

Fundamentos de programación con Python

Institución: Becas Santander y Emtech

Autor: Ángel Padilla Domínguez

Profesor: Jaime Saúl Alonso Sánchez

Grupo: 1, Pythonistas 🐍

15/09/2021



Índice

Introducción	3
Definición del código.	3
Solución al problema	10
Conclusión	10

Introducción

Este proyecto busca solucionar problemas de acumulación de inventario causado por las bajas ventas en algunos productos, a través del uso y manejo de los conceptos básicos de programación en Python. El objetivo es poner en práctica las bases de programación en Python para análisis y clasificación de datos mediante la creación de programas de entrada de usuario y validaciones, uso y definición de variables y listas, operadores lógicos y condicionales para la clasificación de información.

Definición del código.

El código que se muestra a continuación no contiene las listas utilizadas para realizar los análisis.

```
""" ----- ANALISIS DE DATOS ----- """

""" ----- Productos con mayores ventas ----- """

# Creamos una lista donde guardaremos la información [id_producto, ventas]
ventas_por_producto = []

# En este ciclo rellenamos la lista anterior con el id y un cero.
for producto in lifestore_products:
    lista_aux = [producto[0], 0]
    ventas_por_producto.append(lista_aux)

# No todas las ventas son validas ya que hay algunas devoluciones.
# En este ciclo se cuentan las ventas de cada producto
# y se agrega la información a la lista, siempre que no existan devoluciones.
for venta in lifestore_sales:
    producto_vendido = venta[1]
    devuelto = venta[4]
    # venta[1] nos da la información del id.
    # venta[4] tiene dos posibles valores 1 o 0, si es 1 el producto ha sido devuelto
    # por lo tanto la compra no es valida.

    # Si devuelto es 0 entonces la compra es valida.
    if devuelto == 0:
        # Entonces incrementamos la cantidad de articulos de ventas en 1
        # Le restamos 1 al valor de producto vendido ya que las listas comienzan en 0
        # y el id de producto en 1. Recordemos que esto solo funciona si la lista esta
ordenada
        # y solo se incrementa en 1 el numero de id para cada producto.
        # La suma se realiza en la posición [1] que comienza en 0 para todos los
productos.
        ventas_por_producto[producto_vendido - 1][1] += 1

# Ordenamiento de mayor a menor
# Para esto es necesario crear la siguiente función
# La cual entregara el valor de lista[1] para poder indicarle al metodo sort
```

```

# La columna que deseamos ordenar dentro de la lista ventas_por_producto
def segundo_elemento(lista):
    return lista[1]

# Refiriendonos a la documentación del método sort() encontramos que
# puede recibir dos parametros: reverse y key
# reverse es un booleano que indica si debe ordenarse en decendiente.
# key es una función que nos indica en base a qué ordenarlo.
ventas_por_producto.sort(reverse=True, key=segundo_elemento)

# Creamos la variable n con la cantidad de productos que deseamos guardar
n = 15
# Utilizando slicing obtenemos esta lista
# lista [a:b]
# a es elemento inicial que queremos
# b es el último elemento (no incluido)
# por lo que solo es necesario guardar los primeros elementos de la lista anteriormente
creada
# Creamos la lista con los artículos más vendidos
mas_vendidos = ventas_por_producto[:n]

""" ----- Los artículos con mayores búsquedas ----- """

# Creamos una lista donde guardaremos [id_product, búsquedas]
búsquedas_por_producto = []

# En este ciclo rellenamos la lista anterior con el id y un cero.
for producto in lifestore_products:
    lista_aux = [producto[0], 0]
    búsquedas_por_producto.append(lista_aux)

# En este ciclo incrementamos en 1 la cantidad de búsquedas
# cada vez que se encuentre un producto en la lista lifestore_searches.
for busqueda in lifestore_searches:
    producto_buscado = busqueda[1]
    búsquedas_por_producto[producto_buscado - 1][1] += 1

# Ordenamos la lista por la cantidad de búsquedas
búsquedas_por_producto.sort(reverse=True, key=segundo_elemento)

n = 20 # Cantidad de elementos que guardaremos en la lista de mas_buscados
mas_buscados = búsquedas_por_producto[:n]

""" ----- Las mejores y peores reseñas ----- """

# Creamos una lista auxiliar donde se guardara [id_product, suma de cada reseñas,
cantidad de reseñas]
resenas_por_producto_aux = []

# Rellenamos la lista con el id y ceros.
for producto in lifestore_products:
    lista_aux = [producto[0], 0, 0]
    resenas_por_producto_aux.append(lista_aux)

# Contamos cuantas veces se ha realizado una reseña
for resenas in lifestore_sales:
    resena_producto = resenas[1]
    resenas_por_producto_aux[resena_producto - 1][2] += 1

