

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Машинное обучение»
Тема: Частотный анализ

Студент гр. 6304

Иванов Д. В.

Преподаватель

Жангиров Т. Р.

Санкт-Петербург

2020

Цель работы

Ознакомиться с методами частотного анализа из библиотеки

MLxtend

Загрузка данных

1. Скачан и загружен датасет в датафрейм.

```
import pandas as pd
import numpy as np
all_data = pd.read_csv('dataset_group.csv', header=None)
```

2. Получен список всех id покупателей, которые есть в файле: 1139.

```
unique_id = list(set(all_data[1]))
print(len(unique_id))
```

3. Получен список всех товаров, которые есть в файле: 38.

```
items = all_data[2].unique()
print(items.size)
```

4. Сформирован датасет, подходящий для частотного анализа.

```
dataset = [[elem for elem in all_data[all_data[1] == id][2] if elem
in items] for id in unique_id]
```

Подготовка данных

1. Датасет закодирован в виде матрицы с использованием TransactionEncoder:

```
te = TransactionEncoder()
te_ary = te.fit_transform(dataset)
df = pd.DataFrame(te_ary, columns=te.columns_)
```

2. Результат кодирования (рис. 1-2)

	all- purpose	aluminum foil	bagels	beef	butter	cereals	cheeses
0	True	True	False	True	True	False	False
1	False	True	False	False	False	True	True
2	False	False	True	False	False	True	True
3	True	False	False	False	False	True	False
4	True	False	False	False	False	False	False
...
1134	True	False	False	True	False	True	True
1135	False	False	False	False	False	True	True
1136	False	False	True	True	False	False	False
1137	True	False	False	True	False	False	True
1138	False	False	False	False	False	False	False

	coffee/tea	dinner rolls	dishwashing liquid/detergent	...	shampoo
0	False	True	False	...	True
1	False	False	True	...	True
2	False	True	False	...	True
3	False	False	False	...	False
4	False	True	False	...	False
...
1134	True	True	True	...	True
1135	True	True	True	...	False
1136	False	True	True	...	True
1137	False	False	False	...	False
1138	False	False	False	...	True

	soap	soda	spaghetti sauce	sugar	toilet paper	tortillas	\
0	True	True	False	False	False	False	
1	False	False	False	False	True	True	
2	True	True	True	False	True	False	
3	False	True	False	False	True	False	
4	False	True	True	False	True	True	
...	
1134	True	False	False	True	False	False	
1135	True	False	True	False	False	False	
1136	True	False	False	True	False	True	
1137	True	True	True	True	True	False	
1138	False	True	False	False	False	False	

	vegetables	waffles	yogurt
0	True	False	True
1	True	True	True
2	True	False	False
3	False	False	False
4	True	True	True
...
1134	False	False	False
1135	True	False	False
1136	True	False	True
1137	True	True	True
1138	True	False	False

Рис. 1-2 — dataframe, полученный после кодирования.

После кодирования датафрейм представляет из себя матрицу: строки – id покупателей, столбцы – товары, а их пересечение (т.е. ячейки матрицы) указывает на отсутствие или наличие факта покупки товара покупателем (True/False).

Ассоциативный анализ с использованием алгоритма Apriori

1. К полученному датафрейму применен алгоритм apriori с минимальным уровнем поддержки 0.3. Результат – товары или наборы товаров, которые встречаются не реже, чем в 0.3 наборах товаров.

