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

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Машинное обучение»

Студенты гр. 6304 _____ Тимофеев А.А.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2020

Цель работы

Ознакомиться с методами частотного анализа из библиотеки MLxtend

Ход работы

Загрузка данных

1. Был создан датафрейм Pandas на основе загруженного датасета (https://www.kaggle.com/acostasg/random-shopping-cart)

Подготовка данных

1. Был получен датасет, где строки представляют собой списки покупок отдельных клиентов. Если клиент купил соответствующий товар, в столбце стоит Ttrue, иначе - False.

	all-purpose	aluminum foil	bagels	beef	butter	cereals	cheeses	coffee/tea	dinner rolls	dishwashing liquid/detergent	...	shampoo	soap	soda	spaghetti sauce	sugar	toilet paper	tortillas	vegetables	waffles	yogurt
0	True	True	False	True	True	False	False	False	True	False	...	True	True	True	False	False	False	False	True	False	True
1	False	True	False	False	False	True	True	False	False	True	...	True	False	False	False	False	True	True	True	True	True
2	False	False	True	False	False	True	True	False	True	False	...	True	True	True	True	False	True	False	True	False	False
3	True	False	False	False	False	True	False	False	False	False	...	False	False	True	False	False	True	False	False	False	False
4	True	False	False	False	False	False	False	False	True	False	...	False	False	True	True	False	True	True	True	True	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1134	True	False	False	True	False	True	True	True	True	True	...	True	True	False	False	True	False	False	False	False	False
1135	False	False	False	False	False	True	True	True	True	True	...	False	True	False	True	False	False	False	True	False	False
1136	False	False	True	True	False	False	False	False	True	True	...	True	True	False	False	True	False	True	True	False	True
1137	True	False	False	True	False	False	True	False	False	False	...	False	True	True	True	True	True	False	True	True	True
1138	False	False	False	False	False	False	False	False	False	False	...	True	False	True	False	False	False	False	True	False	False

Рисунок 1 – Получившийся датасет

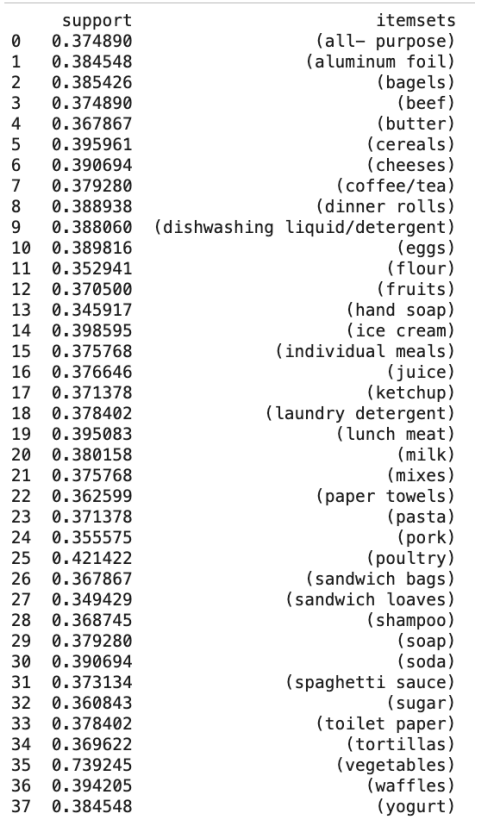
Ассоциативный анализ с использованием алгоритма Apriori

1. Были получены наборы с уровнем поддержки выше 0.3. Данные наборы были куплены не менее 30% клиентами.

