

EMTECH

Emerging Technologies Institute

EMTECH

Proyecto final 1

INTRODUCCIÓN A PYTHON

Grupo: DataSapiens

Andrade Zuñiga Maria Guadalupe

Domingo 5 de diciembre de 2021

ÍNDICE

Introducción	3
Definición del código	4
Resolución al problema	9
Conclusión	10

INTRODUCCIÓN

En este trabajo realizaremos listas donde se encuentren los productos más y menos vendidos, más y menos buscados, con mejores y peores reseñas, al igual, sabremos cuáles fueron las ganancias totales y las ventas promedio mensuales, total de ventas anuales y meses con más ventas al año.

Con esto, sabremos qué productos son los más rezagados de la empresa, los más vendidos, y cómo se puede intervenir para que esto cambie y la empresa tenga más productos con mejores ventas y reseñas en el mercado.

Todo esto se logrará utilizando el programa PYTHON en el cuál haremos un código que contendrá listas de variables en las que se encontraran los resultados a lo antes planteado.

DEFINICIÓN DEL CÓDIGO

Comenzamos el código ingresando un LOGIN, el cual dará acceso a las listas que se hicieron para lograr el objetivo. Si no se ingresa el ID (usuario) correcto, se le pedirá que vuelva a escribirlo, dándole un número de intentos para lograr ingresar.

“usuario: data amigo”

```
1429
1430 usuario = "data amigo" # Este es el usuario correcto
1431
1432 entrada = input("ID ")
1433 usuario_correcto = False
1434
1435 intentos = 3
1436
1437 while usuario_correcto == False:
1438     if entrada == usuario:
1439         print("Bienvenido jefe DataSapiens")
1440         usuario_correcto = True
1441     else:
1442         if intentos == 0:
1443             print("3312, tenemos un 3312")
1444             break
1445         entrada = input("Lo sentimos. Este código se destruirá despues de " + str(intentos) + " intentos. Por f
1446         usuario_correcto = False
1447
1448         intentos = intentos - 1
1449
1450
```

→ La primera lista que se encuentra es la de “ventas” en la cual se reflejan los 5 productos con mayor número de ventas ordenados de mayor a menor.

Para lograrlo, se pidió primero una lista que estuviera ordenada de mayor a menor con todos los productos existentes reflejando el número de ventas totales de cada producto, la cual no se ve reflejada en la terminal, posteriormente, se creó un acceso exclusivo en el cual solo se ven reflejados los 5 productos con mayores ventas totales, de igual manera ordenados de mayor a menor y siendo los únicos reflejados en la terminal.

```
1451
1452 # Creamos tabla para contar ventas de los productos #
1453 lista_ventas = []
1454 for seq in range(1, len(lifystore_products) + 1):
1455     lista_ventas.append([seq, 0])
1456
1457
1458 cantidad_registros_ventas = len(lifystore_sales)
1459
1460 for seq in range(0, cantidad_registros_ventas):
1461     producto = lifystore_sales[seq][1]
1462     lista_ventas[producto - 1][1] = lista_ventas[producto - 1][1] + 1
1463
1464
1465 from operator import itemgetter
1466 lista_ordenada = sorted(lista_ventas, key = itemgetter(1), reverse = True)
1467
1468 print("\n Los productos con más ventas son: \n")
1469
1470
1471 for seq in range(0, 5):
1472     print(lista_ordenada[seq])
1473     print(lifystore_products[lista_ordenada[seq][0] - 1][1])
1474
```

- La segunda lista es la de “**búsquedas**” en esta se reflejan los 10 productos con más búsquedas en el sistema, ordenados de mayor a menor.