support	itemsets	length	support	itemsets	length
0,37	(all- purpose)	1	0,37	(sandwich bags)	1
0,38	(aluminum foil)	1	0,35	(sandwich loaves)	1
0,39	(bagels)	1	0,37	(shampoo)	1
0,37	(beef)	1	0,38	(soap)	1
0,37	(butter)	1	0,39	(soda)	1
0,40	(cereals)	1	0,37	(spaghetti sauce)	1
0,39	(cheeses)	1	0,36	(sugar)	1
0,38	(coffee/tea)	1	0,38	(toilet paper)	1
0,39	(dinner rolls)	1	0,37	(tortillas)	1
0,39	(dishwashing liquid/detergent)	1	0,74	(vegetables)	1
0,39	(eggs)	1	0,39	(waffles)	1
0,35	(flour)	1	0,38	(yogurt)	1
0,37	(fruits)	1	0,31	(vegetables, aluminum foil)	2
0,35	(hand soap)	1	0,30	(vegetables, bagels)	2
0,40	(ice cream)	1	0,31	(cereals, vegetables)	2
0,38	(individual meals)	1	0,31	(cheeses, vegetables)	2
0,38	(juice)	1	0,31	(vegetables, dinner rolls)	2
0,37	(ketchup)	1	0,31	(vegetables, dishwashing liquid/detergent)	2
0,38	(laundry detergent)	1	0,33	(vegetables, eggs)	2
0,40	(lunch meat)	1	0,30	(ice cream, vegetables)	2
0,38	(milk)	1	0,31	(laundry detergent, vegetables)	2
0,38	(mixes)	1	0,31	(lunch meat, vegetables)	2
0,36	(paper towels)	1	0,33	(poultry, vegetables)	2
0,37	(pasta)	1	0,31	(soda, vegetables)	2
0,36	(pork)	1	0,32	(waffles, vegetables)	2
0,42	(poultry)	1	0,32	(yogurt, vegetables)	2

2. К датафрейму применен алгоритм *apriori* с тем же уровнем поддержки, но с ограничением максимального размер набора равным единице.

Результат – товары, встречающиеся не менее, чем в 0.3 наборах.

support	itemsets	length	support	itemsets	length
0,37	(all- purpose)	1	0,40	(lunch meat)	1
0,38	(aluminum foil)	1	0,38	(milk)	1
0,39	(bagels)	1	0,38	(mixes)	1
0,37	(beef)	1	0,36	(paper towels)	1
0,37	(butter)	1	0,37	(pasta)	1
0,40	(cereals)	1	0,36	(pork)	1
0,39	(cheeses)	1	0,42	(poultry)	1
0,38	(coffee/tea)	1	0,37	(sandwich bags)	1
0,39	(dinner rolls)	1	0,35	(sandwich loaves)	1
0,39	(dishwashing liquid/detergent)	1	0,37	(shampoo)	1
0,39	(eggs)	1	0,38	(soap)	1
0,35	(flour)	1	0,39	(soda)	1
0,37	(fruits)	1	0,37	(spaghetti sauce)	1
0,35	(hand soap)	1	0,36	(sugar)	1
0,40	(ice cream)	1	0,38	(toilet paper)	1
0,38	(individual meals)	1	0,37	(tortillas)	1
0,38	(juice)	1	0,74	(vegetables)	1
0,37	(ketchup)	1	0,39	(waffles)	1
0,38	(laundry detergent)	1	0,38	(yogurt)	1

3. К датафрейму применен алгоритм *apriori* и выведены только те наборы, которые имеют размер 2, а также количество таких наборов: 14.

support	itemsets	length
0,31	(vegetables, aluminum foil)	2
0,30	(vegetables, bagels)	2
0,31	(cereals, vegetables)	2
0,31	(cheeses, vegetables)	2
0,31	(vegetables, dinner rolls)	2
0,31	(vegetables, dishwashing liquid/detergent)	2
0,33	(vegetables, eggs)	2

0,30	(ice cream, vegetables)	2
0,31	(laundry detergent, vegetables)	2
0,31	(lunch meat, vegetables)	2
0,33	(poultry, vegetables)	2
0,31	(soda, vegetables)	2
0,32	(waffles, vegetables)	2
0,32	(yogurt, vegetables)	2

4. Построена зависимость количества наборов от уровня поддержки.

