
LIFESTORE

Reporte Anual 2020



Reporte anual que se presenta de acuerdo con las disposiciones de carácter general aplicables a las emisoras de valores y a otros participantes del mercado por el año terminado el 31 de diciembre de 2020.

ÍNDICE

	Página
I. Introducción	3
II. Análisis por categoría	
a) Procesadores	4
b) Tarjetas de video	6
c) Tarjetas madre	8
d) Discos duros	9
e) Memorias USB	10
f) Pantallas	10
g) Bocinas	10
h) Audífonos	10
III. Análisis general y conclusiones	11
IV. Análisis de los datos mediante el uso de la herramienta Python	14

Introducción

Hoy en día, un buen modelo de negocios no es suficiente para lograr que un negocio prospere, ya sea mayorista o minorista. En los últimos años, con el avance de la tecnología se han hecho cada vez más populares los *data-driven business model* y no es para menos, considerando que la importancia de los datos no se puede poner en duda y es un activo importante, sino vital para cualquier giro de negocio. El análisis de datos aplicado a los negocios puede obtenerse de distintas maneras, por ejemplo, se pueden recabar datos libres que se encuentran en internet y acotarlos al mercado que buscas atacar utilizando herramientas como la minería de datos. Otro ejemplo es utilizar los datos que se generan a través de la interacción con los clientes mediante el servicio a los mismos. O bien, se pueden mezclar ambos métodos para la generación de nuevas analíticas y datos funcionales para crear estrategias que mejoren la rentabilidad del negocio.

En el presente reporte se hace uso de la herramienta Python y Excel para la generación de análisis de los datos de un periodo de 12 meses y posteriormente sugerir estrategias para resolver las problemáticas que enfrenta la empresa Lifestore. Esta empresa se dedica a vender una amplia gama de artículos tecnológicos, sin embargo, 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.

Análisis por categoría

Procesadores

Los procesadores representan un 37% de las ventas totales de la empresa, además de tener buena aceptación por parte de los clientes con una calificación de 4.4 de 5 y únicamente 1 devolución de las 104 ventas efectuadas. En cuanto al análisis de producto por producto en esta categoría, podemos ver que los productos:

Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth

Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth

Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire

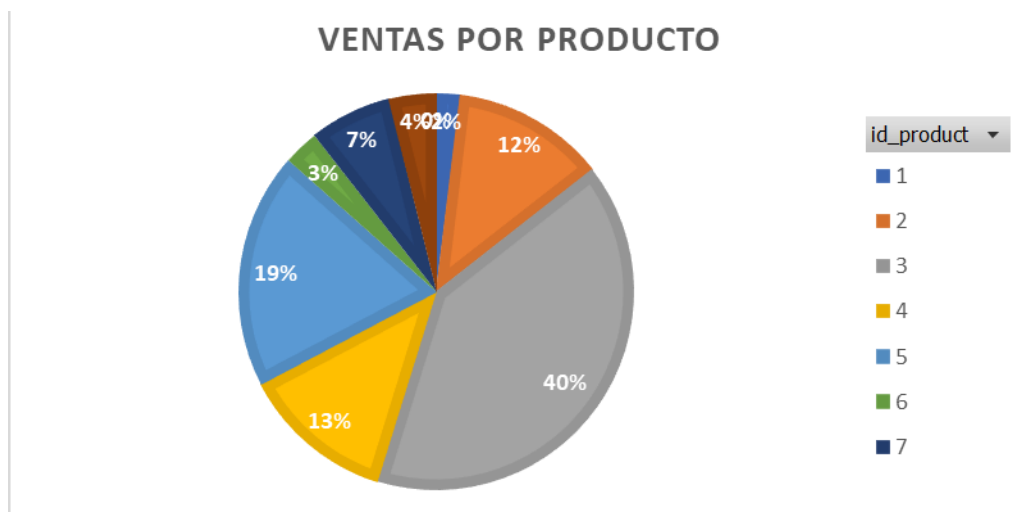
Procesador Intel Core i9-9900K, S-1151, 3.60GHz, 8-Core, 16MB Smart Cache (9na. Generación Coffee Lake)

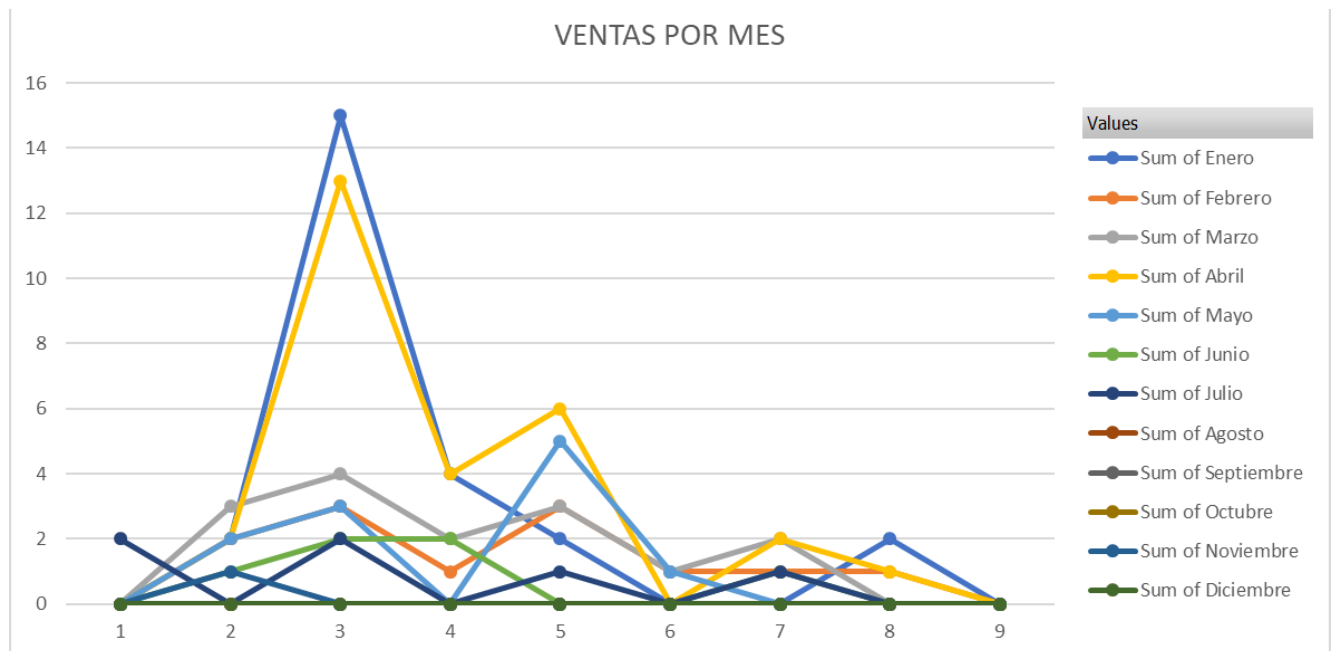
Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)

