



Proyecto 01

Data Science

Contreras Aguilar Fernando

Curso Data Science. EmTECH

07 de Septiembre de 2020



[github/fershous](https://github.com/fershous)

fercho_0412@hotmail.com
2281 39 69 82
www.github.com/fershous

Indice

Introducción.....	2
Definición del código.....	3
Variables.....	4
Importaciones básicas.....	5
Código.....	6
Solución al problema.....	9
Conclusión.....	13

Introducción

“[Data scientists are] able to think of ways to use data to solve problems that otherwise would have been unsolved, or solved using only intuition.”
-Peter Skomoroch, Former Principal Data Scientist at LinkedIn.

Desde la capacidad de tomar decisiones, hasta predecir comportamientos a futuro, son algunas de las capacidades que la ciencia de datos ha traído al mundo. Cualquiera se queda corto ennumerando las posibilidades que esta ciencia nos ha abierto.

Lifestore es una empresa dedicada a la venta de artículos tecnológicos. En el siguiente documento, veremos expuesto un ejemplo de la Ciencia de Datos aplicada a las ventas de una empresa y cómo esta es el pilar para la toma de decisiones de cualquier área de una empresa, ya sea grande o pequeña, las aplicaciones de la ciencia de datos son casi infinitas.

Descripción del código

Proyecto 01
Contreras Aguilar Fernando

Variables:

productDictionary:

"total_sold"	Ventas totales por producto
"pieces_sold"	Piezas vendidas por producto
"id"	ID del producto
"name"	Nombre del producto
"price"	Precio del producto
"category"	Categoría del producto
"stock"	Stock del producto
"reviews"	Cantidad de calificaciones
"score"	Promedio de calificaciones
"searches"	Búsquedas del producto

months:

- [0] **ID del mes**
- [1] **Nombre del mes**
- [2] **Ventas del mes en pesos**
- [3] **Cantidad de ventas**

total_annual = Total anual de ventas en pesos

pieces_sold = Total anual de piezas vendidas

average_sold = Promedio mensual de piezas vendidas

top_selling = Arreglo de los meses con mejores ventas

top_products = Arreglo de los productos con mas piezas vendidas

worst_selling = Productos menos vendidos

categories_count = Arreglo con el nombre de la categoria y cuantas veces se vendio

Importaciones básicas

```
1  from flask import Flask, jsonify, request
2  from flask_cors import CORS, cross_origin
3  from lifestore_file import lifestore_products, lifestore_sales, lifestore_searches
4  from auth import users
5
6  app = Flask(__name__)
7  app.config["DEBUG"] = True
8  CORS(app, support_credentials=True)
9
10 products = lifestore_products
11 sales = lifestore_sales
12 searches = lifestore_searches
```

Se importan y declaran las listas necesarias con la información de las ventas, productos y búsquedas. Así como los módulos para levantar el servidor backend de la aplicación

```
14 productDictionary = [] #Diccionario de productos
15 productDictionary.append(0) #El primer elemento es cero para coincidir el id del producto con su posición
16 for product in products:
17     newProduct = {
18         "total_sold" : 0 ,
19         "pieces_sold": 0 ,
20         "id" : product[0],
21         "name" : product[1],
22         "price" : product[2],
23         "category" : product[3],
24         "stock" : product[4],
25         "reviews" : [0] ,
26         "score" : 0 ,
27         "searches" : 0
28     }
29     productDictionary.append(newProduct)
30
```

Se define un diccionario de datos para facilitar el acceso a la información de cada producto, resumiendo su información de ventas y búsquedas.

Código

```
43 "Monthly sales"
44 monthly_average = 0
45 months = [1,"Jan", 0, 0], [2,"Feb", 0, 0], [3,"Mar", 0, 0],[4,"Apr", 0, 0], [5,"May", 0, 0].
46
```

Así mismo, se define un arreglo para los 12 meses, esto con fines de mostrar sus nombres. Posteriormente se llenarán con sus respectivas ventas.

```
47 "Product's sales"
48 for sale in sales:
49     productDictionary[sale[1]]["pieces_sold"] += 1 #Sum 1 to the product's sales
50     productDictionary[sale[1]]["total_sold"] += productDictionary[sale[1]]["price"] #Sum the product's pr
51
52     "Get monthly sales"
53     month = sale[3].split('/')[1] #Split the date string and extract the month
54     months[int(month) - 1][2] += productDictionary[sale[1]]["price"] #Get the sale and add it in it's mont
55     months[int(month) - 1][3] += 1 #Get the sale and add it in it's month(-1)
56
57     "Customer's reviews"
58     productDictionary[sale[1]]["reviews"].append(sale[2]) #Array of customer's reviews
59     sale += [0]
60     sale[5] = productDictionary[sale[1]]["name"] #Name of the product
61
```

Ciclo encargado de contabilizar las piezas vendidas de cada producto, así como el total de ingresos que generó cada producto y sus respectivas calificaciones. También se llena el arreglo de meses con sus ventas.

```
62 "Yearly sales"
63 total_annual = 0
64 top_products = [] * len(productDictionary)
65
66 for i in range(1, len(productDictionary)):
67     total_annual += productDictionary[i]["total_sold"]
68
69     "Review average"
70     if ((len(productDictionary[i]["reviews"]) - 1) > 0): #Avoid dividing by zero
71         productDictionary[i]["score"] = sum(productDictionary[i]["reviews"]) / (len(productDictionary[i]
72
73     "Searches"
74     productDictionary[searches[i-1][1]]["searches"] += 1
75
```

Parecido al ciclo anterior, este se encarga de obtener el promedio de calificaciones por producto y el total de sus búsquedas. Así como obtener el total de ventas anuales.

