



JORGE RODRIGUE RUEDA

PRIMER REPORTE INTRODUCCIÓN A PYTHON

Índice

Contents

| | |
|---|----|
| Introducción | 3 |
| Explicación código | 4 |
| #1 Mayores ventas | 4 |
| #2 Mayores búsquedas | 5 |
| # 3 Menores Ventas por categoría | 6 |
| # 4 Menores búsquedas por categoría | 7 |
| # 5 Productos por reseña en el servicio | 8 |
| Solución al problema | 10 |
| Conclusión | 11 |

Introducción

A continuación, se muestra una posible solución a la problemática que presenta LifeStore, una tienda virtual que tiene un problema con su colección de inventario y esto ha generado problemas de ventas en su ultimo semestre. Usando herramientas tecnológicas como lo es Python, podemos crear una posible solución a sus problemas con los datos, categorizando y agrupando de diferentes modos para establecer un reporte y una posible solución a sus problemas.

Explicación código

#1 Mayores ventas

El código no funciona, pero la idea detrás de ello era crear un contador donde mientras `i` sea menor o igual dentro de la longitud de `lifestore_sales` produzca lo siguiente:

`lista_ventas = [0,0]` Crear un nuevo dataset

`i = 0 u = 1 x = 0 c = 0` Variables que funcionan como contadores

siento esto entrar en una función while donde hará lo siguiente

`while i <= len(lifestore_sales):`

`if lifestore_products[i][1] == u:` revisar que la posición `[i]` sea igual a `u` para que haga lo siguiente

`continue` en caso de que no encuentre esa condición se pide que continúe

`x = 1 + x` Hacer el contador de las veces que encuentra `u` dentro de la lista de `lifestore_products`

`lista_ventas[i] = x` Asigna `x` en la nueva lista de ventas

`lista_ventas[c][1] = lifestore_sales [c][0]` Asigna el producto que se está contabilizando a un renglón de la lista ventas

`else:`

`x = 0` Cuando se acabara de cumplir la función el contador regresa a 0

`u = u + 1` Ahora lo que va a buscar sería el producto 2

`c = c + 1` Ahora se va a asignar dentro de otro espacio dentro de `lista_ventas`. Para no super poner

`i= i+1` Para que siga la condición while

```
lista_ventas.sort()) ordenamos nuestros valores
```

```
While y <= 50:
```

```
    print(lista_ventas[y])
```

```
    y = 1+y
```

Para visualizar de manera ordenada los 50 productos que se está pidiendo use un while.

#2 Mayores búsquedas

Es un código similar al anterior, solo que cambiando las variables y la posición de los datos

```
lista_busquedas = [0,0]
```

```
i = 0 u = 1 x = 0 c = 0
```

 Variables que funcionan como contadores

siento esto entrar en una función while donde hará lo siguiente

```
while i <= len(lifestore_searches):
```

```
    if lifestore_products[i][1] == u:
```

 tomará la posición 0 dentro de la columna 1 y verificará que sea un 1 y hará lo siguiente

```
        continue
```

```
        x = 1 + x
```

```
        lista_busquedas[i] = x
```

```
        lista_busquedas[i][1] = lifestore_products [i][1]
```

```
    else:
```

```
        x = 0
```

```
        u = u + 1
```

```
        i = i+1
```

```
        c = c + 1
```

```
lista_busquedas.sort())
```

```
While y <= 100:
```

```
    print(lista_busquedas[y])
```

```
    y = 1+y
```

Para visualizar de manera ordenada los 100 productos que se está pidiendo use un while.

