Laboratorio 2 Módulo 3 Bloque 5

20/06/2022

Nombre: _____

Instrucciones

El presente examen se compone de un solo problema con varias partes incrementales. Tendrás un tiempo de 110 minutos para resolverlo, más 10 minutos para entrega en línea. Te recomendamos revisar rápidamente los requisitos de la solución abajo, y luego ir a la sección titulada "Observaciones sobre la implementación" antes de comenzar a resolver el examen.

eCommerce (6 puntos)

Tendrás un lista de productos electrónicos, cada producto estará caracterizado por 7 campos definidos en un diccionario con el siguiente esquema:

Listado 1: Representación de productos

```
producto = {
        'id' : string,
        'precio_max' : float,
        'precio_min' : float,
        'vendedor': string,
        'marca': string,
        'descripcion': string,
        'peso': string
    }
```

A continuación se muestra un ejemplo de uno de los productos del catálogo a procesar:

```
{'id': 'AVphAbip1cnluZ0-9HBE',
 'precio_max': 550.0,
 'precio_min': 550.0,
 'vendedor': 'Tunny LLC',
 'marca': 'ASUS',
 'descripcion': 'MG278Q 27 Widescreen LED Backlit TN Gaming Monitor',
 'peso': '16.9 lb'}
```

Se te dará un código base *l2b5m3_base.py* con funciones para cargar los datos y reportar todos los productos. Un ejemplo de invocación a estas funciones se muestra a continuación:

```
import catalog
data = catalog.read_catalog()
reportar(data)
```

Funciones a implementar

En el código base entregado encontrarás cada una de las definiciones de las funciones a implementar. Deberás completar cada una de ellas de acuerdo con las siguientes especificaciones:

1. [1.0 pt] `top_vendedores(data, N)`: Recibe el catálogo y un valor entero N. Devuelve los N vendedores con más productos (top N). Aquí puedes retornar cualquier estructura que permita representar a cada vendedor junto a la cantidad de productos. (Puedes usar diccionario, listas de tuplas o 2 listas.)

```
print(top_vendedores(data, 5))

[('Bestbuy.com', 2806),
('bhphotovideo.com', 1509),
('Walmart.com', 664),
('Beach Camera', 201),
('AMI Ventures Inc', 63)]
```

2. [1.0 pt] `top_marcas(data, N)`: Recibe el catálogo y un valor entero N. Devuelve las N marcas con más productos. Aquí puedes retornar cualquier estructura que permita representar a cada marca junto a la cantidad de productos. (Puedes usar diccionario, listas de tuplas o 2 listas.)

```
print(top_marcas(data, 5))
[('Sony', 784),
('Samsung', 743),
('Apple', 246),
('Yamaha', 238),
('Pioneer', 175)]
```

3. [1.0 pt] `top_marcas_precio(data, N)`: Recibe el catálogo y un valor entero N. Devuelve las N marcas con el o los productos más caros. Aquí puedes retornar cualquier estructura que permita representar a cada marca junto al precio del producto más caro. (Puedes usar diccionario, listas de tuplas o 2 listas.)

```
print(top_marcas_precio(data, 5))
[('Samsung', 6999.99),
('Sony', 4999.99),
('LG', 4999.99),
('Razer', 4399.99),
('SunBriteTV', 4295.98)]
```

4. [1.0 pt] `top_marcas_vendedor(data, N)`: Recibe el catálogo y un valor entero N. Devuelve las N marcas con más vendedores. Aquí puedes retornar cualquier estructura que permita representar a cada marca junto al número de vendedores. (Puedes usar diccionario, listas de tuplas o 2 listas.)

```
print(top_marcas_vendedor(data, 5)
[('Sony', 98),
('Samsung', 75),
('Pioneer', 41),
('Logitech', 35),
('Seagate', 31)]
```

5. [2.0 pt] `top_vendedores_productos_precio(data, N)`: Recibe el catálogo y un valor entero N. Devuelve los N vendedores con el producto más caro junto a los M productos más caros por vendedor. Aquí deberás retornar un diccionario indexado por nombre del vendedor, y donde cada vendedor tiene asociado una estructura con los productos y precios ordenados por precio.

```
print(top_vendedor_productos_precio(data, 5, 2))
{'Walmart.com':
[('AV_Ic1sLYSSHbkXwqI3O', 6999.99), ('AVqVGUFCv8e3D1O-ldFF', 5199.99)]
'Beach Camera':
[('AVs4UxVHU2_QcyX9P_Gp', 4999.99), ('AVzxqGmivKc47QAVfTIA', 4999.99)],
'World Wide Stereo':
[('AVqkH8TtU2_QcyX9OOrJ', 4999.99), ('AVzxqGmivKc47QAVfTIA', 4999.99)],
'Bestbuy.com':
[('AVpg9lcJ1cnluZO-8h3k', 4295.98), ('AVs4UxVHU2_QcyX9P_Gp', 3999.99)],
'bhphotovideo.com':
[('AVpg9lcJ1cnluZO-8h3k', 4295.0), ('AVphvkgGLJeJML43eVIb', 3499.95)]}
```

Observaciones sobre la implementación

1. No está permitido modificar la firma (nombre y parámetros) de las funciones que debes implementar. Sí está permitido agregar otras funciones.