Para lograrlo, se pidió primero una lista que estuviera ordenada de mayor a menor con las búsquedas totales de todos los productos existentes, la cual no se ve reflejada en la terminal, posteriormente, se creó un acceso exclusivo en el cual solo se ven reflejados los 10 productos con mayor número de búsquedas totales, los cuales de igual manera se encuentran ordenados de mayor a menor y siendo los únicos reflejados en la terminal.

```
1475 # Creamos tabla para contar búsquedas de los productos #
1476 lista_búsquedas = []
1477 for seq in range(1, len(lifestore_products) + 1):
1478     lista_búsquedas.append([seq, 0])
1479
1480
1481 cantidad_registros_búsquedas = len(lifestore_searches)
1482
1483 for seq in range(0, cantidad_registros_búsquedas):
1484     producto1 = lifestore_searches[seq][1]
1485     lista_búsquedas[producto1 - 1][1] = lista_búsquedas[producto1 - 1][1] + 1
1486
1487
1488 from operator import itemgetter
1489 lista_ordenada1 = sorted(lista_búsquedas, key = itemgetter(1), reverse = True)
1490
1491 print("\n Los productos más buscados son: \n")
1492
1493
1494 for seq in range(0, 10):
1495     print(lista_ordenada1[seq])
1496     print(lifestore_products[lista_ordenada1[seq][0] - 1][1])
1497
1498 print("\n")
1499
```

- La tercera lista es la de “**ventas menores**” en la cual se encuentran reflejados los 5 productos con menor número de ventas totales ordenados de menor a mayor.

Para lograrlo, se pidió primero una lista que estuviera ordenada de menor a mayor con todos los productos existentes reflejando el número de ventas totales de cada producto, la cual no se ve reflejada en la terminal, posteriormente, se creó un acceso exclusivo en el cual solo se ven reflejados los 5 productos con menores ventas totales, los cuales también se encuentran ordenados de menor a mayor y siendo los únicos reflejados en la terminal.

```

1501 # Creamos tabla para contar menores ventas de los productos #
1502 lista_ventas_menores = []
1503 for seq in range(1, len(lifestore_products) + 1):
1504     lista_ventas_menores.append([seq, 0])
1505
1506
1507 cantidad_registros_ventas_menores = len(lifestore_sales)
1508
1509 for seq in range(0, cantidad_registros_ventas_menores):
1510     producto2 = lifestore_sales[seq][1]
1511     lista_ventas_menores[producto2 - 1][1] = lista_ventas_menores[producto2 - 1][1] + 1
1512
1513
1514 from operator import itemgetter
1515 lista_ordenada2 = sorted(lista_ventas_menores, key = itemgetter(1), reverse = False)
1516
1517 print("\n Los productos con menos ventas son: \n")
1518
1519
1520 for seq in range(0, 5):
1521     print(lista_ordenada2[seq])
1522     print(lifestore_products[lista_ordenada2[seq][0] - 1][1])
1523 print("\n")
1524

```

→ La cuarta lista es la de “**búsquedas menores**” en la cual se encuentran reflejados los 10 productos con menor número de búsquedas en el sistema, ordenados de menor a mayor.

Para lograrlo, se pidió primero una lista que estuviera ordenada de menor a mayor con las búsquedas totales de todos los productos existentes, la cual no se ve reflejada en la terminal, posteriormente, se creó un acceso exclusivo en el cual solo se ven reflejados los 10 productos con menor número de búsquedas totales, los cuales también se encuentran ordenados de menor a mayor y siendo los únicos reflejados en la terminal.

```

1524
1525 # Creamos tabla para contar menores busquedas de los productos #
1526 lista_busquedas_menores = []
1527 for seq in range(1, len(lifestore_products) + 1):
1528     lista_busquedas_menores.append([seq, 0])
1529
1530
1531 cantidad_registros_busquedas_menores = len(lifestore_searches)
1532
1533 for seq in range(0, cantidad_registros_busquedas_menores):
1534     producto3 = lifestore_searches[seq][1]
1535     lista_busquedas_menores[producto3 - 1][1] = lista_busquedas_menores[producto3 - 1][1] + 1
1536
1537
1538 from operator import itemgetter
1539 lista_ordenada3 = sorted(lista_busquedas_menores, key = itemgetter(1), reverse = False)
1540
1541 print("\n Los productos con menos búsquedas son: \n")
1542
1543
1544 for seq in range(0, 10):
1545     print(lista_ordenada3[seq])
1546     print(lifestore_products[lista_ordenada3[seq][0] - 1][1])
1547 print("\n")
1548

```

→ La quinta lista es la de “**mejores reseñas**” en la cual se encuentran los resultados de los productos que han obtenido reseñas favorables por parte de los clientes.