3 Menores Ventas por categoría

```
lista_ventas = [0,0]
i = 0
u = 1
x = 0
c = 0
while i <= len(lifestore_sales):
    if lifestore_products[i][1] == u:
        continue
        x = 1 + x
        lista_ventas[c] = x
        lista_ventas[c][1] = lifestore_sales [i][0]
    else:
        x = 0
        u = u + 1
        c = c + 1
        i = i +1
print(lista_ventas.sort())
```

básicamente lo mismo del anterior, pero con la siguiente parte del código

```
while i <= len(lifestore_products):
    if lifestore_products[i][0] == lista_ventas[i][0]:
        lista_ventas[i] = lifestore_products[i][3] + lista_ventas [i]
```

Lo que traté de hacer con ese código fue después de contar cuantas ventas hubo de cada producto juntar ahora de que "categoría" es el objeto, para después usar:

```
lista_ventas.sort(reverse=TRUE, key = lista_ventas[2])
```

```
While y <= 100:
```

```
    print(lista_ventas[y])
```

```
    y = 1+y
```

Para que lo acomode de menor a menor por categoría que en este caso se representaría con `lista_ventas[2]`

4 Menores búsquedas por categoría

```
lista_busquedas = [0,0]
```

```
i = 1
```

```
u = 1
```

```
x = 0
```

```
while i in len(lifestore_searches):
```

```
    if lifestore_products[i][1] == u:
```

```
        continue
```

```
        x = 1 + x
```

```
        lista_busquedas[i] = x
```

```
        lista_busquedas[i][1] = lifestore_products [i][1]
```

```
    else:
```

```
        x = 0
```

```
        u = u + 1
```

```
        i = i+1
```

```
        c = c + 1
```

```
print(lifestore_searches[0][1])
```

```
while i <= len(lifestore_products):
```

```
if lifestore_products[i][0] == lista_busquedas[i][0]:  
    lista_ventas[i] = lifestore_products[i][3] + lista_busquedas[i]  
lista_busquedas.sort(reverse=TRUE, key = lista_busqueda[2])
```

Básicamente lo mismo que el anterior.

5 Productos por reseña en el servicio

```
lista_1=[] Crear una nueva lista vacia
```

```
i=0
```

```
u=1
```

```
x=0
```

```
c =0
```

```
while i <= len(lifestore_sales):
```

```
    if lifestore_products[i][1] == u:
```

```
        continue
```

```
        x = lifestore_products[i][3] + x Obtener la reseña, e irla sumando
```

```
        c= c + 1 Contar cuantas veces existe el producto en la lista
```

```
        prom = x/c Sacar el promedio
```

```
        lista_1[c][1] = lifestore_sales[i][1] Asignar en la lista el prodcuto
```

```
        lista_1[c][0] = prom Asignar el promedio del producto
```

```
    else:
```

```
        x=0
```

```
        c=0
```

```
        u = u+1
```

```
        i=i+1
```


Imprimir los 20 valores que se piden, primero se ordena para obtener los mejores

```
while i<=20:
```

```
    lista_1.sort()
```

```
    print(str(lista_1[i]))
```

```
    i=i+1
```

obtener los 20 valores que piden, pero esta vez queremos los peores por eso usamos reverse = TRUE para invertir la lista

```
while i<=20:
```

```
    lista_1.sort(reverse=TRUE)
```

```
    print(str(lista_1[i]))
```

```
    i=i+1
```

Solución al problema

Lo que se buscó crear era un programa que permitiera a la tienda, una oportunidad de manejar sus datos, ya que estaban muy mal organizados de la tienda.

Lo que generamos con esto es obtener insights que le muestren lo siguiente:

- Donde están los productos que no se están vendiendo y dejar de comprarlos, o en su defecto liquidarlos para evitar el costo por almacén.
- Cuáles son los mejores productos con las mejores reseñas lo que indicaría cuales son los productos estrella que la empresa debe seguir adquiriendo.
- Cuales productos están siendo devueltos e indicar problemas técnicos o de proveedor
- Cuál es la categoría que vende mejor y que productos son los estrella.

Conclusión

Podemos concluir en como las herramientas tecnológicas como lo es Python, pueden ser de gran ayuda en la implementación de nuevos sistemas y soluciones para las empresas, este fue un ligero ejemplo de lo que puede ser