        print("")
        for n in ingreso_mensual:
          print(n)
        print("""
                  """)
        #Para obtener el mes con mejores ingresos utilizamos excluimos los
valores de 2002 y 2019
        exclusion=['2002.05','2019.11']
        year2020=[]
        mes_max=[]
        for n in ingreso_mensual:
          if n[0] not in exclusion:
            year2020.append(n)
        year_ordenado = sorted(year2020, key=lambda x: x[3], reverse=True)
        #for n in year_ordenado:
         # print(n)

        print("el mes con mayores ingresos en 2020 fue: ",year_ordenado[0][1],
"con un ingreso de: ",year_ordenado[0][3])

        print("""
          """)

        opcion=input("Ir a sección de reportes (s/n):  ")
        print("""
          """)



      elif seleccion=="8":

        ls_sales=[]
```

```
        for venta in lifestore_sales:
          for producto in lifestore_products:
            if venta[1]==producto[0]:
              fecha=venta[3]
              fecha2=datetime.datetime.strptime(fecha,'%d/%m/%Y')
              mes=fecha2.month
              year=fecha2.year
              mes_year= fecha2.strftime('%Y.%m')
              if venta[4]==1:
                precio=producto[2]*-1
              else:
                precio=producto[2]

ls_sales.append([venta[0],venta[1],venta[2],fecha,mes,year,mes_year,year,vent
a[4],precio])
        #print(ls_sales)

        #ventas    columnas:id_venta,    id_producto,score,fecha,mes,    año,
devolucion, precio
        #Crearemos una lista de los periodos con registro unico año.mes
        lista_mes=[]
        lista_year=[]
        for periodo in ls_sales:
          lista_mes.append(periodo[6])
          lista_year.append(periodo[7])
        #print(lista_mes)
        meses_unique=set(lista_mes)
        year_unique=set(lista_year)
        meses=sorted(meses_unique)
        years=sorted(year_unique)

        #creamos la suma de ingresos por año y conteo
        contador2=0
        ingreso2=0
        ingreso_anual=[]
        for n in years:
          for fechas in ls_sales:
            if fechas[7]==n:
              contador2+=1
              ingreso2+=fechas[9]
              avg2=int(ingreso2/contador2)
          ingreso_anual.append(['Año:',n,'Ingreso  Total:  ',ingreso2,'Ingreso
Promedio:',avg2])
```

```
            contador2=0
            ingreso2=0


          print("Ingreso Anual y mejor mes  [8]")
          print("El ingreso total y el ingreso promedio por año es el siguiente:
")
          print("")
          for yyy in ingreso_anual:
            print(yyy)

          print("""
            """)
          opcion=input("Ir a sección de reportes (s/n):  ")
          print("""
            """)