2. Puedes probar el resultado de cada función al descomentar la línea correspondiente al final del programa base.

Sistema de Evaluación

Cada ítem es evaluado con una escala de 1-5 (1: no presenta solución, 2: presenta esbozo de solución incompleto o con errores graves, 3: solución incompleta o imprecisa, 4: error(es) mínimo(s), 5: excelente) y luego un ponderador (1:0, 2:0,25, 3:0,5, 4:0,75, 5:1,0) es aplicado al puntaje del ítem. El puntaje otorgado a la solución se calcula como la suma de los puntajes parciales obtenidos, aplicándose los ponderadores. La nota final es el puntaje calculado más el punto base.

Solución

Listado 2: Código de la solución

```
1
2  """
3  Se presenta una lista de dicctionarios,
4  donde cada diccionario representa a un producto
5
6  """
7  import matplotlib.pyplot as plt
8  #Leer datos de catálogo
9  def read_catalog() :
10     f = open('ecommerce.csv')
11     datalist = f.readlines()
12     data = []
13     for item in datalist[1:]:
14         l = item.strip().split(',')
15         data.append({'id' : l[0],
16          'precio_max' : float(l[1]),
17          'precio_min' : float(l[2]),
18          'vendedor' : l[3],
19          'marca' : l[4],
20          'descripcion' : l[5],
21          'peso': l[6]})
22     return data
23
24 def ordenar(dict):
25     dic_ordenado = sorted(dict.items(), key = lambda x : x[1], reverse = True)
26     return dic_ordenado
27 #Mostrar los 10 vendedores con más productos
28 def obtener_vendedores_productos(data):
29     # vendedor : cantidad
30     dic_conteo = {}
31     for item in data :
32         vendedor = item['vendedor']
33         if vendedor in dic_conteo :
34             dic_conteo[vendedor] += 1
35         else :
36             dic_conteo[vendedor] = 1
37     return dic_conteo
38
39 def obtener_marcas_productos(data):
40     # vendedor : cantidad
41     dic_conteo = {}
42     for item in data :
43         marca = item['marca']
44         if marca in dic_conteo :
45             dic_conteo[marca] += 1
46         else :
47             dic_conteo[marca] = 1
48     return dic_conteo
49
50 def obtener_marcas_mayor_precio(data):
51     dic_precio = {}
52     for item in data :
53         marca = item['marca']
54         precio = item['precio_max']
55         if marca in dic_precio :
56             if precio > dic_precio[marca]:
57                 dic_precio[marca] = precio
58         else :
59             dic_precio[marca] = precio
60     return dic_precio
61
62
63 def obtener_marcas_vendedores(data):
64     # vendedor : cantidad
65     dic = {}
66     for item in data :
67         marca = item['marca']
68         vendedor = item['vendedor']
```

```
69          if not marca in dic :
70              dic[marca] = []
71          if not vendedor in dic[marca] :
72              dic[marca].append(vendedor)
73      for marca in dic :
74          dic[marca] = len(dic[marca])
75      return dic
76
77
78  def obtener_vendedor_productos_caros(data):
79      dic = {}
80      for item in data :
81          vendedor = item['vendedor']
82          id = item['id']
83          precio = item['precio_max']
84          if not vendedor in dic :
85              dic[vendedor] = {}
86          dic[vendedor][id] = precio
87      return dic
88
89  #1
90  def top_vendedores(data, N):
91      dic = obtener_vendedores_productos(data)
92      return ordenar(dic)[:N]
93  #2
94  def top_marcas(data, N):
95      dic = obtener_marcas_productos(data)
96      return ordenar(dic)[:N]
97  #3
98  def top_marcas_precio(data, N):
99      dic = obtener_marcas_mayor_precio(data)
100     return ordenar(dic)[:N]
101 #4
102 def top_marcas_vendedor(data, N):
103     dic = obtener_marcas_vendedores(data)
104     return ordenar(dic)[:N]
105 #5
106 def top_vendedor_productos_precio(data, N, M):
107     dic = obtener_vendedor_productos_caros(data)
108     dic_precios = {}
109     for vendedor in dic :
110         dic[vendedor] = ordenar(dic[vendedor])
111         #print(dic[vendedor])
112         dic_precios[vendedor] = dic[vendedor][0][1]
113     ordenado_por_precio = ordenar(dic_precios)
114     #print(ordenado_por_precio[:10])
115     result = {}
116     for item in ordenado_por_precio[:N] :
117         result[item[0]] = dic[item[0]][:M]
118
119     return result
120
121
122 def reportar(data):
123     for item in data:
124         print(item)
125
126 #5
127 data = read_catalog()
128 #reportar(data))
129 #1
130 #print(top_vendedores(data, 5))
131 #2
132 #print(top_marcas(data, 5))
133 #3
134 #print(top_marcas_precio(data, 5))
135 #4
136 #print(top_marcas_vendedor(data, 5))
137 #5
138 print(top_vendedor_productos_precio(data, 5, 3))
```