Para lograrlo, se pidió primero una lista ordenada de las reseñas de todos los productos, la cual estaba ordenada de mayor a menor, esta lista no se muestra en la terminal, posteriormente, se creó un acceso exclusivo para mostrar los 5 productos que tienen mejores reseñas totales, los cuales se muestran ordenados de mayor a menor.

```

1547
1548 # Creamos tabla para contar reseñas de los productos #
1549 lista_mejores_resenas = []
1550 for seq in range(1, len(lifetore_products) + 1):
1551     lista_mejores_resenas.append([seq, 0, 0, 0, 0, 0]) # Contaremos reseñas de 1 a 5
1552
1553
1554 cantidad_registros_mejores_resenas = len(lifetore_sales)
1555
1556 for seq in range(0, cantidad_registros_mejores_resenas):
1557     producto4 = lifetore_sales[seq][1]
1558     mejor_resena = lifetore_sales[seq][2]
1559     lista_mejores_resenas[producto4 - 1][mejor_resena] = lista_mejores_resenas[producto4 - 1][mejor_resena] + 1
1560
1561
1562 lista_ordenada4 = sorted(lista_mejores_resenas, key = itemgetter(5), reverse = True)
1563
1564
1565 print("\n Los productos con mejores reseñas obtenidas por los clientes son: \n")
1566
1567 for seq in range(0, 5):
1568     #print(lista_ordenada4[seq])
1569     print(lifetore_products[lista_ordenada4[seq][0] - 1][1])
1570     print("(" + str(lista_ordenada4[seq][5]) + " reseñas de 5 estrellas)")
1571     print("(" + str(lista_ordenada4[seq][4]) + " reseñas de 4 estrellas)")
1572     print("(" + str(lista_ordenada4[seq][3]) + " reseñas de 3 estrellas)")
1573     print("(" + str(lista_ordenada4[seq][2]) + " reseñas de 2 estrellas)")
1574     print("(" + str(lista_ordenada4[seq][1]) + " reseñas de 1 estrella) \n")
1575

```

→ La sexta lista es la de “**menores reseñas**” en la cual se encuentra reflejado el resultado de los productos con menor número de reseñas favorables por parte de los clientes.

Para obtener este resultado, se realizó una lista ordenada de las reseñas de los productos totales, la cual se encuentra ordenada de menor a mayor, esta lista no se muestra en la terminal, posteriormente, se creó un acceso en el cual solo se encuentran los 5 productos con menor número de reseñas totales, los cuales se encuentran ordenados de menor a mayor.

```

1575
1576 # Creamos tabla para contar peores reseñas de los productos #
1577
1578 lista_menores_resenas = []
1579 for seq in range(1, len(lifetore_products) + 1):
1580     lista_menores_resenas.append([seq, 0, 0, 0, 0, 0]) # Contaremos las reseñas del numero 1 al 5
1581
1582
1583 cantidad_registros_menores_resenas = len(lifetore_sales)
1584
1585 for seq in range(0, cantidad_registros_menores_resenas):
1586     producto5 = lifetore_sales[seq][1]
1587     menor_resena = lifetore_sales[seq][2]
1588     lista_menores_resenas[producto5 - 1][menor_resena] = lista_menores_resenas[producto5 - 1][menor_resena] + 1
1589
1590
1591 lista_ordenada5 = sorted(lista_menores_resenas, key = itemgetter(1,2), reverse = True)
1592
1593
1594 print("\n Los productos con peores reseñas obtenidas por los clientes son: \n")
1595
1596 for seq in range(0, 5):
1597     #print(lista_ordenada4[seq])
1598     print(lifetore_products[lista_ordenada5[seq][0] - 1][1])
1599     print("(" + str(lista_ordenada5[seq][1]) + " reseñas de 1 estrellas)")
1600     print("(" + str(lista_ordenada5[seq][2]) + " reseñas de 2 estrellas)")
1601     print("(" + str(lista_ordenada5[seq][3]) + " reseñas de 3 estrellas)")
1602     print("(" + str(lista_ordenada5[seq][4]) + " reseñas de 4 estrellas)")
1603     print("(" + str(lista_ordenada5[seq][5]) + " reseñas de 5 estrellas) \n")
1604