A pesar de ser de los productos con mayores ventas, también son los que cuentan con mayor inventario; por lo que convendría no comprar a demasía. Por otro lado, el producto:

Procesador Intel Core i3-8100, S-1151, 3.60GHz, Quad-Core, 6MB Smart Cache (8va. Generación - Coffee Lake)

Es el único en su categoría que no tuvo ventas en 2020, por lo que se le debe hacer más promoción ya que no tiene un precio muy elevado. De igual manera, los procesadores son los segundos productos, con los que cuenta la empresa, más buscados por los usuarios. Por lo tanto, es una buena opción invertir en la mercadotecnia en esta categoría.





En cuanto a las ventas a lo largo del año, podemos observar que enero, abril y mayo son los meses con mayores ventas en esta categoría. Particularmente el producto más vendido es:

Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth

El periodo de menores ventas es de agosto a diciembre. Sin embargo, la categoría en su conjunto representa el mayor ingreso de la empresa con la cantidad anual de \$371,726.00.

Tarjetas de video

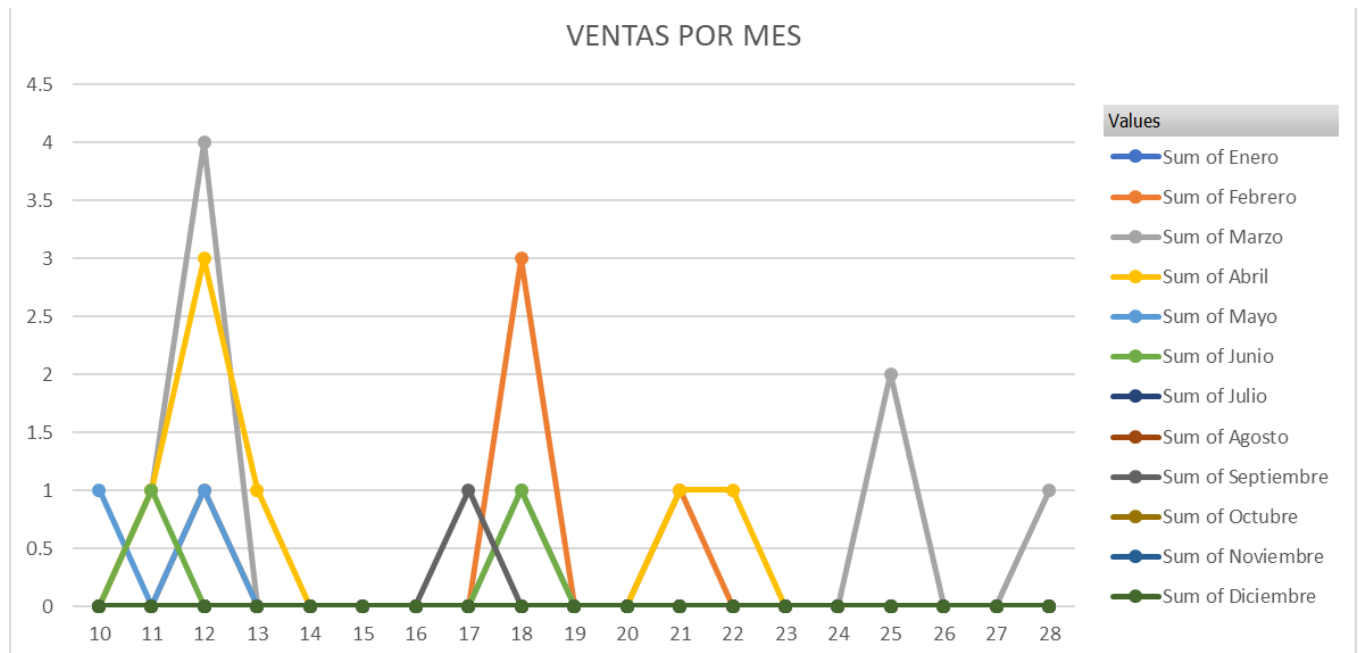
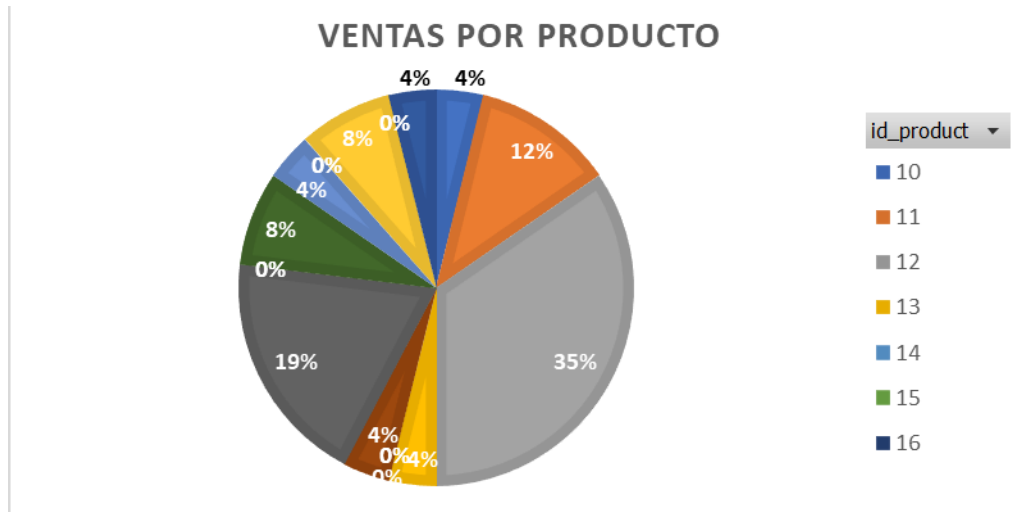
Las tarjetas de video representan sólo el 9% de las ventas totales de la empresa, además, prácticamente la mitad de los productos en esta categoría no tuvieron ventas en todo 2020. No obstante, en su mayoría no se cuenta con gran cantidad de inventario, a excepción de los productos:

Tarjeta de Video VisionTek AMD Radeon HD 5450, 1GB DDR3, PCI Express x16 2.1

Tarjeta de Video VisionTek AMD Radeon HD5450, 2GB GDDR3, PCI Express x16

Tarjeta de Video EVGA NVIDIA GeForce GT 710, 2GB 64-bit GDDR3, PCI Express 2.0

Siendo el primer producto aquí mostrado el de mayor inventario de la categoría con 180 piezas. Por lo que convendría reevaluar la compra de estos productos a futuro. En general, las reseñas de los productos vendidos son buenas, con una calificación de 4.3 de 5 y una devolución de las 26 ventas efectuadas. No obstante, esta categoría es de las menos buscadas, por lo que, convendría enfocarse en aquellos productos con mayores ventas de la categoría y hacerles publicidad. O bien, poner en oferta aquellos otros que no han tenido éxito.



En las ventas por mes, vemos que las mayores fueron en febrero, marzo y abril. De igual manera, en el periodo de julio a diciembre las ventas son prácticamente nulas. Aunque, representa el segundo lugar de mayores ingresos para la compañía con una cantidad anual de \$136,224.00.

Tarjetas madre

Las tarjetas madre representan el 17% de las ventas totales de la empresa, sin embargo, son los productos con mayor tasa de devolución y peores reseñas de los clientes, teniendo como resultado una calificación de 3.4 de 5 y 5 devoluciones de las 49 ventas efectuadas. Al mismo tiempo, esta categoría es de las más buscadas por los clientes, por lo que convendría, más que enfocarse en la mercadotecnia, enfocarse en estratos específicos de estos productos. Es decir,

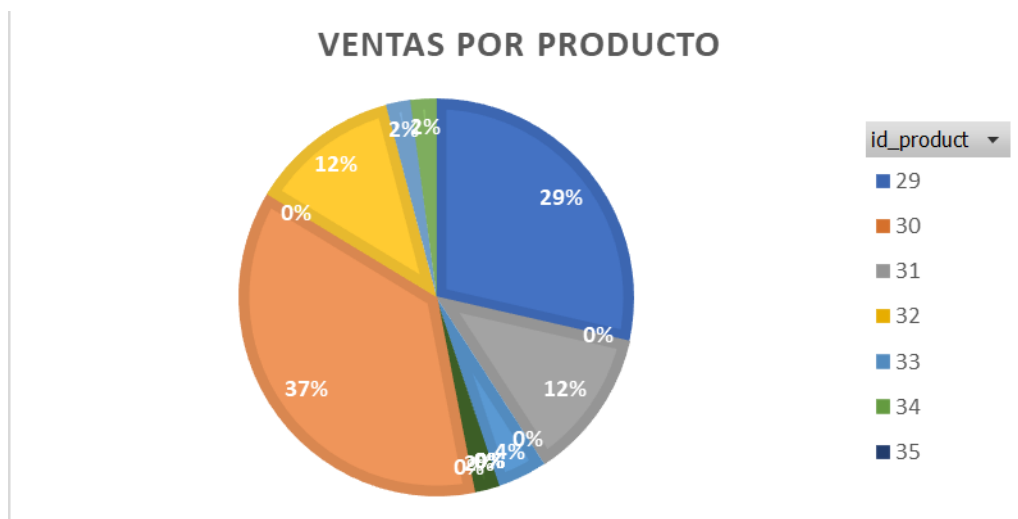
se observa que los productos de menor precio son los que más se venden, así como el más caro.