        elif seleccion=="9":
            print("""
            """)
            opcion="n"
            login='true'
            print("""
                     Fin del programa
                                      """)
```

Solución del Problema

Categorías con menores ventas y categorías con menores búsquedas.

Las menores ventas registradas por producto-categoria son principalmente las siguientes

Bottom 50 Ventas productos [2]

Descripcion	categoria	Frecuencia
Cougar Audífonos Gamer Phontum Essential, Alámb...	audifonos	1
Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti...	tarjetas de video	1
Tarjeta de Video Gigabyte AMD Radeon R7 370 OC,...	tarjetas de video	1
Tarjeta de Video Asus NVIDIA GeForce GTX 1050 T...	tarjetas de video	1
Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2, ...	tarjetas madre	1
Tarjeta Madre Gigabyte XL-ATX TRX40 Designare, ...	tarjetas madre	1
Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151,...	tarjetas madre	1
TV Monitor LED 24TL520S-PU 24, HD, Widescreen, ...	pantallas	1
TCL Smart TV LED 55S425 54.6, 4K Ultra HD, Wide...	pantallas	1
SSD Crucial MX500, 1TB, SATA III, M.2	discos duros	1
Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 ...	tarjetas de video	1
HyperX Audífonos Gamer Cloud Flight para PC/PS4...	audifonos	1
Kit Memoria RAM Corsair Dominator Platinum DDR4...	memorias usb	1
Logitech Audífonos Gamer G332, Alámbrico, 2 Met...	audifonos	1
MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PCI...	tarjetas de video	1
Logitech Bocinas para Computadora con Subwoofer...	bocinas	2

Los productos con menos busquedas en la plataforma

Los 100 MENOS buscados [4]

Descripcion	Frecuencia
SSD Samsung 860 EVO, 1TB, SATA III, M.2	1
Ghia Bocina Portátil BX800, Bluetooth, Inalámbr...	1
Ginga Audífonos con Micrófono GI18ADJ01BT-RO, B...	1
Tarjeta Madre Gigabyte micro ATX Z390 M GAMING,...	1
Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151,...	1
Samsung Smart TV LED 43, Full HD, Widescreen, N...	1
Tarjeta de Video VisionTek AMD Radeon HD5450, 2...	1
MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PCI...	1
Procesador Intel Core i3-8100, S-1151, 3.60GHz,...	1
Acteck Bocina con Subwoofer AXF-290, Bluetooth,...	2
Genius GHP-400S Audífonos, Alámbrico, 1.5 Metro...	2
Tarjeta de Video Asus NVIDIA GeForce GTX 1050 T...	2
SSD para Servidor Lenovo Thinksystem S4500, 480...	2
Tarjeta de Video Gigabyte AMD Radeon R7 370 OC,...	3

Categorías con mayores ventas y categorías con mayores búsquedas.

El producto TOP de ventas es la memoria Kingston con una frecuencia de transacciones de 50 veces durante el periodo analizado.

Top 50 Ventas Productos [1]

Descripcion	Frecuencia
SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm	50
Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Si...	42
Procesador Intel Core i3-9100F, S-1151, 3.60GHz...	20
Tarjeta Madre ASRock Micro ATX B450M Steel Lege...	18
SSD Adata Ultimate SU800, 256GB, SATA III, 2.5'...	15
Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAM...	14
Procesador AMD Ryzen 3 3200G con Gráficos Radeo...	13
Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32...	13
SSD XPG SX8200 Pro, 256GB, PCI Express, M.2	11
Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 S...	9
SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0, M2	9

Los productos mas buscados, repite la memoria kingstone con 263 busquedas

Los 100 MAS buscados [3]

Descripcion	Frecuencia
SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm	263
SSD Adata Ultimate SU800, 256GB, SATA III, 2.5'...	107
Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAM...	60
Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Si...	55
Procesador AMD Ryzen 3 3200G con Gráficos Radeo...	41
Logitech Audífonos Gamer G635 7.1, Alámbrico, 1...	35
TV Monitor LED 24TL520S-PU 24, HD, Widescreen, ...	32
Procesador Intel Core i7-9700K, S-1151, 3.60GHz...	31

Sugerir una estrategia de productos a retirar del mercado así como sugerencia de cómo reducir la acumulación de inventario considerando los datos mensuales.

La estrategia es optimizar los inventarios retirando de la venta los productos que en los reportes de menor venta y menor busqueda presenten coincidencia (dentro los 50 registros mostrados en cada reporte). Otra forma de mejorar la rotación de productos para productos de bajo desplazamiento es armar promociones o kits de venta con productos TOP.

Considerar una estrategia de gestion de precio y promociones en productos top que no represente riesgo de caida de venta para lo que debe tenerse un reporte de sensibilida de precio en el top 10 de productos por ingreso

Existe información que debe ser analizada ya que presenta inconsistencias en las fechas , validar sí las fechas son correctas y de no ser así verificar en sistemas el error que originó la incosistencia.

Conclusiones

La ciencia de datos hoy dia requiere estar actualizado en herramientas de analisis como lo son Python, Rstudio, Power BI, etc. herramientas que mas que una moda son un facilitador con prestaciones bastante flexibles e interactivas para poder analizar volumenes muy importantes de datos.

En lo personal yo que vengo de herramientas que requieren menos programación como Tableau y PowerBI, he iniciado mi formación en paquetes de Rstudio y obviamente Python; siempre se me habia complicado el poder entender un lenguaje de programación pero para el caso de estudio de Python encontre un grado de facilidad mayor (vs rstudio) en el armado de código para resolver el caso.