```

```

# Sumamos las calificaciones de las reseñas
for valoracion in lifestore_sales:
    resena_producto = valoracion[1]
    aux = valoracion[2] # Guardamos el valor de la reseña y
    # Los vamos sumando para cada producto
    resenas_por_producto_aux[resena_producto - 1][1] += aux

# Creamos la lista donde guardaremos [id_product, valor promedio de las reseñas]
resenas_por_producto = []

for producto in resenas_por_producto_aux:
    suma_de_resenas = producto[1]
    cantidad_de_resenas = producto[2]

    # Si el producto no tiene reseñas el valor sera 0
    if cantidad_de_resenas == 0:
        lista_aux = [producto[0], 0]

    # Si no: se realiza el calculo del promedio.
    else:
        promedio = suma_de_resenas / cantidad_de_resenas
        lista_aux = [producto[0], promedio]

    # Se agrgan los datos a la lista correspondiente
    resenas_por_producto.append(lista_aux)

# Ordenamiento de mayor a menor
resenas_por_producto.sort(reverse=True, key=segundo_elemento)

# 10 con mejores reseñas
n = 10 # Seleccionamos la cantidad de articulos que queremos en la lista

# Guardamos los datos en la lista mejores_resenas
mejores_resenas = resenas_por_producto[:n]

# 10 con peores reseñas
m = 10 # Cantidad de articulos que queremos en la lista

# Creamos la lista vacia donde agregaremos los datos de menor a mayor
peores_resenas = []

# En este ciclo se realiza un conteo de 1 a m+1 (11)
for i in range(1, m+1):
    # Utilizamos indices negativos para ir agregando los datos empezando por
    # el ultimo articulo de la lista resenas_por_producto.
    peores_resenas.append(resenas_por_producto[-i])

""" ----- Total de ingresos mensuales ----- """

# Creamos una lista vacia donde se guardara [mes en numero, ingresos]
ingresos_por_mes = []

# Recorremos los numeros del 1 al 12 para guardarlos de manera ordenada
for i in range(1,13):
    mes_aux = [i, 0]
    ingresos_por_mes.append(mes_aux)

# En este ciclo obtenemos que producto se vendió, en que mes,
# si fue devuelto y cual es su precio.
for venta in lifestore_sales:

```

```

mes = int(venta[3][3:5]) # Convertimos el mes a un numero entero
producto = venta[1] # Obtenemos el id para poder buscar su precio
precio = lifestore_products[producto - 1][2] # Obtenemos el precio del producto
devuelto = venta[4] # Devuelto es = 0 si el producto no fue devuelto

# Si La compra es valida
if devuelto == 0:
    # Sumamos el precio del producto en el mes que corresponda a su venta
    ingresos_por_mes[mes - 1][1] += precio

""" ----- Ventas promedio mensuales ----- """

# Creamos La variable ventas donde guardaremos La cantidad de ventas realizadas
ventas = 0

# Contamos cuantas ventas validas se realizaron
for venta in ventas_por_producto:
    aux = venta[1]
    ventas += aux # Y las vamos sumando en la variable

# Sacamos el promedio de las ventas por mes
ventas_promedio_mensuales = ventas / 12

""" ----- Total de ingresos anual ----- """

# Creamos La variable donde guardaremos La suma de los ingresos
total_de_ingresos = 0

for venta in ventas_por_producto:
    producto = venta[0] # Obtenemos el id para despues obtener el precio
    ventas = venta[1] # Sacamos La cantidad de ventas
    precio = lifestore_products[producto - 1][2]
    # Multiplicamos la cantidad de ventas por el precio del producto
    total_de_ingresos += (ventas * precio) # y lo sumamos a la variable

""" ----- Meses con más ventas al año ----- """

# Creamos una lista con [mes, 0]
ventas_por_mes = []
for i in range(1,13):
    mes_aux = [i, 0]
    ventas_por_mes.append(mes_aux)

# Sumamos en 1 ventas_por_mes[mes][1]
# por cada venta segun el mes en el que se realizo
for venta in lifestore_sales:
    mes = int(venta[3][3:5])
    ventas_por_mes[mes - 1][1] += 1

# Ordenamos La lista de mayor a menor por ventas
ventas_por_mes.sort(reverse=True, key=segundo_elemento)

""" ----- Productos por categoria ----- """

# Productos por categoria
# lifestore_searches = [id_search, id product]
# lifestore_sales = [id_sale, id_product, score (from 1 to 5), date, refund (1 for true
or 0 to false)]
# lifestore_products = [id_product, name, price, category, stock]