Зависимость количества наборов от уровня поддержки

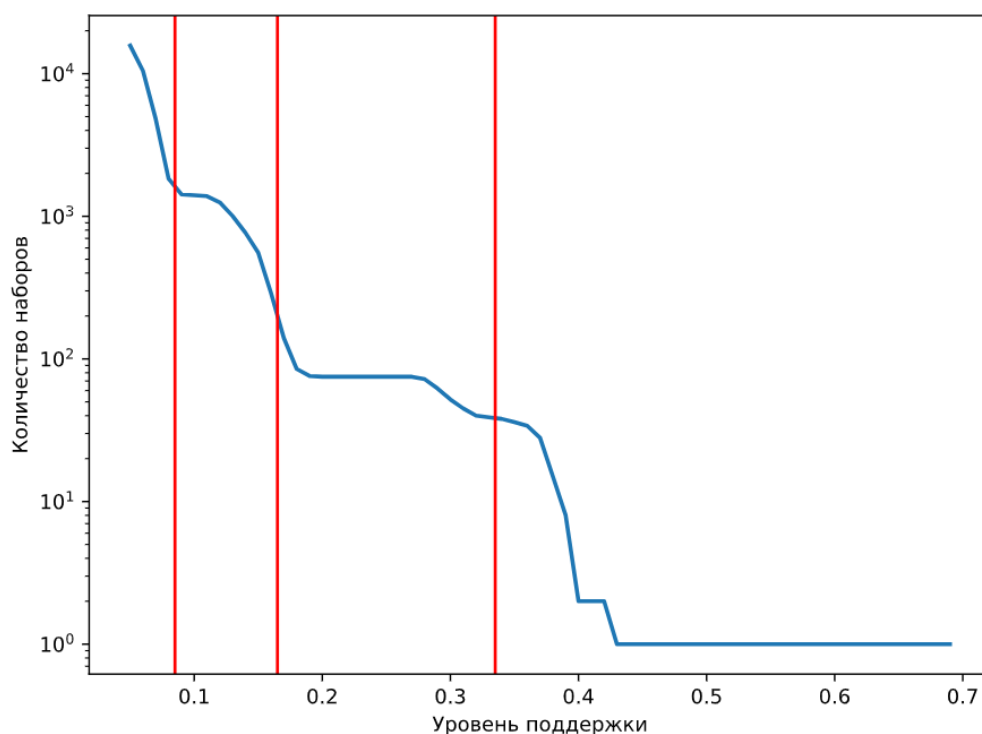


Рис. 3 — Зависимость количества наборов от уровня поддержки.

5. Создан датасет только из элементов, которые попадают в наборы размером 1 при уровне поддержки 0.38. Датасет приведен к формату, подходящему для обработки.

```
results = apriori(df, min_support=0.38, use_colnames=True, max_len=1)
new_items = [ list(elem)[0] for elem in results['itemsets']]
new_dataset = [[elem for elem in all_data[all_data[1] == id][2] if
elem in new_items] for id in unique_id]
te = TransactionEncoder()
te_ary = te.fit_transform(new_dataset)
df_new = pd.DataFrame(te_ary, columns=te.columns_)
```

6. Проведен анализ для уровня поддержки 0.3 нового датасета. Отличие от исходного датасета: в новом датасете присутствуют наборы длины 1, только с минимальным уровнем поддержки 0.38.

support	itemsets	length	support	itemsets	length
0,38	(aluminum foil)	1	0,39	(dishwashing liquid/detergent)	1
0,39	(bagels)	1	0,31	(vegetables, aluminum foil)	2
0,40	(cereals)	1	0,30	(vegetables, bagels)	2
0,39	(cheeses)	1	0,31	(cereals, vegetables)	2
0,39	(dinner rolls)	1	0,31	(cheeses, vegetables)	2
0,38	(yogurt)	1	0,31	(vegetables, dinner rolls)	2
0,39	(eggs)	1	0,31	(dishwashing liquid/detergent, vegetables)	2
0,40	(ice cream)	1	0,33	(vegetables, eggs)	2
0,40	(lunch meat)	1	0,30	(ice cream, vegetables)	2
0,38	(milk)	1	0,31	(lunch meat, vegetables)	2
0,42	(poultry)	1	0,33	(poultry, vegetables)	2
0,39	(soda)	1	0,31	(soda, vegetables)	2
0,74	(vegetables)	1	0,32	(waffles, vegetables)	2
0,39	(waffles)	1	0,32	(yogurt, vegetables)	2

7. Проведен ассоциативный анализ при уровне поддержки 0.15 для нового датасета. Выведены все наборы, размер которых больше 1 и в котором есть 'yogurt' или 'waffles'.