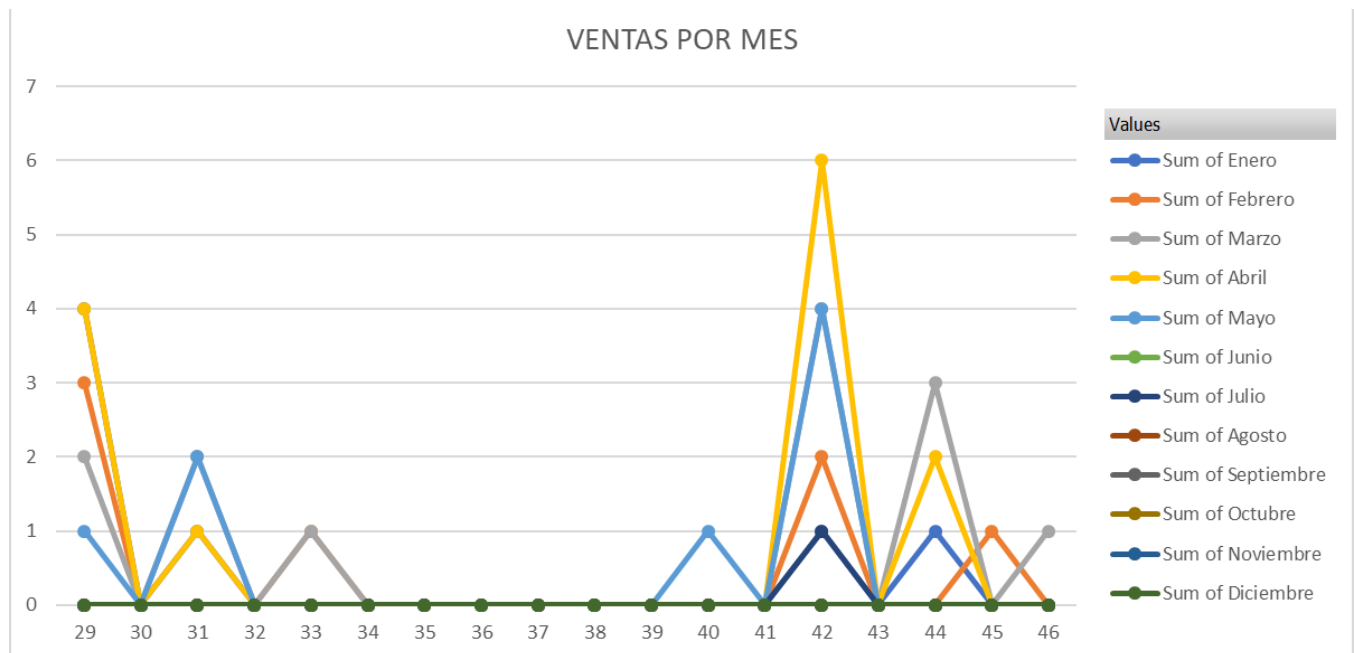
Esto parece lógico ya que el negocio de reventa de tarjetas madre es bastante competitivo, por lo que los clientes o buscan los precios más bajos o aquellos productos de calidad aunque sean de “lujo”. Por esto, convendría enfocarse en esos dos extremos de los productos y cuidando el stock de estos, ya que en general es bastante alto para la cantidad que se vende.

Por otro lado, es importante resaltar que el producto:

Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD

Tiene un 50% de tasa de devolución y malas reseñas, entonces conviene evaluarlo o en su defecto, sacarlo del mercado. Por último, en cuanto a los ingresos por ventas, las tarjetas madre, ocupan el tercer lugar de la gama de categorías con las que cuenta la empresa.





En relación con las ventas por mes, marzo abril y mayo son aquellos meses con mayores ventas, pero en el periodo de junio a diciembre, no hay ventas.

Discos duros

Los discos duros representan el 33% de las ventas totales de la empresa, además de tener buena aceptación por parte de los clientes con una calificación de 4.8 de 5 y únicamente 1 devolución de las 94 ventas efectuadas. En cuanto al análisis de producto por producto en esta categoría, podemos ver que los productos:

SSD Addlink Technology S70, 512GB, PCI Express 3.0, M.2

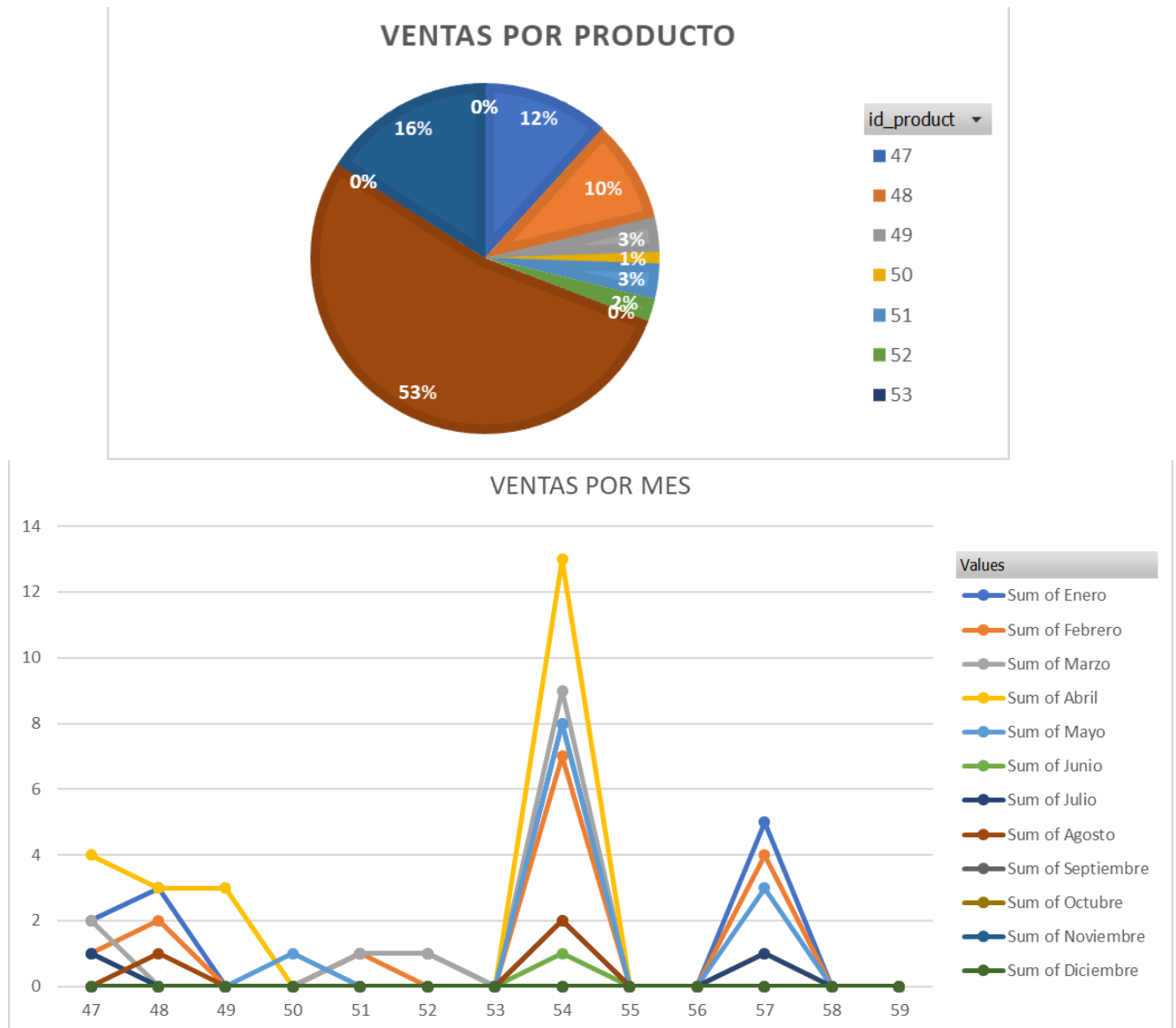
SSD para Servidor Supermicro SSD-DM128-SMCMVN1, 128GB, SATA III, mSATA, 6Gbit/s

SSD para Servidor Lenovo Thinksystem S4500, 480GB, SATA III, 3.5", 7mm

SSD para Servidor Lenovo Thinksystem S4510, 480GB, SATA III, 2.5", 7mm

SSD Samsung 860 EVO, 1TB, SATA III, M.2

Son aquellos que no han tenido ventas de la categoría, sin embargo tienen poco inventario, por lo que no representan un gran problema pero sí convendría una estrategia de promoción. Estos productos se encuentran en el mismo rango de precio de los productos más exitosos de la categoría; y los discos duros son los productos más buscados con 463 búsquedas por lo que una estrategia de discriminación de precios de segundo grado sería la adecuada.



Los ingresos de esta categoría es el cuarto lugar en los ingresos totales de la empresa. Mientras que en las ventas mensuales, abril, febrero, marzo y mayo son los meses con mayores ventas y el periodo de septiembre a diciembre no tiene ventas.

Memorias USB

Las memorias USB no representan ingresos significativos para la empresa Lifestore, por lo que si se quiere expandir su comercialización se debe considerar la diversificación de productos con poco inventario para comenzar ya que es una categoría con nulas búsquedas por parte de los clientes.

Pantallas

Las pantallas únicamente representan el 1% de los ingresos de la empresa y es una categoría con varios productos y mucho inventario. No obstante, las 2 ventas efectuadas obtuvieron las mejores reseñas con una calificación de 5 de 5 y las búsquedas de los clientes son 56 pero pueden aumentar. Por lo tanto, es una categoría con potencial de crecimiento, pero hace falta enfocarse en vender el inventario primero, antes de diversificar con más productos.

Bocinas

Las bocinas, al igual que las pantallas, representan el 1% de los ingresos de la empresa y cuenta con varios productos, pero a diferencia de las pantallas, no cuenta con un gran inventario. De igual manera, las 2 ventas efectuadas tienen una calificación alta de 4.5 de 5. Sin embargo, esta categoría no cuenta con tantas búsquedas, por lo que lo óptimo sería poner en oferta el inventario y enfocarse en productos con mayor demanda.

Audífonos

Los audífonos representan el 2% de los ingresos de la empresa y es una categoría con varios productos y mucho inventario. Los siguientes productos son los únicos que han tenido ventas de su categoría y muy pocos en comparación con su inventario:

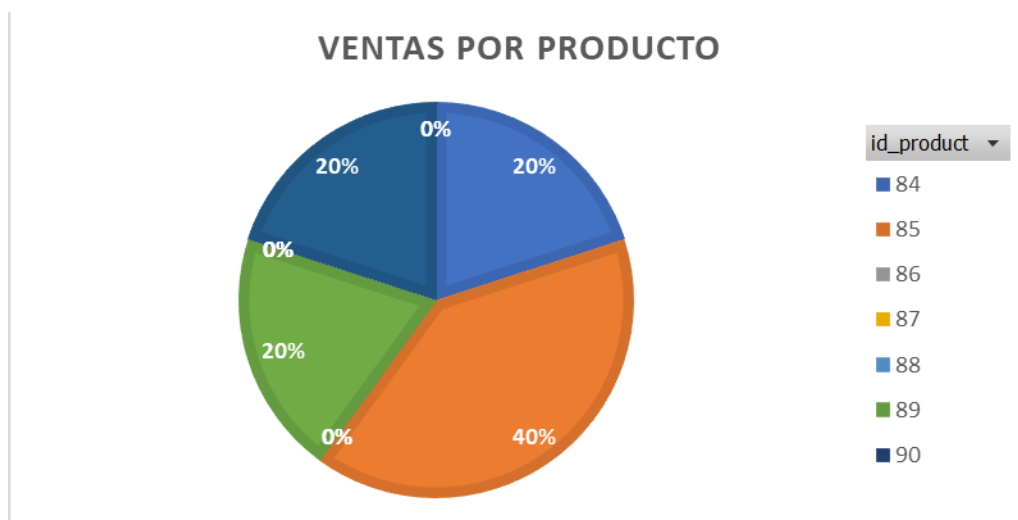
Logitech Audífonos Gamer G332, Alámbrico, 2 Metros, 3.5mm, Negro/Rojo

