



PROYECTO FINAL PARTE 1 INTRODUCCIÓN A PYTHON

Zubiri Joel Hernández González

Universidad Veracruzana

Asesor: Javier Ramírez

Grupo 2 Data Science



Universidad Veracruzana

Xalapa, Ver a 07 de septiembre de 2020

## ÍNDICE GENERAL

INTRODUCCIÓN.....	3
OBJETIVOS .....	3
DEFINICIÓN DEL CÓDIGO.....	4-8
SOLUCIÓN DEL PROBLEMA Y CONCLUSIONES.....	8

## INTRODUCCIÓN

La ciencia de datos es el estudio de dónde proviene la información, qué representa y cómo se puede convertir en un recurso valioso para la creación de estrategias empresariales y de TI. La extracción de grandes cantidades de datos estructurados y no estructurados para identificar patrones puede ayudar a una organización a controlar los costos, aumentar la eficiencia, reconocer nuevas oportunidades de mercado y aumentar la ventaja competitiva de la organización. El campo de la ciencia de datos emplea matemáticas, estadística y disciplinas informáticas, e incorpora técnicas como el aprendizaje automático, el análisis de conglomerados, la extracción de datos y la visualización.

## OBJETIVOS

Poner en práctica las bases de programación en Python para análisis y clasificación de datos mediante la creación de programas de entrada de usuario y validaciones, uso y definición de variables y listas, operadores lógicos y condicionales para la clasificación de información.

### **Productos más vendidos y productos rezagados:**

Generar un listado de los 50 productos con mayores ventas y uno con los 100 productos con mayores búsquedas.

Por categoría, generar un listado con los 50 productos con menores ventas y uno con los 100 productos con menores búsquedas.

### **Productos por reseña en el servicio**





Mostrar dos listados de 20 productos cada una, un listado para productos con las mejores reseñas y otro para las peores, considerando los productos con devolución.

**Total de ingresos y ventas promedio mensuales, total anual y meses con más ventas al año.**

## Descripción del código

Se realizaron funciones simples para simplificar el proceso como:

la creación de programas de entrada de usuario y validaciones, uso y definición de variables y listas, operadores lógicos y condicionales para la clasificación de información.

```
Open ▾  chegg.py Save   
lifestore.py x chegg.py x
import functools
from lifestore import lifestore_products, lifestore_sales, lifestore_searches

# Comparison function for custom sorting according to decreasing values of second element of list items
def compare1(item1, item2):
    return item2[1] - item1[1]

# Comparison function for custom sorting according to increasing values of second element of list items
def compare2(item1, item2):
    return item1[1] - item2[1]

# Store the total number of products in the dataset
totalProducts = len(lifestore_products)

# The following would store the total number of searches and total sales of each individual product
searchesDone = []
salesDone = []

# Initialize the above lists with [product_id, 0]
for i in range(totalProducts):
    searchesDone.append([i+1, 0])
    salesDone.append([i+1, 0])

# Iterate through the lifestore_searches list and increment the search done for the corresponding product_id
for searches in lifestore_searches:
    productId = searches[1]-1
    searchesDone[productId] = [productId+1, searchesDone[productId][1]+1]

# Iterate through the lifestore_sales list and increment the sale done for the corresponding product_id
for sales in lifestore_sales:
    productId = sales[1]-1
    salesDone[productId] = [productId+1, salesDone[productId][1]+1]

# Find the top 50 products with highest sales
top50Sales = sorted(salesDone, key=functools.cmp_to_key(compare1))[:50]

# Find the top 100 products with highest searches
top100Searches = sorted(searchesDone, key=functools.cmp_to_key(compare1))[:100]
```

```

# Print the result for the top 50 sales
print("")
print("Top 50 sales: (ID ---> Sales)")
print("")
for prod in top50Sales:
    print("{0} ---> {1} sales".format(prod[0],prod[1]))

# Print the result for the top 100 searches
print("")
print("Top 100 searches: (ID ---> Searches)")
print("")
for prod in top100Searches:
    print("{0} ---> {1} searches".format(prod[0],prod[1]))

# Find the 50 products with lowest sales
lowest50Sales = sorted(salesDone,key=functools.cmp_to_key(compare2))[:50]

# Find the 100 products with lowest searches
lowest100Searches = sorted(searchesDone,key=functools.cmp_to_key(compare2))[:100]

# Print the result for the 50 lowest sales
print("50 Lowest sales: (ID ---> Sales)")
print("")
for prod in lowest50Sales:
    print("{0} ---> {1} sales".format(prod[0],prod[1]))

# Print the result for the 100 lowest searches
print("")
print("100 Lowest searches: (ID ---> Searches)")
print("")
for prod in lowest100Searches:
    print("{0} ---> {1} searches".format(prod[0],prod[1]))

