РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра информационных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

Дисциплина: Интеллектуальный анализ данных

Студент: Бармина Ольга Константиновна

Группа: НФИбд-01-19

Москва 2022

Вариант №25

задание:

1. При помощи модуля sqlite3 откройте базу данных Instacart в файле instacart.db.
2. Загрузите таблицы `departments` и `products` в датафреймы Pandas. При помощи запроса SELECT извлеките из таблицы `order_products__train` записи, соответствующие указанным в индивидуальном задании дню недели (поле `order_dow` таблицы `orders`) и коду департамента (поле `department_id` таблицы `products`) и загрузите в датафрейм Pandas. Определите количество строк в полученном датафрейме и определите количество товаров (стоолбец `product_id`) в транзакциях датафрейма.
3. Определите пять наиболее популярных товаров в датафрейме транзакций и определите количество покупок (транзакций) этих товаров.
4. Постройте транзакционную базу данных из полученного датафрейма, используя в качестве идентификатора транзакции столбец `order_id` , а в качестве названий товаров - поле `product_name` из датафрейма для таблицы `products` , соответствующее столбцу `product_id` . Найдите в транзакционной базе данных три транзакции с наибольшим количеством товаров и выведите их на экран.
5. Постройте по транзакционной базе данных бинарную базу данных в формате датафрейма пакета `mlxtend` . По бинарной базе данных определите пять наиболее популярных товаров и определите количество покупок (транзакций) этих товаров.
6. При помощи указанного в индивидуальном задании метода построения популярных наборов предметов постройте популярный набор предметов с минимальной поддержкой не менее 3, имеющий максимальную длину. При отсутствии таких наборов уменьшите поддержку до 2. В случае нехватки вычислительных ресурсов (слишком долгой работы программы) при построении популярных наборов предметов сокращайте число записей в наборе данных (например, делая выборку половины записей набора).

7. Используя пакет `mlxtend` или реализацию на Python, постройте набор ассоциативных правил для полученного популярного наборов предметов. Используйте уровень достоверности (confidence), равный 0.6.
8. Для построенного набора ассоциативных правил вычислите показатель (меру) оценки ассоциативных правил, указанную в индивидуальном задании, и определите ассоциативные правила с наилучшим значением показателя оценки.

Индивидуальный вариант:

1. Алгоритм: FPGrowth
2. День недели (поле order_dow таблицы orders): “3”
3. Код департамента (поле department_id таблицы products): “11”
4. Показатель оценки ассоциативных правил: лифт (lift)

Ввод [1]:

```
import numpy as np
import pandas as pd
import sqlite3

# откроем базу данных
conn = sqlite3.connect('instacart.db')
cursor = conn.cursor()
```

Ввод [2]:

```
# создадим датафреймя из таблиц departments и orders
df_dept = pd.read_sql_query("SELECT * FROM departments", conn)
df_dept = df_dept.set_index('department_id')
df_dept
```

Out[2]:

department_id	department
1	frozen
2	other
3	bakery
4	produce
5	alcohol
6	international
7	beverages
8	pets
9	dry goods pasta
10	bulk
11	personal care
12	meat seafood
13	pantry
14	breakfast
15	canned goods
16	dairy eggs
17	household
18	babies
19	snacks
20	deli
21	missing

Ввод [3]:

```
df_prod = pd.read_sql_query("SELECT * FROM products", conn)
df_prod
```

Out[3]:

	product_id	product_name	aisle_id	department_id
0	1	Chocolate Sandwich Cookies	61	19
1	2	All-Seasons Salt	104	13
2	3	Robust Golden Unsweetened Oolong Tea	94	7
3	4	Smart Ones Classic Favorites Mini Rigatoni Wit...	38	1
4	5	Green Chile Anytime Sauce	5	13
...	...	...	...	...
49683	49684	Vodka, Triple Distilled, Twist of Vanilla	124	5
49684	49685	En Croute Roast Hazelnut Cranberry	42	1
49685	49686	Artisan Baguette	112	3
49686	49687	Smartblend Healthy Metabolism Dry Cat Food	41	8
49687	49688	Fresh Foaming Cleanser	73	11