	support	itemsets	length
0	0.374890	(all- purpose)	1
1	0.384548	(aluminum foil)	1
2	0.385426	(bagels)	1
3	0.374890	(beef)	1
4	0.367867	(butter)	1
5	0.395961	(cereals)	1
6	0.390694	(cheeses)	1
7	0.379280	(coffee/tea)	1
8	0.388938	(dinner rolls)	1
9	0.388060	(dishwashing liquid/detergent)	1
10	0.389816	(eggs)	1
11	0.352941	(flour)	1
12	0.370500	(fruits)	1
13	0.345917	(hand soap)	1
14	0.398595	(ice cream)	1
15	0.375768	(individual meals)	1
16	0.376646	(juice)	1
17	0.371378	(ketchup)	1
18	0.378402	(laundry detergent)	1
19	0.395083	(lunch meat)	1
20	0.380158	(milk)	1
21	0.375768	(mixes)	1
22	0.362599	(paper towels)	1
23	0.371378	(pasta)	1
24	0.355575	(pork)	1
25	0.421422	(poultry)	1
26	0.367867	(sandwich bags)	1
27	0.349429	(sandwich loaves)	1
28	0.368745	(shampoo)	1
29	0.379280	(soap)	1
30	0.390694	(soda)	1
31	0.373134	(spaghetti sauce)	1
32	0.360843	(sugar)	1
33	0.378402	(toilet paper)	1
34	0.369622	(tortillas)	1
35	0.739245	(vegetables)	1
36	0.394205	(waffles)	1
37	0.384548	(yogurt)	1
38	0.310799	(aluminum foil, vegetables)	2
39	0.300263	(bagels, vegetables)	2
40	0.310799	(cereals, vegetables)	2
41	0.309043	(vegetables, cheeses)	2
42	0.308165	(vegetables, dinner rolls)	2
43	0.306409	(dishwashing liquid/detergent, vegetables)	2
44	0.326602	(eggs, vegetables)	2
45	0.302897	(ice cream, vegetables)	2
46	0.309043	(laundry detergent, vegetables)	2
47	0.311677	(lunch meat, vegetables)	2
48	0.331870	(poultry, vegetables)	2
49	0.305531	(soda, vegetables)	2
50	0.315189	(waffles, vegetables)	2
51	0.319579	(vegetables, yogurt)	2

Рисунок 2 – Наборы с уровнем поддержки выше 0.3

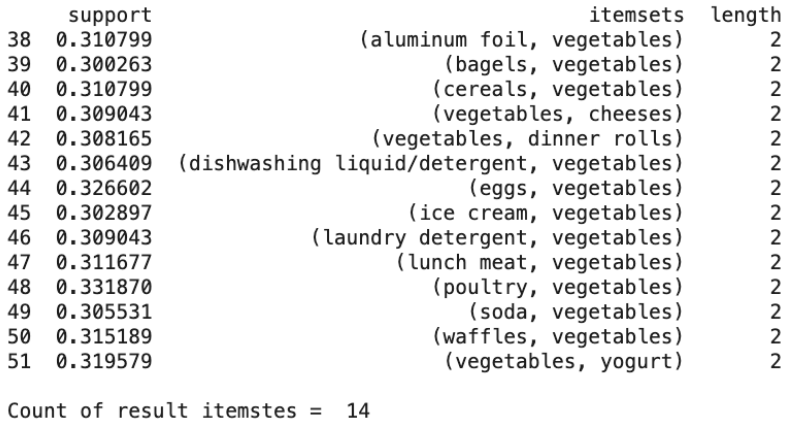
2. Были получены наборы с уровнем поддержки выше 0.3 и размером 1.



	support	itemsets
0	0.374890	(all- purpose)
1	0.384548	(aluminum foil)
2	0.385426	(bagels)
3	0.374890	(beef)
4	0.367867	(butter)
5	0.395961	(cereals)
6	0.390694	(cheeses)
7	0.379280	(coffee/tea)
8	0.388938	(dinner rolls)
9	0.388060	(dishwashing liquid/detergent)
10	0.389816	(eggs)
11	0.352941	(flour)
12	0.370500	(fruits)
13	0.345917	(hand soap)
14	0.398595	(ice cream)
15	0.375768	(individual meals)
16	0.376646	(juice)
17	0.371378	(ketchup)
18	0.378402	(laundry detergent)
19	0.395083	(lunch meat)
20	0.380158	(milk)
21	0.375768	(mixes)
22	0.362599	(paper towels)
23	0.371378	(pasta)
24	0.355575	(pork)
25	0.421422	(poultry)
26	0.367867	(sandwich bags)
27	0.349429	(sandwich loaves)
28	0.368745	(shampoo)
29	0.379280	(soap)
30	0.390694	(soda)
31	0.373134	(spaghetti sauce)
32	0.360843	(sugar)
33	0.378402	(toilet paper)
34	0.369622	(tortillas)
35	0.739245	(vegetables)
36	0.394205	(waffles)
37	0.384548	(yogurt)

