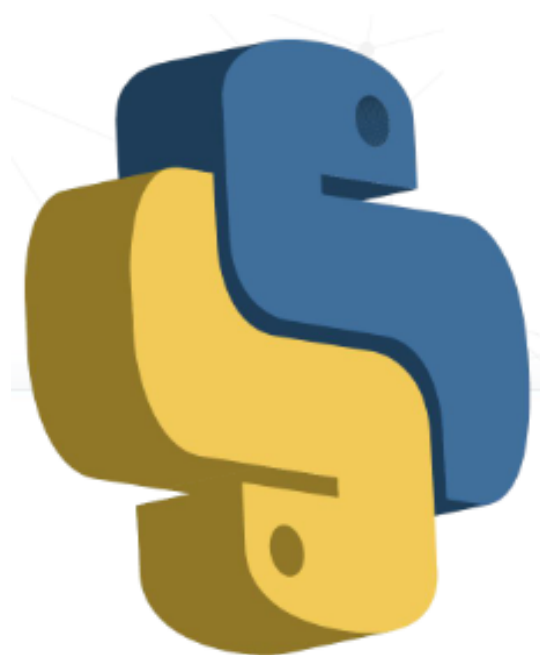



Introducción a Python

Proyecto 1



Leonardo Garcia Gutierrez

14 de septiembre del 2021



Introducción	1
Definición del Código	2
Login:	2
Mejores ventas	2
Busca categorías	3
Productos Vendidos Por Categoría	3
100 Productos con Mayores Búsquedas	5
Productos con Menores Búsquedas Por Categoría	5
Mejores y Peores Reseñas de Productos	7
Ventas e Ingresos por Mes	8
Ventas e Ingreso Anual	8
Mayores Ventas por Mes	9
Bloques de Código para mostrar Resultados	9
Solución del Problema	9
Conclusión	9



Introducción

En este proyecto se pusieron en práctica las bases de programación en Python vistas en las semanas pasadas, en el proyecto consiste en analizar y clasificar datos, usando validaciones, variables, listas, operadores lógicos y condicionales.

El caso donde lo estamos aplicando en una tienda virtual que ha tenido problema con acumulacion de inventario, también se percataron que se han reducido las búsquedas en un grupo importante de sus productos.

También se sugiere una estrategia para retirar ciertos productos del mercado.

Definición del Código

Login:

El login te permite intentar ingresar 3 veces antes de que de por terminado el programa con ayuda de un while que hace que funcione mientras los intentos no sean más de 3, si se ingresa correctamente el programa entra a un if que es donde está la mayor parte del programa sino entra en el else y a intentos se le suma uno, el usuario y la contraseña están preestablecidas simulando ser el administrador, si se ingresa el usuario o contraseña incorrectos.

Mejores ventas

Creamos una lista donde guardaremos las ventas, con idea de que después podamos ordenarla y guardarla en otra lista, `max_product_id_number` es una variable que guarda el número más alto del `product_id` que se tiene actualmente en los datos, filtramos las ventas por producto los guardamos en listas con los productos, con mayores ventas en la lista `best_sales` que la ordenamos con ayuda del método `sort` pero en reversa y los productos con menos ventas en la lista `worst_sales`, de igual manera se pueden ordenar con el método `sort` y se tiene un formato en las listas de `[number_of_sales, id_product]`.

```

for product in range(max_product_id_number + 1):
    counter = 0
    for sale in range(283):
        if (product == lifestore_sales[sale][1]):
            counter += 1
        sales.append([counter, product])

sales.sort()
worst_sales = [i for i in sales]
sales.sort(reverse = True)
best_sales = [i for i in sales]
# sales = [number_of_sales, id_product]

```

Busca categorías

Quise buscar categorías que este bloque de código solo para comprobar que no hubiera categorías muy parecidas y que esto pudiera afectar el análisis después.

```

# Un bloque de código que busca que categorías tenemos y las
# guarda en una lista llamada list_of_categories
list_of_categories = []
category = ""
for product in range(len(lifestore_products)):
    if lifestore_products[product][3] != category:
        list_of_categories.append(lifestore_products[product][3])
        category = lifestore_products[product][3]