Código: Información individual

```
76     "Average pieces sold"
77     pieces_sold = len(sales)
78     average_sold = pieces_sold / 12
79
80     "Monthly average"
81     monthly_average = total_annual/12
82
83     "Top 5 Selling months"
84     top_selling = []
85     for i in range (0, 5):
86         top_selling.append(months[i]) #Insert first 5 months
87
88     top_selling.sort(reverse=True) #Order descending
89
90     for i in range(5, 12):
91         if(months[i][3] > top_selling[4][3]): #If the last one (the smallest) is smaller than
92             top_selling.pop()
93             top_selling.append(months[i])
94
95     "Top Selling products"
96     sorted_products = []
97     sorted_products = productDictionary[1:len(productDictionary)]
98     sorted_products = sorted(sorted_products, key = lambda k: k['pieces_sold'], reverse=True)
99     top_products = sorted_products[0:51]
100    worst_products = sorted_products[51:len(productDictionary)]
```

Cumpliendo con el objetivo, las siguientes partes del código definen diferentes aspectos de la información mostrada. Tales como:

- Promedio de piezas vendidas.
- Promedio de ventas mensuales.
- Top 5 de los meses con mejores ventas.
- Listas con los productos más vendidos y menos vendidos.
- Mejores y peores calificaciones.

Código: Peticiones HTTP

```
139     @app.route('/reviews', methods=['GET'])
140     @cross_origin(supports_credentials=True)
141     def get_reviews():
142
143
144     return jsonify({
145         'top_reviews': top_reviews,
146         'worst_reviews': worst_reviews,
147         'best_seller': best_seller,
148         'worst_seller': worst_seller,
149         'most_popular': most_popular,
150         'less_popular': less_popular,
151         'categories': categories_count
152     })
153
154     @app.route('/stock', methods=['GET'])
155     @cross_origin(supports_credentials=True)
156     def get_stock():
157
158     return jsonify({
159         'products': productDictionary,
160         'categories': categories_count
161     })
162
```

La última parte del código se encarga de manejar las peticiones HTTP solicitadas del lado del cliente, devolviendo a este la información requerida para cada página.

Solución al problema

Proyecto 01
Contreras Aguilar Fernando

Resumen

Ventas

Anuales \$760,177.00

Mensuales \$63,348.08

Piezas

Anuales 283

Mensuales 23

Categorías

1. Procesadores

104 Piezas vendidas

2. Discos duros

94 Piezas vendidas

3. Tarjetas Madre

49 Piezas vendidas

4. Tarjetas de video

26 Piezas vendidas

5. Audífonos

05 Piezas vendidas

6. Bocinas

02 Piezas vendidas

7. Bocinas

02 Piezas vendidas

8. Memorias USB

01 Piezas vendidas

Productos más vendidos y productos rezagados

Los procesadores y discos duros lideran las ventas con más del 60% de las ventas totales.

Mejores ventas:

1. SSD Kingston A400
2. AMD Ryzen 5 2600
3. Intel Core i3-9100F

Tanto en ventas, comp en reseñas y búsquedas, estos productos se posicionan por encima de todos, siendo estos de las categorías Procesadores y Discos Duros; seguidos por las tarjetas madre.

Estas categorías no presentan escases de inventario, exceptuando el procesador AMD Ryzen 3 3300X quien a su vez es uno de los productos más populares en reseñas y búsquedas, por lo que se puede dar más atención a los productos rezagados. Se espera una falta de inventario para varios de estos productos de al menos 4 - 5 meses, así que el reabastecimiento no es necesario.

Mientras que el SSD Kingston A400 lidera en ventas y popularidad, Klip Xtreme Audífonos Blast presentan un bajo rendimiento, siendo el producto menos vendido, con menos búsquedas y con menos reseñas de todos (0 en total en todas ellas). Es recomendable retirar el producto del mercado y cambiarlo por un producto más popular.

A su vez, la categoría audífonos presenta bajas ventas y varios productos tienen gran stock. Se recomienda reducir el abastecimiento de estos productos para realizar un cambio de estos por productos más populares en el mercado. De igual manera la categoría Pantallas presenta bajas ventas y varios artículos con gran stock.

La categoría con menos ventas, Memorias USB, presenta un desabasto en sus productos siendo únicamente 2 de su tipo. Además de hacer un cambio en el nombre de esta categoría, puesto que es incorrecta, se recomienda introducir más productos de su tipo al mercado.

Resumen

Puesto que los procesadores y discos duros presentan buenas ventas y un stock adecuado. Se recomienda reducir el abastecimiento de estos para dar lugar a la rotación de productos rezagados e introducción de nuevos productos.

Las tarjetas madre pueden seguir con su abastecimiento normal.

Las categorías Bocinas, Audífonos y pantallas presentan bajas ventas y acumulacion de stock, además de poca reacción de los usuarios. Se recomienda disminuir el abastecimiento de estos para dar lugar a la introducción de prudctos mas populares.

Se recomienda eliminar la categoría Memorias USB, introduciendo más productos de la nueva categoría Memorias RAM.

Conclusión

Incluso es más pequeño y aparentemente insignificante análisis puede hacer una gran diferencia a la hora de tomar decisiones, para una empresa, una oferta de trabajo o el realizar una actividad. Tenemos los datos siempre presentes pero a veces no notamos la importancia que tiene estos.

Podría decirse incluso que cualquier científico es un científico de datos, puesto que de ellos la ciencia tiene sentido y de ellos se modela la vida como la conocemos. No hay área donde la ciencia de datos no pueda hacer de las suyas para cambiar el rumbo de su crecimiento.

Proyecto:

https://github.com/fershous/PROYECTO_01_CONTRERAS_AGUilar_FERNANDO