Рисунок 3 – Наборы с уровнем поддержки выше 0.3 и размером 1

3. Были получены наборы с уровнем поддержки выше 0.3 и размером 2, а также их количество.



	support	itemsets	length
38	0.310799	(aluminum foil, vegetables)	2
39	0.300263	(bagels, vegetables)	2
40	0.310799	(cereals, vegetables)	2
41	0.309043	(vegetables, cheeses)	2
42	0.308165	(vegetables, dinner rolls)	2
43	0.306409	(dishwashing liquid/detergent, vegetables)	2
44	0.326602	(eggs, vegetables)	2
45	0.302897	(ice cream, vegetables)	2
46	0.309043	(laundry detergent, vegetables)	2
47	0.311677	(lunch meat, vegetables)	2
48	0.331870	(poultry, vegetables)	2
49	0.305531	(soda, vegetables)	2
50	0.315189	(waffles, vegetables)	2
51	0.319579	(vegetables, yogurt)	2

Count of result itemstes = 14

Рисунок 4 – Наборы с уровнем поддержки выше 0.3 и размером 2

4. Построен график зависимости количества наборов от уровня поддержки, а также определены значения уровня поддержки, при которых перестают генерироваться наборы размера 1,2,3, и т.д. График представлен на рисунке 5.

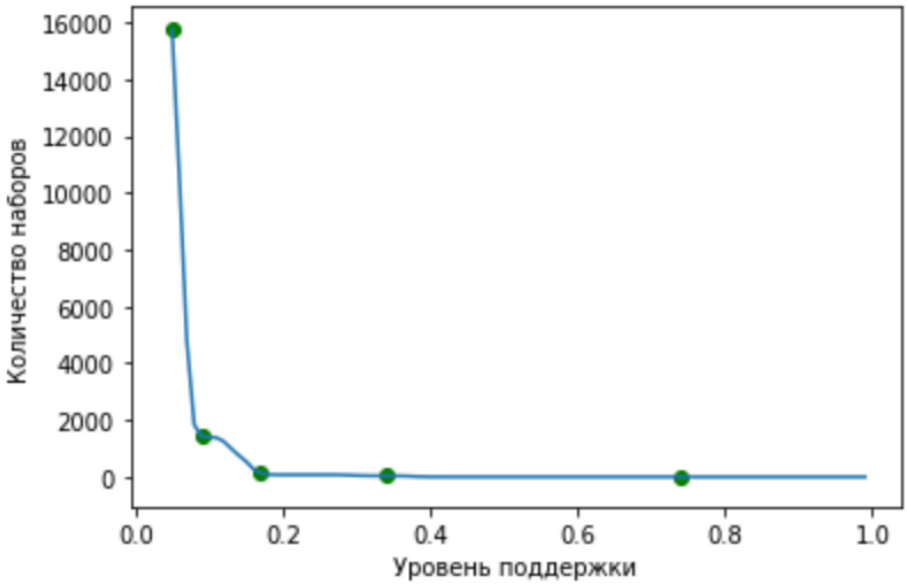


Рисунок 5 – График зависимости количества наборов от уровня поддержки

5. Был построен датасет только из тех элементов, которые попадают в наборы размером 1 при уровне поддержки 0.38 и приведен к формату, который можно обработать.
6. Для нового датасета был проведен анализ с уровнем поддержки 0.3, в результатах отсутствуют наборы, чей уровень поддержки в исходном датасете был ниже 0.38.