```

→ La séptima lista es la de “**ventas promedio meses**” en la cual se encuentran reflejados los ingresos totales y el promedio de cada uno, recaudados de las ventas reflejadas de todos los productos. De igual manera, en esta lista se encuentran los **ingresos** monetarios totales y el promedio mensual.

Para obtener este resultado, se realizó una lista ordenada en la cual se pide información de las carpetas necesarias para poder obtener los datos de ventas y costos, con estos datos, se hicieron diferentes comandos, de los cuales se obtuvo

- El promedio mensual y cantidad total, tanto de ventas como de ingresos monetarios.
- Se ordenaron los meses de mayor a menor dependiendo de sus ingresos.
- Se ordenaron los meses de mayor a menor dependiendo de sus ventas.

```

1633 print("\n Las cantidades totales y promedio obtenidas son las siguientes \n /
1634
1635 print("La cantidad total de ventas en el año es de: ", ventas_totales)
1636 print("La cantidad total de ingresos en el año es de: $", ingresos_totales)
1637 print("La cantidad promedio de ventas por mes es: ", promedio_ventas_mes)
1638 print("La cantidad promedio de ingresos por mes es de: $", promedio_ingreso_mes)
1639
1640 print("\n")
1641
1642 meses_año = ["Enero", "Febrero", "Marzo", "Abril", "Mayo",
1643             "Junio", "Julio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre"]
1644
1645         # Ordenamos por número total de cantidades de ventas #
1646
1647 lista_ordenada7 = sorted(lista_ventas_promedio_meses, key = itemgetter(1), reverse = True)
1648
1649 print("\n Si ordenamos de mayor a menor cantidad de ventas, los meses se presentan de la siguiente manera: \n")
1650
1651 for seq in range(0, 12):
1652     print(meses_año[int(lista_ordenada7[seq][0]) - 1], "tiene " + str(lista_ordenada7[seq][1]) + " ventas total
1653
1654         # Ordenamos por cantidad de ingresos #
1655 lista_ordenada8 = sorted(lista_ventas_promedio_meses, key = itemgetter(2), reverse = True)
1656
1657
1658 print("\n Si ordenamos de mayor a menor cantidad de ingresos, los meses quedan de la siguiente manera: \n")
1659
1660 for seq in range(0, 12):
1661     print(meses_año[int(lista_ordenada8[seq][0]) - 1], "tiene $" + str(lista_ordenada8[seq][2]) + " de ingresos
1662