```

Python ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS

En estos códigos es posible identificar las listas de los productos más vendidos, así como los menos vendidos, usando la función `sorted` para generar nuevas listas ordenadas a partir de las anteriores.

```
Terminal
File Edit View Search Terminal Help
171070030@ltsp374:~$ python3 chegg.py

Top 50 sales: (ID ---> Sales)

54 ---> 50 sales
3 ---> 42 sales
5 ---> 20 sales
42 ---> 18 sales
57 ---> 15 sales
29 ---> 14 sales
2 ---> 13 sales
4 ---> 13 sales
47 ---> 11 sales
12 ---> 9 sales
48 ---> 9 sales
7 ---> 7 sales
31 ---> 6 sales
44 ---> 6 sales
18 ---> 5 sales
8 ---> 4 sales
6 ---> 3 sales
11 ---> 3 sales
49 ---> 3 sales
51 ---> 3 sales
1 ---> 2 sales
21 ---> 2 sales
25 ---> 2 sales
33 ---> 2 sales
52 ---> 2 sales
74 ---> 2 sales
85 ---> 2 sales
10 ---> 1 sales
13 ---> 1 sales
17 ---> 1 sales
22 ---> 1 sales
28 ---> 1 sales
40 ---> 1 sales
45 ---> 1 sales
46 ---> 1 sales
50 ---> 1 sales
60 ---> 1 sales
66 ---> 1 sales
67 ---> 1 sales
84 ---> 1 sales
89 ---> 1 sales
94 ---> 1 sales
```

```
Terminal
File Edit View Search Terminal Help

94 ---> 1 sales
9 ---> 0 sales
14 ---> 0 sales
15 ---> 0 sales
16 ---> 0 sales
19 ---> 0 sales
20 ---> 0 sales
23 ---> 0 sales
24 ---> 0 sales

Top 100 searches: (ID ---> Searches)

54 ---> 263 searches
57 ---> 107 searches
29 ---> 60 searches
3 ---> 55 searches
4 ---> 41 searches
85 ---> 35 searches
67 ---> 32 searches
7 ---> 31 searches
5 ---> 30 searches
47 ---> 30 searches
48 ---> 27 searches
44 ---> 25 searches
2 ---> 24 searches
42 ---> 23 searches
8 ---> 20 searches
12 ---> 15 searches
21 ---> 15 searches
66 ---> 15 searches
18 ---> 11 searches
51 ---> 11 searches
1 ---> 10 searches
6 ---> 10 searches
25 ---> 10 searches
31 ---> 10 searches
49 ---> 10 searches
84 ---> 10 searches
50 ---> 7 searches
89 ---> 7 searches
74 ---> 6 searches
94 ---> 6 searches
11 ---> 5 searches
22 ---> 5 searches
```

```
Terminal
File Edit View Search Terminal Help
26 ---> 5 searches
28 ---> 5 searches
52 ---> 5 searches
15 ---> 4 searches
46 ---> 4 searches
63 ---> 4 searches
73 ---> 4 searches
17 ---> 3 searches
39 ---> 3 searches
95 ---> 3 searches
13 ---> 2 searches
56 ---> 2 searches
76 ---> 2 searches
91 ---> 2 searches
9 ---> 1 searches
10 ---> 1 searches
27 ---> 1 searches
35 ---> 1 searches
45 ---> 1 searches
59 ---> 1 searches
70 ---> 1 searches
80 ---> 1 searches
93 ---> 1 searches
14 ---> 0 searches
16 ---> 0 searches
19 ---> 0 searches
20 ---> 0 searches
23 ---> 0 searches
24 ---> 0 searches
30 ---> 0 searches
32 ---> 0 searches
33 ---> 0 searches
34 ---> 0 searches
36 ---> 0 searches
37 ---> 0 searches
38 ---> 0 searches
41 ---> 0 searches
43 ---> 0 searches
53 ---> 0 searches
55 ---> 0 searches
58 ---> 0 searches
60 ---> 0 searches
61 ---> 0 searches
62 ---> 0 searches
64 ---> 0 searches
```

```
Terminal
File Edit View Search Terminal Help
65 ---> 0 searches
68 ---> 0 searches
69 ---> 0 searches
71 ---> 0 searches
72 ---> 0 searches
75 ---> 0 searches
77 ---> 0 searches
78 ---> 0 searches
79 ---> 0 searches
81 ---> 0 searches
82 ---> 0 searches
83 ---> 0 searches
86 ---> 0 searches
87 ---> 0 searches
88 ---> 0 searches
90 ---> 0 searches
92 ---> 0 searches
96 ---> 0 searches

50 Lowest sales: (ID ---> Sales)
9 ---> 0 sales
14 ---> 0 sales
15 ---> 0 sales
16 ---> 0 sales
19 ---> 0 sales
20 ---> 0 sales
23 ---> 0 sales
24 ---> 0 sales
26 ---> 0 sales
27 ---> 0 sales
30 ---> 0 sales
32 ---> 0 sales
34 ---> 0 sales