Logitech Audífonos Gamer G635 7.1, Alámbrico, 1.5 Metros, 3.5mm, Negro/Azul

Cougar Audífonos Gamer Phontum Essential, Alámbrico, 1.9 Metros, 3.5mm, Negro

HyperX Audífonos Gamer Cloud Flight para PC/PS4/PS4 Pro, Inalámbrico, USB, 3.5mm, Negro

No obstante, al igual que las pantallas, son productos con búsquedas considerables por parte de los clientes por lo que conviene la misma estrategia que la categoría de pantallas.

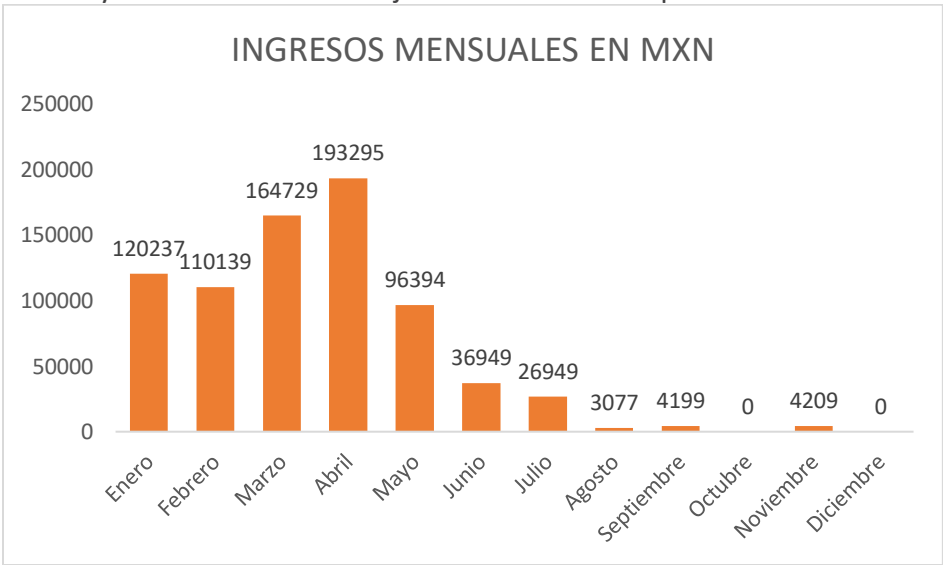




En cuanto a las ventas por mes, se observa que las ventas se efectuaron en enero, abril y mayo.

Análisis general

En general, la empresa Lifestore presenta mayores ingresos en los primeros cuatro meses del año, disminuye los siguientes tres meses y los últimos cinco meses son los de menores ventas, siendo septiembre y noviembre los de mejores ventas de este periodo.



Los productos con mayores ventas son los procesadores y discos duros. De igual manera, las pantallas y los audífonos son un mercado con potencial. Por otro lado, las tarjetas madre, a pesar de representar un ingreso alto, requiere de una mejora en su gestión y calidad el cual se podría invertir en otras categorías con mejores reseñas o mayores búsquedas.



La solución al problema de acumulación de inventario se puede resolver desde una estrategia de mercadotecnia, donde se defina primero el segmento de mercado en el cual los productos tienen mayor éxito y posteriormente se haga una discriminación de precios de acuerdo con los resultados. De igual forma, la publicidad se debe enfocar en aquellos productos y/o categorías que han tenido mejores respuestas por parte de los clientes pero que no han sido adquiridos por falta de conocimiento de su existencia.

También sería buena idea implementar un software de gestión de almacén para que automatice las órdenes de reposición y no tener productos rezagados o faltantes. Otra manera de atacar dicho problema es reduciendo el inventario y en su lugar invertir en mejoras en logística y distribución para reducir los costos en la cadena de valor y así poder ofrecer precios más competitivos y vender más volumen. Mientras que, en el caso de la publicidad, se podrían aplicar estrategias de economía del comportamiento como lo son la familiaridad, el efecto arrastre y la disponibilidad.

Por otro lado, el tema de las búsquedas conviene atacarlo con publicidad en redes sociales o en los buscadores más populares para que el algoritmo cambie y así poder acercarse más a los clientes. Otra forma, sería con embajadores de acuerdo con la edad de quienes han comprado los productos para crear una mejor fidelización con los clientes.

Asimismo, para hacer un análisis más exhaustivo, habría que evaluar el servicio a clientes y si los tiempos de entrega (en el caso de ventas virtuales) son los correctos. En cuanto a la venta física, se debe evaluar la manera en que los productos son exhibidos y sus especificaciones presentadas al tratarse de productos tecnológicos.