	support	itemsets
0	0.384548	(aluminum foil)
1	0.385426	(bagels)
2	0.395961	(cereals)
3	0.390694	(cheeses)
4	0.388938	(dinner rolls)
5	0.388060	(dishwashing liquid/detergent)
6	0.389816	(eggs)
7	0.398595	(ice cream)
8	0.395083	(lunch meat)
9	0.380158	(milk)
10	0.421422	(poultry)
11	0.390694	(soda)
12	0.739245	(vegetables)
13	0.394205	(waffles)
14	0.384548	(yogurt)
15	0.310799	(aluminum foil, vegetables)
16	0.300263	(bagels, vegetables)
17	0.310799	(cereals, vegetables)
18	0.309043	(vegetables, cheeses)
19	0.308165	(vegetables, dinner rolls)
20	0.306409	(dishwashing liquid/detergent, vegetables)
21	0.326602	(eggs, vegetables)
22	0.302897	(ice cream, vegetables)
23	0.311677	(lunch meat, vegetables)
24	0.331870	(poultry, vegetables)
25	0.305531	(soda, vegetables)
26	0.315189	(waffles, vegetables)
27	0.319579	(vegetables, yogurt)

Count of result itemstes = 28

Рисунок 6 – Результаты анализа для новой выборки

7. Был проведен ассоциативный анализ при уровне поддержки 0.15 для нового датасета, а также выведены все наборы, размер которых больше 1 и в которых есть 'yogurt' или 'waffles'.

```
      support                                itemsets   res
27   0.169447                (aluminum foil, waffles)  True
28   0.177349                 (aluminum foil, yogurt)  True
40   0.159789                       (bagels, waffles)  True
41   0.162423                        (bagels, yogurt)  True
52   0.160667                     (cereals, waffles)  True
53   0.172081                      (cereals, yogurt)  True
63   0.172959                     (waffles, cheeses)  True
64   0.172081                      (yogurt, cheeses)  True
73   0.169447                (waffles, dinner rolls)  True
74   0.166813                 (dinner rolls, yogurt)  True
82   0.175593  (dishwashing liquid/detergent, waffles)  True
83   0.158033   (dishwashing liquid/detergent, yogurt)  True
90   0.169447                         (eggs, waffles)  True
91   0.174715                          (eggs, yogurt)  True
97   0.172959                   (ice cream, waffles)  True
98   0.156277                    (ice cream, yogurt)  True
103  0.184372                  (lunch meat, waffles)  True
104  0.161545                   (lunch meat, yogurt)  True
108  0.167691                         (milk, yogurt)  True
111  0.166813                     (poultry, waffles)  True
112  0.180860                      (poultry, yogurt)  True
114  0.177349                        (waffles, soda)  True
115  0.167691                         (soda, yogurt)  True
116  0.315189                  (waffles, vegetables)  True
117  0.319579                   (vegetables, yogurt)  True
118  0.173837                      (waffles, yogurt)  True
119  0.152766     (aluminum foil, vegetables, yogurt)  True
128  0.157155               (eggs, vegetables, yogurt)  True
130  0.157155       (lunch meat, waffles, vegetables)  True
131  0.152766           (poultry, vegetables, yogurt)  True

Count of result itemstes =  30
```

Рисунок 7 – Результаты анализа для новой выборки

8. Был построен датасет, из тех элементов, которые не попали в датасет в п. 5, и приведен к удобному для анализа виду.
9. Был проведен анализ aprioti для полученного датасета.