```

```

1607         # Creamos tabla para contar ventas mensuales totales y promedio de los productos #
1608
1609 lista_ventas_promedio_meses = []
1610 for seq in range(1, 13):
1611     if seq < 10:
1612         lista_ventas_promedio_meses.append(["0" + str(seq), 0, 0]) # Comenzamos a contar ganancias y ventas ac
1613     else:
1614         lista_ventas_promedio_meses.append([str(seq), 0, 0])
1615 cantidad_registros_ventas_pmeses = len(lifefore_sales)
1616 for seq in range(0, cantidad_registros_ventas_pmeses):
1617     producto6 = lifefore_sales[seq][1]
1618     mes = lifefore_sales[seq][3][3:5]
1619     if lifefore_sales[seq][4] != 1: # No tomamos en cuenta los productos que tengan devolución
1620         for seq1 in range(0, 12):
1621             if mes == lista_ventas_promedio_meses[seq1][0]:
1622                 lista_ventas_promedio_meses[seq1][1] = lista_ventas_promedio_meses[seq1][1] + 1
1623                 lista_ventas_promedio_meses[seq1][2] = lista_ventas_promedio_meses[seq1][2] + lifefore_product
1624 ventas_totales = 0
1625 ingresos_totales = 0
1626 for seq in range(0, len(lista_ventas_promedio_meses)):
1627     ventas_totales = ventas_totales + lista_ventas_promedio_meses[seq][1]
1628     ingresos_totales = ingresos_totales + lista_ventas_promedio_meses[seq][2]
1629
1630 promedio_ventas_mes = ventas_totales/12
1631 promedio_ingreso_mes = ingresos_totales/12
1632
1633 print("\n Las cantidades totales y promedio obtenidas son las siguientes \n")
1634
1635 print("La cantidad total de ventas en el año es de: ", ventas_totales)
1636 print("La cantidad total de ingresos en el año es de: $", ingresos_totales)

```

★ Link del código en github: <https://github.com/maguanzu/Proyecto-final-1->

RESOLUCIÓN DEL PROBLEMA

Tomando en cuenta los resultados obtenidos anteriormente, se dará una resolución y consejos para los productos rezagados y los más vendidos.

En el caso de los productos más vendidos por la empresa **LifeStore**, se sugiere continuar adquiriendo estos productos, ya que han sido ventas importantes. Por otro lado se sugiere poner los productos que tienen menores ventas en promoción, rebajas, o algo parecido para poder venderlos y no perder todo el dinero de la inversión, y claramente, no volver a adquirir estos productos, si es que algún cliente estuviera interesado en adquirir alguno de estos cuando ya no estén en existencia, lo que se le puede decir que solo se adquieren por pedido especial y pedir un porcentaje de adelanto para asegurar la compra/venta de este producto.

Si nos basamos en los productos de mejores reseñas, de igual manera recomiendo tomar en cuenta los comentarios dados por los clientes, para así saber si es la marca correcta para el producto, ya que, el producto puede ser funcional pero podría serlo aún más y mejor. Lo mismo pasa con los productos con malas reseñas, tal vez lo malo no es el producto, si no la marca en específico y lo que se puede hacer en este caso, es cambiar de proveedores o de sucursales dónde se adquiere el producto.

Si **LifeStore** tiene dudas de saber qué productos serían los apropiados para reemplazar los productos que no se vendieron bien. lo que podría hacer es una encuesta con sus clientes físicos y en su página de internet, en la cual pregunte “cuáles son los productos que más les interesaría comprar con ellos”, “los que más utilizan y no pueden conseguir fácilmente”, o algo por el estilo, así esto les ayudará a conocer más sobre los gustos de sus clientes y tal vez podrían impulsar el negocio con productos complementarios a los que ya venden.

Por último, se les sugiere hacer un estudio interno y externo, en el cual se vea reflejado “qué es lo que tenían u ofrecían en los primeros meses, que en los últimos ya no”, “si han tenido mucha rotación de personal y a qué se ha debido”, “qué opinan los clientes sobre el trato que se tenía al inicio en comparación con el que le ofrecen los empleados actualmente”, y por último, los dueños de la empresa y principales socios podrían hacer una introspección y autoevaluación de lo que ha sucedido a lo largo de los años de esta empresa, para así saber en qué pueden mejorar aún más.

CONCLUSIÓN

LifeStore debería cambiar los artículos que han tenido menores ventas y poner estos en oferta para así ya no tener productos rezagados por más tiempo y poder rescatar un poco del dinero invertido en estos, de igual manera, se recomienda mantener en monitoreo constante los productos que presentan peores reseñas, ya que podrían ser los siguientes productos en dejar de venderse.

Sería bueno hacer un estudio de mercado para entender de una mejor manera el panorama y las necesidades de los locatarios y así poder poner a la venta productos que sean de mayor interés para los clientes y posibles clientes de la empresa.