49688 rows × 4 columns

Ввод [4]:

```
# select p.product_name as prod

# посмотрим таблицу из записей, удовлетворяющих условиям варианта
for row in cursor.execute("""
    select opp.order_id, opp.product_id, opp.add_to_cart_order, opp.reordered
    from order_products__train opp, products p, orders ord
    where p.product_id = opp.product_id
    AND ord.order_id = opp.order_id
    AND ord.order_dow = '3'
    AND p.department_id = '11'
    """):
    print(row)
```

```
('1472290', '22', '1', '0')
('257077', '146', '2', '1')
('372892', '146', '2', '0')
('237826', '203', '23', '1')
('218041', '251', '5', '0')
('1263923', '280', '3', '1')
('1516183', '280', '1', '1')
('2132926', '280', '3', '0')
('2293421', '280', '3', '0')
('231694', '280', '4', '0')
('2634727', '280', '1', '1')
('2861288', '280', '2', '1')
('2964680', '280', '2', '1')
('3046057', '280', '14', '0')
('750908', '280', '3', '1')
('850313', '280', '25', '0')
('2082667', '308', '14', '1')
('1310415', '351', '12', '0')
('1445427', '351', '2', '0')
```

Ввод [5]:

```
# запишем эту таблицу в датафрейм
df_opp = pd.read_sql_query("""
                        select opp.order_id, opp.product_id, p.product_name, opp.add_to_car
                        from order_products__train opp, products p, orders ord
                        where p.product_id = opp.product_id
                        AND ord.order_id = opp.order_id
                        AND ord.order_dow = '3'
                        AND p.department_id = '11'""", conn)
df_opp
# получаем, что нашим условия удовлетворяют 2556 записей
```

Out[5]:

	order_id	product_id	product_name	add_to_cart_order	reordered
0	1472290	22	Fresh Breath Oral Rinse Mild Mint	1	0
1	257077	146	Anti Diarrheal Caplets	2	1
2	372892	146	Anti Diarrheal Caplets	2	0
3	237826	203	Rescue Remedy, Spray	23	1
4	218041	251	Extra Strength Melatonin Adult Gummies	5	0
...	...	...	...	...	...
2551	1793846	49433	3D White Whitestrips Luxe Glamorous White	20	0
2552	2565422	49501	Pro Health Antigingivitis And Sensitive Teeth ...	21	1
2553	2823643	49572	Honey & Lemon Menthol Cough Suppressants	13	0
2554	1476570	49649	Refreshing Remover Cleansing Towelettes	5	1
2555	2198380	49688	Fresh Foaming Cleanser	10	0

2556 rows × 5 columns

Ввод [70]:

```
# посмотрим количество уникальных продуктов
np.size(df_opp['product_name'].unique())
```

Out[70]:

```
1421
```

Ввод [6]:

```
# посмотрим 5 самых часто заказываемых товаров
df5 = df_opp.groupby('product_name').count().nlargest(5, 'order_id')['order_id']
df5
```

Out[6]:

```
product_name
Lavender Hand Soap                                                  36
Cotton Swabs                                                        32
Fluoride-Free Antiplaque & Whitening Peppermint Toothpaste          18
Lemon Verbena Hand Soap                                             18
Natural Anticavity Silly Strawberry Fluoride Toothpaste for Children    15
Name: order_id, dtype: int64
```

Ввод [12]:

```
# для построения транзакционной БД объеденим таблицы products и order_products__train, оста
# только 2 стобца - номер заказа и название продукта
df_tmp = pd.merge(df_prod, df_opp, on='product_id')[['order_id','product_name_x']].copy()
df_tmp.columns = ['order_id', 'product_name']
df_tmp
```

Out[12]:

	order_id	product_name
0	1472290	Fresh Breath Oral Rinse Mild Mint
1	257077	Anti Diarrheal Caplets
2	372892	Anti Diarrheal Caplets
3	237826	Rescue Remedy, Spray
4	218041	Extra Strength Melatonin Adult Gummies
...	...	...
2551	1793846	3D White Whitestrips Luxe Glamorous White
2552	2565422	Pro Health Antigingivitis And Sensitive Teeth ...
2553	2823643	Honey & Lemon Menthol Cough Suppressants
2554	1476570	Refreshing Remover Cleansing Towelettes
2555	2198380	Fresh Foaming Cleanser