En conclusión, Lifestore debería enfocarse en fortalecer sus categorías de producto con mayores ventas cuidando la cantidad que compran a proveedores para no incurrir en rezagos. Posteriormente, dejar ir categorías en las que no es fuerte o que no son una oportunidad emergente para enfocar esos recursos en la mejora de sus procesos logísticos y de fidelización de sus clientes para poder bajar sus precios y mejorar su margen de ganancia a largo plazo.

Análisis de los datos mediante el uso de la herramienta Python

A continuación, se presenta el código utilizado para realizar la plataforma de consulta del reporte anual para el administrador de la empresa Lifestore.

Comenzando por el menú donde se utilizó la siguiente estructura:

```
lista_a=['lala', 'lolo','stemem']
lista_b=['123','456','789']
lista_c=['admin','user','user']
name = input('Ingrese su usuario: ')
pw = input('Ingrese su contraseña: ')

correcto = 0
while correcto != 1:
    if name in lista_a and pw in lista_b:
        correcto = 1
        if lista_a.index(name) == lista_b.index(pw):
            if lista_c[lista_a.index(name)] == 'admin':
                print('Bienvenido administrador')
                input('Escribe el número de la acción que desees consultar: ')
            else:
                print('Bienvenido usuario')
        else:
            print('Usuario o contraseña invalido, intentalo de nuevo:')
            name = input('Ingrese su usuario: ')
            pw = input('Ingrese su contraseña: ')
    else:
        print('Usuario o contraseña invalido, intentalo de nuevo:')
        name = input('Ingrese su usuario: ')
        pw = input('Ingrese su contraseña: ')
    correcto = 0
```

Primero se crearon las listas, una para los nombres de los usuarios, otra para las contraseñas y por último otra para verificar que sea el administrador o algún usuario. Posteriormente con el if anidado se comprueba primero que cada elemento de la lista a coincida con su elemento correspondiente de la lista b. Después se verifica si los datos ingresados pertenecen al valor admin o user, de ser admin se da acceso a los datos, de ser usuario únicamente se da la bienvenida. Pero, si los datos ingresados no se encuentran dentro de las listas (con que uno sea incorrecto, ya que se utilizó and), te dirá que lo intentes de nuevo. La variable extra agregada fue correcto: que se utiliza para comprobar si los datos son correctos o no y de esa manera terminar con el ciclo while.

Procedemos con la primera consigna que es el top 50 de mayores ventas (en este caso sólo hubo 42 productos con ventas)

```

contador = 0
total_ventas = [] ##[[id,nombre,contador], [id2,contador2]]

for producto in lifestore_products:
    for venta in lifestore_sales:
        categoria = producto[3]
        if producto[0] == venta[1]:
            contador += 1
    formato_ideal = [producto[0],producto[1],contador,categoria]
    total_ventas.append(formato_ideal)
    contador = 0

aux = None
for i in range (0,len(total_ventas)):
    for j in range (0,len(total_ventas)):
        if(total_ventas[i][2] > total_ventas[j][2]):
            aux = total_ventas[i]
            total_ventas[i] = total_ventas[j]
            total_ventas[j] = aux

counter = 1
for i in range(0,len(total_ventas)):
    if total_ventas[i][2] != 0: # Omite las búsquedas con 0
        print(counter,'.- El producto: ', total_ventas[i][0], '\n Se vendió: ', total_ventas[i][2],
              'veces', '\n Pertenece a la categoría: ', total_ventas[i][3])
        counter += 1

```

Aquí lo que se hizo fue encontrar el id del producto en dos listas diferentes accediendo a sus índices, para que si estos coincidían, se agregaran a una nueva lista con las ventas y su categoría. Todo esto mediante un ciclo for que va iterando sobre las listas para que sea dinámico. Posteriormente, en el segundo bloque de código se hace un “método burbuja” donde con el tamaño de la nueva lista creada y accediendo de nuevo a sus índices se compara con ayuda de una variable auxiliar sus valores para ir iterando y de esta manera ordenar la lista de mayor a menor. Por último, se imprime el resultado en el formato deseado. De igual manera para la lista de productos con menores ventas se utiliza el mismo código a excepción del rango en el último ciclo for para únicamente mostrar los productos con cero ventas.

```

counter = 1
for i in range(0,55): ##son 54 con
venta cero
    print(counter,'.- El producto: ',
          total_ventas[i][1], '\n Se vendió: ',
          total_ventas[i][2],'veces', '\n
          Pertenece a la categoría: ',
          total_ventas[i][3])
    counter += 1

```

La siguiente consigna fue obtener el top 100 de mayores búsquedas y top 100 de menores búsquedas (en este caso sólo hay 96 productos).

```

contador = 0
total_búsquedas = [] ##[[id,nombre,contador], [id2,contador2]]

for producto2 in lifestore_products:
    for busqueda in lifestore_searches:
        categoria = producto2[3]
        if producto2[0] == busqueda[1]:
            contador += 1
    formato_ideal2 = [producto2[0],producto2[1],contador,categoria]
    total_búsquedas.append(formato_ideal2)
    contador = 0

aux = None
for i in range (0,len(total_búsquedas)):
    for j in range (0,len(total_búsquedas)):
        if(total_búsquedas[i][2] > total_búsquedas[j][2]):
            aux = total_búsquedas[i]
            total_búsquedas[i] = total_búsquedas[j]
            total_búsquedas[j] = aux

counter = 1
for i in range(0,len(total_búsquedas)):
    print(counter,'.- El producto: ', total_búsquedas[i][1], '\n Se buscó: ',
          total_búsquedas[i][2],'veces', '\n Pertenece a la categoría: ',
          total_búsquedas[i][3])
    counter += 1

```

Para esta consigna se siguió la misma lógica del punto anterior, solo que iterando sobre una lista diferente para encontrar las búsquedas. Para el ordenamiento se utilizó el mismo método y sólo se cambió el signo de mayor que a menor que. En este caso se tomó como rango la longitud de la lista entera para abarcar los 96 elementos.

