РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра информационных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

Дисциплина: Интеллектуальный анализ данных

Студент: Ким Реачна

Группа: НПИбд-01-20

Москва 2023

Вариант № 14

Для закрепленного за Вами варианта лабораторной работы:

1. При помощи модуля sqlite3 откройте базу данных Instacart в файле instacart.db.

In [1]:

```
import numpy as np
import pandas as pd
import sqlite3
from mlxtend.preprocessing import TransactionEncoder
import itertools
import warnings

warnings.filterwarnings("ignore")
conn = sqlite3.connect('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, количество транзакций (покупок) и определите количество товаров (столбец product_id) в транзакциях датафрейма

In [2]:

```
data1 = pd.read_sql_query("SELECT * FROM departments", conn)
data2 = pd.read_sql_query("SELECT * FROM products", conn)
```

In [3]:

```
data1.head()
```

Out[3]:

	department_id	department
0	1	frozen
1	2	other
2	3	bakery
3	4	produce
4	5	alcohol

In [4]:

```
data2.head()
```

Out[4]:

	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

In [5]:

```
data = pd.read_sql_query(
    """
SELECT
  opt.order_id,
  opt.product_id,
  add_to_cart_order,
  reordered,
  product_name,
  order_hour_of_day
FROM
  order_products__train as opt,
  orders as ord,
  products as pr
WHERE
  ord.order_id = opt.order_id
  AND pr.product_id = opt.product_id
  AND ord.order_dow = 4
  AND pr.department_id = 5

  """, conn)
```

In [6]:

```
data.head()
```

Out[6]:

	order_id	product_id	add_to_cart_order	reordered	product_name	order_hour_of_day
0	877974	1808	1	0	Champagne	11
1	1859940	15511	1	1	Draft Sake	17
2	3409264	2120	3	1	Sauvignon Blanc	08
3	1881604	10607	6	1	Prosecco	12
4	1881604	29509	2	1	80 Vodka Holiday Edition	12

In [7]:

```
len(data)
```

Out[7]:

958

In [8]:

```
data.order_id.unique().shape
```

Out[8]:

```
(455,)
```

In [9]:

```
len(data['product_id'].unique())
```

Out[9]:

```
314
```

3. Выполните к датафрейму запрос, указанный в индивидуальном задании.

In [10]:

```
hourly_order_counts = data.groupby('order_hour_of_day')['order_id'].nunique()
hour_with_most_orders = hourly_order_counts.idxmax()

hour_with_most_orders
```

Out[10]:

```
'13'
```

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 транзакцию с наибольшим количеством товаров и выведите ее на экран.

In [11]:

```
dataset = data.groupby('order_id')['product_name'].apply(list).to_dict()
dict(itertools.islice(dataset.items(), 10))
data[['order_id', 'product_name']].groupby('order_id')['product_name'].count().sort_
```

Out[11]:

```
order_id
2253479    13
762757     11
1969586    10
1463512    10
2529473    10
           ..
2764641     1
2760266     1
1423906     1
2750744     1
997158      1
Name: product_name, Length: 455, dtype: int64
```

In [12]:

```
for key, group in data[['order_id', 'product_name']].groupby(['order_id']):
    if key == '2253479':
        print('**', key, '**')
        print(group)
        print('-'*9)
```

```
** 2253479 **
    order_id                     product_name
942  2253479                 Westfalia Red Ale
943  2253479       Little Sumpin' Sumpin' Ale
944  2253479                     Belgium Beer
945  2253479            Mighty Dry Hard Cider
946  2253479          Belgian White Wheat Ale
947  2253479     Crisp Hard Cider Crisp Apple
948  2253479               Cabernet Sauvignon
949  2253479     Scrimshaw Pilsner Style Beer
950  2253479                           Merlot
951  2253479          Day Time Fractional IPA
952  2253479           90 Minute Imperial Ipa
953  2253479  Red Wine, Dark, California, 2013
954  2253479                     Villager Ipa
---------
```

5. Постройте по транзакционной базе данных бинарную базу данных в формате датафрейма пакета mlxtend. По бинарной базе данных определите три наиболее популярных товара и определите количество покупок (транзакций) этих товаров.

In [13]:

```
te = TransactionEncoder()
dataset_bin = te.fit([i for i in dataset.values()]).transform([i for i in dataset.va
df = pd.DataFrame(dataset_bin, columns=te.columns_, index=[i for i in dataset.keys()
df.head()
```

Out[13]:

	12 Oz Beer	12 Oz Lager	12 Year Old Single Malt Scotch Speyside	1664	312 Urban Wheat	312 Urban Wheat Ale	46 / 94 Proof Bourbon Kentucky Whiskey	60 Minute IPA	80 Vodka Holiday Edition	80
1007120	True	True	False	False	False	False	False	False	False	Fals
1007997	False	False	False	False	False	False	False	False	False	Fals
1009684	False	False	False	False	False	False	False	False	False	Fals
1009730	False	False	False	False	False	False	False	False	False	Fals
1014150	False	False	False	False	False	False	False	False	False	Fals

5 rows × 314 columns

In [14]:

```
df1 = []
for i in df.columns:
    df1.append((df[i]==True).sum())
pd.Series(df1,index=df.columns).sort_values(ascending=False)[:3]
```

Out[14]:

```
Beer                  45
Cabernet Sauvignon    43
Sauvignon Blanc       38
dtype: int64
```

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).

In [19]:

```
from mlxtend.frequent_patterns import fpmax

itemsets = fpmax(df, min_support=2/df.shape[0],use_colnames=True)
itemsets
```

Out[19]:

	support	itemsets
0	0.017582	(Prosecco)
1	0.008791	(India Pale Ale Racer 5)
2	0.006593	(Fresh Squeezed IPA)
3	0.006593	(12 Oz Lager)
4	0.006593	(Ksa Ko?Lsch Style Ale)
...	...	...
298	0.004396	(Scrimshaw Pilsner Style Beer, Little Sumpin' ...
299	0.004396	(Scrimshaw Pilsner Style Beer, Crisp Hard Cide...
300	0.004396	(Extra Beer Bottles, Belgian Style Wheat Ale)
301	0.004396	(312 Urban Wheat, Belgian Style Wheat Ale)
302	0.004396	(Ale, Amber, Beer)

303 rows × 2 columns

7. Используя пакет mlxtend или реализацию на Python, постройте набор ассоциативных правил для полученного популярного наборов предметов. Используйте уровень достоверности (confidence), равный 0.65.

In [20]:

```
from mlxtend.frequent_patterns import association_rules

data = association_rules(itemsets, metric="confidence", min_threshold=0.65)
data.head(10)
```

Out[20]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
0	(Fresh Squeezed IPA)	(Sauvignon Blanc)	0.006593	0.083516	0.004396	0.666667	7.982456
1	(Cabernet Sauvignon, India Pale Ale)	(Pinot Noir Wine)	0.006593	0.021978	0.004396	0.666667	30.333333
2	(Cabernet Sauvignon, Pinot Noir Wine)	(India Pale Ale)	0.006593	0.068132	0.004396	0.666667	9.784946
3	(India Pale Ale, Pinot Noir Wine)	(Cabernet Sauvignon)	0.006593	0.094505	0.004396	0.666667	7.054264
4	(Belgian White Wheat Ale, India Pale Ale)	(Beer)	0.004396	0.098901	0.004396	1.000000	10.111111
5	(Bitters Liqueur)	(Ale, India Pale, Brew Free! Or Die IPA)	0.006593	0.010989	0.004396	0.666667	60.666667
6	(Cabernet Sauvignon, India Pale Ale)	(Little Sumpin' Sumpin' Ale)	0.006593	0.021978	0.004396	0.666667	30.333333
7	(Little Sumpin' Sumpin' Ale, India Pale Ale)	(Cabernet Sauvignon)	0.006593	0.094505	0.004396	0.666667	7.054264
8	(Little Sumpin' Sumpin' Ale, India Pale Ale)	(Beer)	0.006593	0.098901	0.004396	0.666667	6.740741
9	(Little Sumpin' Sumpin' Ale, Beer)	(India Pale Ale)	0.004396	0.068132	0.004396	1.000000	14.677419

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.

In [22]:

```
data.sort_values('leverage', ascending=False)[['antecedents','consequents', 'leverag
```

Out[22]:

	antecedents	consequents	leverage
429	(Cabernet Sauvignon Sonoma County)	(Cabernet Sauvignon)	0.005970
295	(Variety Pack Hard Cider, Little Sumpin' Sumpi...	(Cabernet Sauvignon, Premium Belgian Lager, Pi...	0.004376
159	(Variety Pack Hard Cider, Little Sumpin' Sumpi...	(Premium Belgian Lager, Pinot Noir Wine)	0.004376
157	(Variety Pack Hard Cider, Premium Belgian Lager)	(Little Sumpin' Sumpin' Ale, Pinot Noir Wine)	0.004376
303	(Cabernet Sauvignon, India Pale Ale, Pinot Noi...	(Little Sumpin' Sumpin' Ale, Premium Belgian L...	0.004376
...	...	...	...
7	(Little Sumpin' Sumpin' Ale, India Pale Ale)	(Cabernet Sauvignon)	0.003772
3	(India Pale Ale, Pinot Noir Wine)	(Cabernet Sauvignon)	0.003772
37	(Belgian White Beer)	(Beer)	0.003744
8	(Little Sumpin' Sumpin' Ale, India Pale Ale)	(Beer)	0.003744
436	(Ale, Amber)	(Beer)	0.003744

437 rows × 3 columns