	support	itemsets
0	0.374890	(all- purpose)
1	0.374890	(beef)
2	0.367867	(butter)
3	0.379280	(coffee/tea)
4	0.352941	(flour)
5	0.370500	(fruits)
6	0.345917	(hand soap)
7	0.375768	(individual meals)
8	0.376646	(juice)
9	0.371378	(ketchup)
10	0.378402	(laundry detergent)
11	0.375768	(mixes)
12	0.362599	(paper towels)
13	0.371378	(pasta)
14	0.355575	(pork)
15	0.367867	(sandwich bags)
16	0.349429	(sandwich loaves)
17	0.368745	(shampoo)
18	0.379280	(soap)
19	0.373134	(spaghetti sauce)
20	0.360843	(sugar)
21	0.378402	(toilet paper)
22	0.369622	(tortillas)

Рисунок 8 – Результаты анализа для новой выборки

10. Было написано правило, для вывода всех наборов, в которых хотя бы два элемента начинаются на 's'.

```
      support                                    itemsets   res
675   0.137840               (sandwich bags, sandwich loaves)  True
676   0.146620                       (shampoo, sandwich bags)  True
677   0.158911                          (soap, sandwich bags)  True
678   0.162423                          (sandwich bags, soda)  True
679   0.147498               (spaghetti sauce, sandwich bags)  True
680   0.131694                         (sandwich bags, sugar)  True
686   0.150132                     (shampoo, sandwich loaves)  True
687   0.158033                        (soap, sandwich loaves)  True
688   0.141352                        (sandwich loaves, soda)  True
689   0.150132             (spaghetti sauce, sandwich loaves)  True
690   0.136962                        (sugar, sandwich loaves)  True
696   0.151010                                 (shampoo, soap)  True
697   0.150132                                 (shampoo, soda)  True
698   0.139596                     (shampoo, spaghetti sauce)  True
699   0.147498                               (shampoo, sugar)  True
705   0.174715                                   (soap, soda)  True
706   0.160667                      (soap, spaghetti sauce)  True
707   0.154522                                (soap, sugar)  True
713   0.167691                      (spaghetti sauce, soda)  True
714   0.162423                                (sugar, soda)  True
720   0.144864                     (spaghetti sauce, sugar)  True
1351  0.115013   (sandwich bags, sandwich loaves, vegetables)  True
1352  0.122915           (shampoo, sandwich bags, vegetables)  True
1353  0.129939              (soap, sandwich bags, vegetables)  True
1354  0.129061               (sandwich bags, soda, vegetables)  True
1355  0.123793   (spaghetti sauce, sandwich bags, vegetables)  True
1356  0.113257             (sandwich bags, sugar, vegetables)  True
1361  0.129061         (shampoo, sandwich loaves, vegetables)  True
1362  0.132572            (soap, sandwich loaves, vegetables)  True
1363  0.121159            (sandwich loaves, soda, vegetables)  True
1364  0.122915  (spaghetti sauce, sandwich loaves, vegetables)  True
1365  0.121159           (sandwich loaves, sugar, vegetables)  True
1370  0.124671                    (shampoo, soap, vegetables)  True
1371  0.128183                    (shampoo, soda, vegetables)  True
1372  0.117647          (shampoo, spaghetti sauce, vegetables)  True
1373  0.122037                   (shampoo, sugar, vegetables)  True
1378  0.141352                       (soap, soda, vegetables)  True
1379  0.136962             (soap, spaghetti sauce, vegetables)  True
1380  0.127305                      (soap, sugar, vegetables)  True
1385  0.138718             (spaghetti sauce, soda, vegetables)  True
1386  0.136084                      (sugar, soda, vegetables)  True
1391  0.124671            (spaghetti sauce, sugar, vegetables)  True

Count of result itemstes =  42
```

Рисунок 9 – Результаты применения правила

11. Было написано правило, для вывода всех наборов, для которых уровень поддержки изменяется от 0.1 до 0.25.

	support	itemsets	res
38	0.157155	(aluminum foil, all- purpose)	True
39	0.150132	(all- purpose, bagels)	True
40	0.144864	(all- purpose, beef)	True
41	0.147498	(all- purpose, butter)	True
42	0.151010	(cereals, all- purpose)	True
...	...	...	...
1401	0.135206	(waffles, vegetables, toilet paper)	True
1402	0.130817	(yogurt, vegetables, toilet paper)	True
1403	0.121159	(tortillas, waffles, vegetables)	True
1404	0.130817	(tortillas, vegetables, yogurt)	True
1405	0.146620	(waffles, vegetables, yogurt)	True

Рисунок 10 – Результаты применения правила

Выводы

В ходе выполнения данной лабораторной работы было произведено знакомство с методами частотного анализа из библиотеки MLxtend.