Para obtener las dos listas de 20 productos cada una para los productos con mejores y peores reseñas lo que se hizo fue lo siguiente

```

ranked_products = []
product_stars = [ [] for i in range(len(lifestore_products)) ]

for sale in lifestore_sales:
    product_stars[sale[1] - 1].append(sale[2])

for product in lifestore_products:
    category = product[3]
    if(len(product_stars[product[0]-1])):
        ranked_products.append( (sum(product_stars[product[0] - 1]) / len
                                (product_stars[product[0] - 1]), product[1], category) )
    else:
        ranked_products.append( (0.0, product[0], category) )

ordered_ranking = sorted(ranked_products, reverse=True)

counter = 1
for i in range(0,20):
    if ordered_ranking[i][0] != 0.0:
        print(counter,'.- El producto: ', ordered_ranking[i][1], '\n Tiene
          una reseña de: ', ordered_ranking[i][0], '\n Pertenece a la
          categoría: ', ranked_products[i][2])
        counter += 1

```


En este código se pasa por cada una de las ventas tomando en cuenta el id del producto y la calificación. De esta manera, `product_stars` tiene en cada lugar del producto una lista vacía (definida al inicio) y con cada venta se agrega la calificación al lugar correspondiente del producto (id) (lugar 0 es donde va el producto 1). Así, al final de ese ciclo se crea una lista de listas de calificaciones por producto.

Con `sum` se hace el promedio la suma de todas las calificaciones de ese producto entre el total de calificaciones. Después para cada uno de los productos se calcula por cada uno el promedio de su lista correspondiente a su lugar. También se obtiene el nombre de los productos en la tupla de la lista `ranked_products`.

En este caso, en lugar de utilizar el método burbuja, se implementó `sorted` para ordenarla de mayor a menor si `reverse=True` y viceversa si `reverse=False`. En este caso, `len` se utiliza para saber qué tantas calificaciones tiene cada producto y en caso de que fueran 0, se añade un 0.0 a la lista final y se salta el cálculo (ya sabes que a la computadora no le gustan las divisiones entre 0). Dado que se agregó el 0.0 para la lista de peores reseñas se modifica el rango considerando que se dejan fuera los productos que no tuvieron reseña alguna.

Para mostrar las devoluciones en la lista de productos se utilizó lo siguiente

```
devoluciones = []
contador = 0
for productodev in lifestore_products:
    for productovend in lifestore_sales:
        catego = productodev[3]
        if productodev[0] == productovend[1]:
            if productovend[4] == 1: ##1 es que fue devuelto
                contador += 1
if contador != 0: ##omite los 0 que no fueron devueltos
    formato_ideal3 = [productodev[1], contador, catego]
    devoluciones.append(formato_ideal3)
    contador = 0

aux = None
for i in range(0, len(devoluciones)):
    for j in range(0, len(devoluciones)):
        if devoluciones[i][1] > devoluciones[j][1]:
            aux = devoluciones[i]
            devoluciones[i] = devoluciones[j]
            devoluciones[j] = aux

counter = 1
for i in range(0, len(devoluciones)):
    if devoluciones[i][0] != 0:
        print(counter, '- El producto: ', devoluciones[i][0], '\n Se devolió: ',
              devoluciones[i][1], 'veces', '\n Pertenece a la categoría: ', devoluciones[i][2])
        counter += 1
```

La lógica es similar a los de los puntos anteriores en ventas y reseñas, con la diferencia de que en el último `for` se hace una exclusión de los valores que son igual a cero (en este caso significa que no tuvieron devolución)

Por último, se obtuvieron los ingresos totales y divididos por meses

```

precio = []
mes = []
for i in range(0,len(lifestore_products)):#se hace una lista con los precios pero se
itera muchas veces
    precio.append(lifestore_products[i][2])
for i in range(0,len(lifestore_sales)):
    venta2 = lifestore_sales[i][1]-1 ##se saca el id
    fechav = lifestore_sales[i][3].split("/") [1]##se saca el mes
    if len(mes) == 0:
        mes.append([precio[venta2], fechav])

    else:
        bandera = False
        for m in mes:
            if m[1] == fechav:
                m[0] += precio[venta2]
                bandera = True
        if bandera == False:
            mes.append([precio[venta2], fechav])

aux=[0,0]
for i in range (0,len(mes)):
    for j in range (0,len(mes)):
        if(mes[i][1] < mes[j][1]):
            aux[0] = mes[i]
            mes[i] = mes[j]
            mes[j] = aux[0]
    print(aux)

bigtotal = 0
for i in range (0,len(mes)):
    aux = mes[i][0]
    bigtotal = bigtotal + aux
promedio = bigtotal/12

for i in range(0,len(mes)):
    print("Mes:",mes[i][1],"Ventas de: $" +str(mes[i][0]))
print("\nLos meses 10 y 12 no tuvieron ventas\n\nEl total de ventas anuales fue: $:",
str(bigtotal))
print("El promedio mensual fue: $",str(promedio))

```

En este código se utilizó el Split ya que las fechas vienen divididas con "/" y de esta manera se rompe la cadena str para poder acceder al mes correspondiente. De igual manera en el método burbuja, la variable auxiliar es de dos elementos ya que se crea una lista de listas.

Link Github: <https://github.com/Isamar6v/REPORTE-01-GARCIA-ISAMAR/find/master>