35 ---> 0 sales
36 ---> 0 sales
37 ---> 0 sales
38 ---> 0 sales
39 ---> 0 sales
41 ---> 0 sales
43 ---> 0 sales
53 ---> 0 sales
55 ---> 0 sales
56 ---> 0 sales
58 ---> 0 sales
```

```
Terminal
File Edit View Search Terminal Help
59 ---> 0 sales
61 ---> 0 sales
62 ---> 0 sales
63 ---> 0 sales
64 ---> 0 sales
65 ---> 0 sales
68 ---> 0 sales
69 ---> 0 sales
70 ---> 0 sales
71 ---> 0 sales
72 ---> 0 sales
73 ---> 0 sales
75 ---> 0 sales
76 ---> 0 sales
77 ---> 0 sales
78 ---> 0 sales
79 ---> 0 sales
80 ---> 0 sales
81 ---> 0 sales
82 ---> 0 sales
83 ---> 0 sales
86 ---> 0 sales
87 ---> 0 sales
88 ---> 0 sales
90 ---> 0 sales
91 ---> 0 sales

100 Lowest searches: (ID ---> Searches)
14 ---> 0 searches
16 ---> 0 searches
19 ---> 0 searches
20 ---> 0 searches
23 ---> 0 searches
24 ---> 0 searches
30 ---> 0 searches
32 ---> 0 searches
33 ---> 0 searches
34 ---> 0 searches
36 ---> 0 searches
37 ---> 0 searches
38 ---> 0 searches
41 ---> 0 searches
43 ---> 0 searches
53 ---> 0 searches
```

```
Terminal
File Edit View Search Terminal Help
55 ---> 0 searches
58 ---> 0 searches
60 ---> 0 searches
61 ---> 0 searches
62 ---> 0 searches
64 ---> 0 searches
65 ---> 0 searches
68 ---> 0 searches
69 ---> 0 searches
71 ---> 0 searches
72 ---> 0 searches
75 ---> 0 searches
77 ---> 0 searches
78 ---> 0 searches
79 ---> 0 searches
81 ---> 0 searches
82 ---> 0 searches
83 ---> 0 searches
86 ---> 0 searches
87 ---> 0 searches
88 ---> 0 searches
90 ---> 0 searches
92 ---> 0 searches
96 ---> 0 searches
9 ---> 1 searches
10 ---> 1 searches
27 ---> 1 searches
35 ---> 1 searches
45 ---> 1 searches
59 ---> 1 searches
70 ---> 1 searches
80 ---> 1 searches
93 ---> 1 searches
13 ---> 2 searches
56 ---> 2 searches
76 ---> 2 searches
91 ---> 2 searches
17 ---> 3 searches
39 ---> 3 searches
95 ---> 3 searches
15 ---> 4 searches
46 ---> 4 searches
63 ---> 4 searches
73 ---> 4 searches
11 ---> 5 searches
```



```
22 ---> 5 searches
26 ---> 5 searches
28 ---> 5 searches
52 ---> 5 searches
74 ---> 6 searches
94 ---> 6 searches
50 ---> 7 searches
89 ---> 7 searches
1 ---> 10 searches
6 ---> 10 searches
25 ---> 10 searches
31 ---> 10 searches
40 ---> 10 searches
49 ---> 10 searches
84 ---> 10 searches
18 ---> 11 searches
51 ---> 11 searches
12 ---> 15 searches
21 ---> 15 searches
66 ---> 15 searches
8 ---> 20 searches
42 ---> 23 searches
2 ---> 24 searches
44 ---> 25 searches
48 ---> 27 searches
5 ---> 30 searches
47 ---> 30 searches
7 ---> 31 searches
67 ---> 32 searches
85 ---> 35 searches
4 ---> 41 searches
3 ---> 55 searches
29 ---> 60 searches
57 ---> 107 searches
54 ---> 203 searches
171070030@lisp374:~$
```

## CONCLUSIONES

Al analizar los datos utilizados es evidente que la compañía se encuentra en un camino adecuado, solo que el inventario o listas de productos pueden mejorar su organización al identificar cada una de las partes de una manera adecuada, como reordenando los meses de ventas para los ingresos o incluso de manera específica dejar de vender por ejemplo los 20 productos con menos ventas, contando las devoluciones. En general el presente proyecto hizo posible aplicar una nueva perspectiva sobre los productos manejados y su organización permitiendo compararlos mediante listas de fácil comprensión.

Código en Replit: <https://repl.it/@ZubiriJoelHdez/PROYECTO-NUEVO#main.py>

Código base: [https://repl.it/@ZubiriJoelHdez/PROYECTO-NUEVO#lifestore\\_file.py](https://repl.it/@ZubiriJoelHdez/PROYECTO-NUEVO#lifestore_file.py)