2556 rows × 2 columns

Ввод [13]:

```
# преобразуем полученную таблицу в словарь, а затем пройдем по всем элементам словаря и зан
df_new = df_tmp.groupby('order_id')['product_name'].apply(list).to_dict().items()
df = []
for row in df_new:
    print(row)
    df.append(row)
```

```
al Clean Dry Spray Antiperspirant Deodorant ', 'Deep Moisture Pump Body Was
h', 'Oatmeal & Shea Butter Body Lotion'])
('1009556', ['FreshBurst Antiseptic Mouthwash'])
('100962', ['Mango Peach Omega Swirl Omage-3 Fish Oil Supplement'])
('1011502', ['Purifying Tea Tree Body Wash'])
('1013793', ['Age Defying Daily Facial Moisturizer'])
('1014185', ['Natural Herb Cough Drops'])
('1021451', ['French Lavender Body Lotion', 'Total Moisture Aloe Fresh Hyd
rating Lotion', 'Ultimate Healing Aloe Skin Therapy Lotion'])
('1021481', ['Gotu Kola Stem Cell + 1% CGF Day Cream'])
('1021685', ['Truly Radiant Whitening & Enamel Strengthening Fresh Mint To
othpaste'])
('1024072', ['Grapeseed Natural Skin Care Oil'])
('1026064', ['Hand Soap Lavender & Coconut', 'Tea Tree Oil Dental Floss',
"Vanilla Al'mondo Protein Shake"])
('1027190', ['Spring Water Antibacterial Liquid Hand Soap with Moisturize
r'])
('1029714', ['Original Vanilla Nutrition Shake', 'Strawberry Nutrition Sha
ke'])
('1033002', ['Pure Castile Peppermint Soap'])
```

Ввод [14]:

```
# найдем 3 транзакции с наибольшим количеством товаров
# отсортируем список по убыванию количества элементов во втором элементе подсписков
# выведем 3 первых элемента
df.sort(key = lambda x:np.size(x[1]), reverse=True)
df[0], df[1], df[2]
```

Out[14]:

```
(('3108267',
  ['Cleansing Towelettes Night Calming Makeup Remover',
   'Plus Soft Toothbrush',
   'Pain Reliever and Fever Reducer Tablets',
   'Total Clean Mint Toothpaste',
   'Loofah Sponges',
   'Neosporin 24 Hour Infection Protection First Aid Antibiotic Ointment',
   'Cotton Swabs',
   'Serenity Bubble Bath',
   'Deep Cleansing Pore Strips']),
 ('1300033',
  ['Wicked Fresh! Cool Peppermint Toothpaste',
   'Natural Anticavity Silly Strawberry Fluoride Toothpaste for Children',
   'Coconut Milk Nourishing Shampoo',
   'Pro-Health Stages Kids Toothbrush With Minnie Mouse',
   'Disney Cars Soft Toothbrush Pro Health Stages 5-7 Yrs',
   'Classic Cherry Lipbalm',
   'Calming Lavender Body Wash',
   'Classic Mouthwash Original Mint Flavor']),
 ('2194307',
  ['Plus Antibiotic Brand Adhesive Bandages',
   'Hurt-Free Antiseptic Wash',
   'Tough Strips',
   'Absolute Waterproof Tape',
   'Nexcare Waterproof Assorted Bandages',
   'Hydrogen Peroxide',
   'Sheer Extra Large All One Size 1 3/4\\" X 4\\" Value']))
```

Ввод [19]:

```
# для преобразования в бинарную БД избавимся от order_id
df_products = [x[1] for x in df]
df_products
```

Out[19]:

```
[['Cleansing Towelettes Night Calming Makeup Remover',
  'Plus Soft Toothbrush',
  'Pain Reliever and Fever Reducer Tablets',
  'Total Clean Mint Toothpaste',
  'Loofah Sponges',
  'Neosporin 24 Hour Infection Protection First Aid Antibiotic Ointment',
  'Cotton Swabs',
  'Serenity Bubble Bath',
  'Deep Cleansing Pore Strips'],
 ['Wicked Fresh! Cool Peppermint Toothpaste',
  'Natural Anticavity Silly Strawberry Fluoride Toothpaste for Children',
  'Coconut Milk Nourishing Shampoo',
  'Pro-Health Stages Kids Toothbrush With Minnie Mouse',
  'Disney Cars Soft Toothbrush Pro Health Stages 5-7 Yrs',
  'Classic Cherry Lipbalm',
  'Calming Lavender Body Wash',
  'Classic Mouthwash Original Mint Flavor'],
 ['Plus Antibiotic Brand Adhesive Bandages',
```