```

[illegible]

```

""" ----- Inicio de sesion ----- """

print(dialogo_inicio)

# El login es bastante sencillo se establece un usuario y contraseña dentro del codigo
while True:
    usuario = input("Ingrese su usuario: ")
    contrasena = input("Ingrese su contraseña: ")

    if usuario == "AngelP" and contrasena == "admin":
        # Si la contraseña y usuario coinciden se rompe el ciclo
        menu == True
        print(limpiar_pantalla)
        print(separador)
        print(menu_inicio)
        break

    else:
        print(limpiar_pantalla)
        print(dialogo_inicio)
        print("Usuario y/o contraseña incorrectos")
        print('Si desea salir presione 1: ')
        a = input()
        if a == '1':
            print('Hasta luego!')
            menu = False
            break
        else:
            print(limpiar_pantalla)
            print(dialogo_inicio)
            print('Reintentar inicio de sesion \n')

# si el usuario no decidio salir se entra en este nuevo ciclo
while menu == True:

    opcion = input()

    if opcion == '1':
        titulo = ['Productos con mayores ventas\n']
        info = mas_vendidos
        print(limpiar_pantalla)
        print(separador)
        print(f'{titulo}')
        print('[Id del producto, Ventas]')
        for i in info:
            print(f'{i}')
        print(menu_aux)

    elif opcion == '2':
        titulo = ['Productos con mayores búsquedas\n']
        info = mas_buscados
        print(limpiar_pantalla)
        print(separador)
        print(f'{titulo}')

        print('[Id del producto, Búsquedas]')
        for i in info:
            print(f'{i}')
        print(menu_aux)

```



```

elif opcion == '3':
    titulo = ['Productos con mejores reseñas\n']
    info = mejores_resenas
    print(limpiar_pantalla)
    print(separador)
    print(f'{titulo}')
    print('[Id del producto, Promedio de las reseñas]')
    for i in info:
        print(f'{i}')
    print(menu_aux)

elif opcion == '4':
    titulo = ['Productos con peores reseñas\n']
    info = peores_resenas
    print(limpiar_pantalla)
    print(separador)
    print(f'{titulo}')
    print('[Id del producto, Promedio de las reseñas]')
    for i in info:
        print(f'{i}')
    print(menu_aux)

elif opcion == '5':
    titulo = ['Total de ingresos mensuales\n']
    info = ingresos_por_mes
    print(limpiar_pantalla)
    print(separador)
    print(f'{titulo}')
    print('[Mes, Ingresos]')
    for i in info:
        print(f'{i}')
    print(menu_aux)

elif opcion == '6':
    titulo = ['Ventas promedio mensuales\n']
    info = ventas_promedio_mensuales
    print(limpiar_pantalla)
    print(separador)
    print(f'{titulo}')
    print('Promedio mensual: ')
    print(f'{info}')
    print(menu_aux)

elif opcion == '7':
    titulo = ['Total de ingresos anuales\n']
    print(limpiar_pantalla)
    print(separador)
    print(f'{titulo}')
    print('Total de ingresos anuales: $')
    print(total_de_ingresos)
    print(menu_aux)

elif opcion == '8':
    titulo = ['Meses con más ventas\n']
    info = ventas_por_mes
    print(limpiar_pantalla)
    print(separador)
    print(f'{titulo}')
    print('[Mes, Ventas]')
    for i in info:

```

```
        print(f'{i}')
        print(menu_aux)

    elif opcion == '0':
        print(limpiar_pantalla)
        print(separador)
        print(menu_inicio)

    else:
        print('Opcion incorrecta intente de nuevo')
        print(menu_aux)
```

Solución al problema

La solución propuesta seria retirar ya sea en su totalidad o un porcentaje de su stock, de aquellos artículos que no han tenido ventas en el año, dejando espacio para incrementar aquellos que han tenido ventas regulares y además tienen buenas reseñas.

Podríamos realizar un análisis entre las dos listas, productos más vendidos y mejores reseñas para obtener aquellos artículos que deseamos conservar y aumentar su stock, y de manera inversa con las listas de menores ventas y peores reseñas, podemos seleccionar los productos que se desean retirar, utilizando una serie de descuentos en los meses con menos ventas, para así deshacernos de los productos rezagados y además nivelar las ventas a lo largo del año, aumentando las ganancias anuales.

Conclusión

Fue complicado unir el rompecabezas de como manejar las listas para obtener datos específicos pero me siento satisfecho con el resultado, ahora es mucho más sencillo enfrentarme a situaciones similares en las que sea necesario ordenar y separar una serie de datos para realizar análisis.

Anexo

Link del código: <https://github.com/AngelPadD/PROYECTO-01--ANGEL-PADILLA>