support	itemsets	length	support	itemsets	length
0,17	(waffles, aluminum foil)	2	0,16	(yogurt, ice cream)	2
0,18	(yogurt, aluminum foil)	2	0,18	(waffles, lunch meat)	2
0,16	(waffles, bagels)	2	0,16	(yogurt, lunch meat)	2
0,16	(yogurt, bagels)	2	0,17	(yogurt, milk)	2
0,16	(cereals, waffles)	2	0,17	(poultry, waffles)	2
0,17	(yogurt, cereals)	2	0,18	(poultry, yogurt)	2
0,17	(waffles, cheeses)	2	0,18	(waffles, soda)	2
0,17	(yogurt, cheeses)	2	0,17	(yogurt, soda)	2
0,17	(waffles, dinner rolls)	2	0,32	(waffles, vegetables)	2
0,17	(yogurt, dinner rolls)	2	0,32	(yogurt, vegetables)	2

0,18	(waffles, dishwashing liquid/detergent)	2	0,17	(yogurt, waffles)	2
0,16	(yogurt, dishwashing liquid/detergent)	2	0,15	(vegetables, yogurt, aluminum foil)	3
0,17	(waffles, eggs)	2	0,16	(yogurt, vegetables, eggs)	3
0,17	(yogurt, eggs)	2	0,16	(waffles, lunch meat, vegetables)	3
0,17	(ice cream, waffles)	2	0,15	(poultry, yogurt, vegetables)	3

8. Сформирован датасет из элементов, которые не попали в датасет в п. 6, датасет приведен к удобному для анализа виду.

```
diff = set(list(df)) - set(list(df_new))
diff_items = [ list(elem)[0] for elem in results['itemsets']]
diff_dataset = [[elem for elem in all_data[all_data[1] == id][2] if
elem not in diff_items] for id in unique_id]
te = TransactionEncoder()
te_ary = te.fit_transform(diff_dataset)
df_new = pd.DataFrame(te_ary, columns=te.columns_)
```

support	itemsets	support	itemsets
0,37	(all- purpose)	0,37	(sandwich bags)
0,38	(aluminum foil)	0,35	(sandwich loaves)
0,39	(bagels)	0,37	(shampoo)
0,37	(beef)	0,38	(soap)
0,37	(butter)	0,39	(soda)
0,40	(cereals)	0,37	(spaghetti sauce)
0,39	(cheeses)	0,36	(sugar)
0,38	(coffee/tea)	0,38	(toilet paper)
0,39	(dinner rolls)	0,37	(tortillas)
0,39	(dishwashing liquid/detergent)	0,74	(vegetables)
0,39	(eggs)	0,39	(waffles)
0,35	(flour)	0,38	(yogurt)
0,37	(fruits)	0,31	(vegetables, aluminum foil)
0,35	(hand soap)	0,30	(vegetables, bagels)
0,40	(ice cream)	0,31	(vegetables, cereals)
0,38	(individual meals)	0,31	(cheeses, vegetables)
0,38	(juice)	0,31	(dinner rolls, vegetables)
0,37	(ketchup)	0,31	(vegetables, dishwashing liquid/detergent)

0,38	(laundry detergent)	0,33	(eggs, vegetables)
0,40	(lunch meat)	0,30	(vegetables, ice cream)
0,38	(milk)	0,31	(vegetables, laundry detergent)
0,38	(mixes)	0,31	(vegetables, lunch meat)
0,36	(paper towels)	0,33	(poultry, vegetables)
0,37	(pasta)	0,31	(vegetables, soda)
0,36	(pork)	0,32	(waffles, vegetables)
0,42	(poultry)	0,32	(vegetables, yogurt)

9. Написано правило для вывода всех наборов, в которых хотя бы два элемента начинаются на 's':

```
two_elems_starts_with_s = lambda df: df[df['itemsets'].apply(
    lambda x:
        np.fromiter(map(lambda y: y.startswith('s'), x),
dtype=bool).sum()>=2
    )
]
```

10. Написано правило для вывода всех наборов, для которых уровень поддержки изменяется от 0.1 до 0.25

```
subset_10_25 = lambda df: df[np.logical_and(df.support>=0.1,
df.support <= 0.25)]
```

Вывод

Изучены методы частотного анализа из библиотеки MLxtend.

Основным алгоритмом при изучении был алгоритм apriori.