Ввод [20]:

```
# построим бинарную БД по транзакционной БД
from mlxtend.preprocessing import TransactionEncoder

te = TransactionEncoder()
dataset_bin = te.fit(df_products).transform(df_products)
df_bin = pd.DataFrame(dataset_bin, columns=te.columns_)
df_bin
```

Out[20]:

	1 Mg Melatonin Sublingual Orange Tablets	1 Razor Handle and 2 Freesia Scented Razor Refills Premium BladeRazor System	1% Hydrocortisone Anti-Itch Cream, Tube Anti-Itch	1,000 Mg Vitamin C Super Orange	1,000 mg Vitamin C Lemon-Lime Flavored Fizzy Drink Mix - 30 PK	100% Cotton Rounds	100% Cotton Swabs	100% Pure Sensitive Skin Care Grape Seed Oil	P V
0	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	
1782	False	False	False	False	False	False	False	False	
1783	False	False	False	False	False	False	False	False	
1784	False	False	False	False	False	False	False	False	
1785	False	False	False	False	False	False	False	False	
1786	False	False	False	False	False	False	False	False	

1787 rows × 1421 columns

Ввод [30]:

```
# найдем 5 самых популярных товаров и количество их заказов
df_bin.sum(axis=0).nlargest(5)
```

Out[30]:

```
Lavender Hand Soap                                                    36
Cotton Swabs                                                          32
Fluoride-Free Antiplaque & Whitening Peppermint Toothpaste            18
Lemon Verbena Hand Soap                                               18
Natural Anticavity Silly Strawberry Fluoride Toothpaste for Children  15
dtype: int64
```

Ввод [65]:

```
# построим минимальный набор предметов с поддержкой не менее 0.001
# (если брать больше, то списки предметов получаются только одноэлементные, что мешает выпо
from mlxtend.frequent_patterns import fpgrowth

itemsets_fpg = fpgrowth(df_bin, min_support=0.001, use_colnames=True)
itemsets_fpg
```

Out[65]:

	support	itemsets
0	0.017907	(Cotton Swabs)
1	0.003917	(Cleansing Towelettes Night Calming Makeup Rem...
2	0.002798	(Pain Reliever and Fever Reducer Tablets)
3	0.001679	(Deep Cleansing Pore Strips)
4	0.001679	(Serenity Bubble Bath)
...	...	...
475	0.001119	(Cotton Rounds, Coconut Milk Nourishing Shampoo)
476	0.001119	(Lavender Hand Soap, Lemon Verbena Hand Soap)
477	0.001119	(Lavender Hand Soap, Olive Oil & Aloe Vera Han...
478	0.001119	(Spearmint + Lemongrass Hand Soap, Hand Soap L...
479	0.001679	(All One Hemp Lavender Castile Soap Bar, All O...

480 rows × 2 columns

Ввод [66]:

```
# построим набор ассоциативных правил
from mlxtend.frequent_patterns import association_rules

rules = association_rules(itemsets_fpg, metric="confidence", min_threshold=0.6)
rules
```

Out[66]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lever
0	(Moroccan Argan Oil + Argan Stem Cell Triple M...	(Hair Shampoos)	0.001679	0.002798	0.001119	0.666667	238.266667	0.001
1	(Meyer Lemon Everyone Hand Soap)	(Children's Outrageous Orange Mango Natural Fl...	0.001679	0.002798	0.001119	0.666667	238.266667	0.001
2	(Active Naturals Positively Nourishing Body Wa...	(Soothing Aloe Vera Moisturizing Body Wash)	0.001119	0.002798	0.001119	1.000000	357.400000	0.001
3	(Spearmint + Lemongrass Hand Soap)	(Hand Soap Lavender & Coconut)	0.001119	0.004477	0.001119	1.000000	223.375000	0.001
4	(All One Hemp Peppermint Castile Soap Bar)	(All One Hemp Lavender Castile Soap Bar)	0.002238	0.004477	0.001679	0.750000	167.531250	0.001

Ввод [68]:

```
# найдем 2 правила с лучшим показателем lift
rules.sort_values(['lift'], ascending=False).head(2)
```

Out[68]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	levera
2	(Active Naturals Positively Nourishing Body Wa...	(Soothing Aloe Vera Moisturizing Body Wash)	0.001119	0.002798	0.001119	1.000000	357.400000	0.0011
0	(Moroccan Argan Oil + Argan Stem Cell Triple M...	(Hair Shampoos)	0.001679	0.002798	0.001119	0.666667	238.266667	0.0011

Ввод [ ]:

```

```