ПРИЛОЖЕНИЕ А

Исходный код

```
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt

mpl.rcParams['figure.dpi'] = 200

all_data = pd.read_csv('dataset_group.csv',header=None)
all_data

unique_id = list(set(all_data[1]))
print('Количество уникальных покупателей : {}'.format(len(unique_id)))

items = list(set(all_data[2]))
print('Количество уникальных товаров: {}'.format(len(items)))

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

from mlxtend.preprocessing import TransactionEncoder

te = TransactionEncoder()
te_ary = te.fit(dataset).transform(dataset)
df = pd.DataFrame(te_ary, columns=te.columns_)
df

from mlxtend.frequent_patterns import apriori

results = apriori(df, min_support=0.3, use_colnames=True)
results['length'] = results['itemsets'].apply(lambda x: len(x))
print(results)

results = apriori(df, min_support=0.3, use_colnames=True, max_len=1)
print(results)

results = apriori(df, min_support=0.3, use_colnames=True)
results['length'] = results['itemsets'].apply(lambda x: len(x))
results = results[results['length'] == 2]
print(results)
print('\nCount of result itemstes = ',len(results))

supports = np.arange(0.05, 1, 0.01)
num_of_sets = []
max_nums = []
max_num = None
for sup in supports:
    sup_results = apriori(df, min_support=sup, use_colnames=True)
    sup_results['length'] = sup_results['itemsets'].apply(lambda x: len(x))
    num = len(sup_results)
    current_max_num = np.max(sup_results['length'])
    if max_num is None:
        max_num = current_max_num
        max_nums.append(sup.round(2))
        plt.scatter(sup, num, c='g')
    elif current_max_num < max_num:
        max_num = current_max_num
        max_nums.append(sup.round(2))
        plt.scatter(sup, num, c='g')
```

```
    elif np.isnan(current_max_num) and not np.isnan(max_num):
        max_num = current_max_num
        max_nums.append(sup.round(2))
        plt.scatter(sup, num, c='g')
    num_of_sets.append(num)
plt.plot(supports, num_of_sets)
plt.xlabel('Уровень поддержки')
plt.ylabel('Количество наборов')
plt.show()

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_ary = te.fit(new_dataset).transform(new_dataset)
ndf = pd.DataFrame(te_ary, columns=te.columns_)
ndf

results = apriori(ndf, min_support=0.3, use_colnames=True)
print(results)
print('\nCount of result itemstes = ',len(results))

results = apriori(ndf, min_support=0.15, use_colnames=True)
results['res'] = results['itemsets'].apply(lambda x: len(x) > 1 and ('yogurt' in x or
'waffles' in x))
results = results[results['res']]
print(results)
print('\nCount of result itemstes = ', len(results))

difference = set(list(df)) - set(list(ndf))
one_more_df = [[elem for elem in all_data[all_data[1] == id][2] if elem in
difference] for id in unique_id]
te = TransactionEncoder()
te_ary = te.fit_transform(one_more_df)
omdf = pd.DataFrame(te_ary, columns=te.columns_)
omdf

results = apriori(omdf, min_support=0.3, use_colnames=True)
results

results = apriori(df, min_support=0.1, use_colnames=True)
results['res'] = results['itemsets'].apply(lambda x: len([item for item in x if
item.startswith('s')]) >= 2)
results = results[results['res']]
print(results)
print('\nCount of result itemstes = ', len(results))

results = apriori(df, min_support=0.1, use_colnames=True)
results['res'] = results['support'].apply(lambda support: support > 0.1 and support <
0.25)
results = results[results['res']]
results
```