```

Productos Vendidos Por Categoría

```
# Creamos listas para filtrar los productos vendidos por categoria
# formato [number_of_sales, id_product]
worst_sales_processors = []
worst_sales_videocards = []
worst_sales_motherboards = []
worst_sales_drives = []
worst_sales_usbs = []
worst_sales_monitors = []
worst_sales_speakers = []
worst_sales_headphones = []
```

```
"""
En este bloque de código clasificamos las peores ventas por categoría con
ayuda de condicionales, aunque pudiera parecer algo complejo solo estamos
clasificando por categoría y agregando a una lista individual por categoría.
"""
for i in range(len(worst_sales)):
    if lifestore_products[worst_sales[i][1]][3] == 'procesadores':
        worst_sales_processors.append([worst_sales[i][0], worst_sales[i][1]])

    elif lifestore_products[worst_sales[i][1]][3] == 'tarjetas de video':
        worst_sales_videocards.append([worst_sales[i][0], worst_sales[i][1]])

    elif lifestore_products[worst_sales[i][1]][3] == 'tarjetas madre':
        worst_sales_motherboards.append([worst_sales[i][0], worst_sales[i][1]])

    elif lifestore_products[worst_sales[i][1]][3] == 'discos duros':
        worst_sales_drives.append([worst_sales[i][0], worst_sales[i][1]])

    elif lifestore_products[worst_sales[i][1]][3] == 'memorias usb':
        worst_sales_usbs.append([worst_sales[i][0], worst_sales[i][1]])

    elif lifestore_products[worst_sales[i][1]][3] == 'pantallas':
        worst_sales_monitors.append([worst_sales[i][0], worst_sales[i][1]])

    elif lifestore_products[worst_sales[i][1]][3] == 'bocinas':
        worst_sales_speakers.append([worst_sales[i][0], worst_sales[i][1]])

    elif lifestore_products[worst_sales[i][1]][3] == 'audifonos':
        worst_sales_headphones.append([worst_sales[i][0], worst_sales[i][1]])
```

100 Productos con Mayores Búsquedas

```
"""Productos con menores y mayores busquedas
[numero de veces, id_product]"""
searches_list = []
"""Creamos una lista donde se agregara cuantas veces aparece cada producto,
esto lo hacemos con ayuda del siguiente bloque de codigo, contamos con ayuda
de una variable contadora cada que aparece el id del producto que esta
iterando el ciclo for y agregamos a la lista search_list el formato
[numero de veces, id_product]"""
for product in range(len(lifestore_products) + 1):
    counter = 0
    for search in range(len(lifestore_searches)):
        if product == lifestore_searches[search][1]:
            counter += 1
    if counter != 0:
        searches_list.append([counter, product])
# Ordenamos los productos de la lista searches_list
searches_list.sort()

# [numero de veces, id_product]
most_searches_list = [] # Lista con los productos con mayores busquedas
less_searches_list = [] # Lista con los productos con menores busquedas

# Tomamos los "100" productos mas vendidos de searches_list
for i in range(len(searches_list)):
    less_searches_list.append(searches_list[i])

# Ordenamos de mayor a menor a partir de la lista search_list
for i in range(-1, -1 * len(searches_list) , -1):
    most_searches_list.append(searches_list[i])
```

Productos con Menores Búsquedas Por Categoría

```
"""filtramos por categoria los productos menos buscados, tomamos como base a la lista
less_searches_list y clasificamos por categorias con ayuda de condicionales"""

#Creamos listas donde agregaremos los productos y cuantas veces aparecieron
searched_processors = []
searched_videocards = []
searched_motherboards = []
searched_drives = []
searched_usbs = []
searched_monitors = []
searched_speakers = []
searched_headphones = []
```

```
for product in range(len(less_searches_list)):

    if(lifestore_products[less_searches_list[product][1] - 1][3] == 'procesadores'):
        searched_processors.append([less_searches_list[product][0],less_searches_list[product][1]])

    if(lifestore_products[less_searches_list[product][1] - 1][3] == 'tarjetas de video'):
        searched_videocards.append([less_searches_list[product][0],less_searches_list[product][1]])

    if(lifestore_products[less_searches_list[product][1] - 1][3] == 'tarjetas madre'):
        searched_motherboards.append([less_searches_list[product][0],less_searches_list[product][1]])

    if(lifestore_products[less_searches_list[product][1] - 1][3] == 'discos duros'):
        searched_drives.append([less_searches_list[product][0],less_searches_list[product][1]])

    if(lifestore_products[less_searches_list[product][1] - 1][3] == 'memorias usb'):
        searched_usbs.append([less_searches_list[product][0],less_searches_list[product][1]])

    if(lifestore_products[less_searches_list[product][1] - 1][3] == 'pantallas'):
        searched_monitors.append([less_searches_list[product][0],less_searches_list[product][1]])

    if(lifestore_products[less_searches_list[product][1] - 1][3] == 'bocinas'):
        searched_speakers.append([less_searches_list[product][0],less_searches_list[product][1]])

    if(lifestore_products[less_searches_list[product][1] - 1][3] == 'audifonos'):
        searched_headphones.append([less_searches_list[product][0],less_searches_list[product][1]])
```

Mejores y Peores Reseñas de Productos

"""Creamos lista de las reseñas de cada producto, en los bloques de código siguientes, buscamos en las ventas los productos y promediamos sus reseñas, después las ordenamos de mayor a menor y de menor a mayor"""

```
average_review_list = [] # [promedio_resena, producto]
for product in range(len(lifestore_products) + 1):
    counter = 0
    review_sum = 0
    for sale in range(len(lifestore_sales)):
        if (product == lifestore_sales[sale][1] and lifestore_sales[sale][4] == 0):
            counter += 1
            review_sum += lifestore_sales[sale][2]
    if counter != 0:
        average_review = review_sum / counter
        average_review_list.append([average_review, product])
average_review_list.sort(reverse = True)
```

```
# Creamos 1 lista con los 20 productos con mejores reseñas
best_reviews = [] # [promedio_resena, producto]
for resena in range(20):
    best_reviews.append(average_review_list[resena])
```

```
# Creamos 1 lista con los 20 productos con peores reseñas
worst_reviews = [] # [promedio_resena, producto]
for i in range(-1, -21, -1):
    worst_reviews.append(average_review_list[i])
```


Ventas e Ingresos por Mes

```
"""
Estamos tomando como ventas, todo lo que se vendio
Por otra parte en ingreso estamos tomando lo que se vendio menos devoluciones
Primero creamos las listas de ventas e ingresos
En la lista de ventas buscamos cada venta y la clasificamos con ayuda de condicionales
acumulamos el total de ventas e ingresos acumulandolas directamente en las listas
que creamos.
"""

sales_per_month = [['Enero', 0], ['Febrero', 0], ['Marzo', 0], ['Abril', 0], ['Mayo', 0],
income_per_month = [['Enero', 0], ['Febrero', 0], ['Marzo', 0], ['Abril', 0], ['Mayo', 0]

for sale in range(len(lifestore_sales)):
    fecha = lifestore_sales[sale][3]
    mes = fecha[3:5]
    anio = fecha[-4:]
    if(mes == '01' and anio == '2020'):
        if(lifestore_sales[sale][4] == 0):
            income_per_month[0][1] += lifestore_products[lifestore_sales[sale][1]-1][2]
            sales_per_month[0][1] += lifestore_products[lifestore_sales[sale][1]-1][2]

    if(mes == '02' and anio == '2020'):
        if(lifestore_sales[sale][4] == 0):
            income_per_month[1][1] += lifestore_products[lifestore_sales[sale][1]-1][2]
            sales_per_month[1][1] += lifestore_products[lifestore_sales[sale][1]-1][2]
```

En la última imagen solo aparecen hasta el mes “02”, pero sería el mismo proceso hasta el mes 12.

Ventas e Ingreso Anual

```
# Con los ingresos y ventas mensuales procedemos a sacar lo acumulado en el año
anual_sales = 0
# Funciona como acumulador y es donde guardamos el resultado de las ventas anual
anual_income = 0
for month in range(len(sales_per_month)):
    anual_income += income_per_month[month][1]
    anual_sales += sales_per_month[month][1]
```

Mayores Ventas por Mes

```
# Creamos una lista nueva lista para las ventas por mes, esta estara ordenada
# empezando con los meses con mas ventas
ordered_list_sales_per_month = [[0, 'Enero'], [0, 'Febrero'], [0, 'Marzo'],[0,
for month in range(len(sales_per_month)):
    ordered_list_sales_per_month[month][0] = sales_per_month[month][1]
ordered_list_sales_per_month.sort(reverse = True)
```

Bloques de Código para mostrar Resultados

Todos son muy similares, para esto básicamente solo estaremos imprimiendo las listas que previamente obtuvimos.


```
# Bloque para mostrar los resultados de los productos con mejores ventas
print(' ')
print('Lugar en // id_product // Numero de ')
print(' ventas // // ventas')
for i in range(50):
    print((f' {i+1} // {best_sales[i][1]} // {best_sales[i][0]}'))
```

Solución del Problema

La recomendación que yo daría es enfocarse en los productos que más se venden, limitar la variedad por categoría, ya que hay productos que a lo largo del año no se han vendido ni una unidad. Es preocupante como han bajado los ingresos de enero a lo que va del año, quizá también podría plantearse sería considerar el cierre del negocio. Aprovechar las temporadas de ventas para poder recuperar capital, disminuir el stock que no se vende mucho y comprar el que sí. Hacer un análisis más profundo sobre las reseñas y las devoluciones, para aclarar si son casos específicos o algo del proveedor.

Conclusión

Fue un proyecto que me enseñó mucho, me gusto mucho programar, de momentos me pareció trabajoso, me pregunto qué más se podrá hacer con herramientas más poderosas si esto solo lo hicimos con python y con el conocimiento que nos brindó el



curso. Me gustaria saber como se trabaja en una empresa en cuanto a la integración de de los diferentes departamentos, que soluciones hubieran dado ellos.