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

ОТЧЕТ

по лабораторной работе № 4

по дисциплине «Машинное обучение»

Тема: Ассоциативный анализ

Студенты гр. 6304 _____ Григорьев И.С.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург
2020

Цель работы

Ознакомиться с методами ассоциативного анализа из библиотеки *MLxtend*.

Ход работы

Загрузка данных

1. Датасет загружен в датафрейм. Вид данных представлен на рис. 1.

	Item(s)	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9	...	Item 23	Item 24	Item 25	Item 26	Item 27
0	4	citrus fruit	semi-finished bread	margarine	ready soups	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
1	3	tropical fruit	yogurt	coffee	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
2	1	whole milk	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
3	4	pip fruit	yogurt	cream cheese	meat spreads	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
4	4	other vegetables	whole milk	condensed milk	long life bakery product	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9830	17	sausage	chicken	beef	hamburger meat	citrus fruit	grapes	root vegetables	whole milk	butter	...	NaN	NaN	NaN	NaN	NaN
9831	1	cooking chocolate	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
9832	10	chicken	citrus fruit	other vegetables	butter	yogurt	frozen dessert	domestic eggs	rolls/buns	rum	...	NaN	NaN	NaN	NaN	NaN
9833	4	semi-finished bread	bottled water	soda	bottled beer	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
9834	5	chicken	tropical fruit	other vegetables	vinegar	shopping bags	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN

9835 rows × 33 columns

Рисунок 1 – Исходные данные

2. Данные переформированы (удалены значения NaN).
3. Получен список уникальных товаров.

```
Количество товаров 169 {'hygiene articles', 'sweet spreads', 'salty snack',
'mayonnaise', 'sparkling wine', 'dessert', 'liqueur', 'curd cheese', 'butter', 'bottled
beer', 'house keeping products', 'frozen meals', 'misc. beverages', 'liquor
(appetizer)', 'pickled vegetables', 'specialty vegetables', 'cooking chocolate',
'kitchen towels', 'potted plants', 'sausage', 'specialty bar', 'bottled water', 'canned
fish', 'soda', 'vinegar', 'sauces', 'grapes', 'liquor', 'whisky', 'organic sausage',
'beverages', 'nut snack', 'shopping bags', 'canned vegetables', 'onions', 'soups',
'chewing gum', 'photo/film', 'light bulbs', 'specialty chocolate', 'salt', 'baking
powder', 'toilet cleaner', 'whipped/sour cream', 'cake bar', 'condensed milk',
'cereals', 'ketchup', 'meat spreads', 'red/blush wine', 'rolls/buns', 'skin care',
'berries', 'butter milk', 'specialty fat', 'frozen fish', 'cling film/bags',
'decalcifier', 'bags', 'fish', 'detergent', 'frozen potato products', 'Instant food
products', 'pip fruit', 'salad dressing', 'beef', 'prosecco', 'curd', 'napkins', 'ham',
'dental care', 'roll products', 'spices', 'oil', 'turkey', 'dish cleaner', 'tidbits',
'tea', 'domestic eggs', 'rum', 'spread cheese', 'flower soil/fertilizer', 'chocolate
marshmallow', 'baby cosmetics', 'baby food', 'brown bread', 'softener', 'candy',
'zwieback', 'male cosmetics', 'abrasive cleaner', 'nuts/prunes', 'soap', 'canned
fruit', 'finished products', 'pasta', 'citrus fruit', 'candles', 'frozen vegetables',
```

```
'honey', 'flour', 'margarine', 'pork', 'liver loaf', 'white wine', 'whole milk',
'potato products', 'long life bakery product', 'cream cheese', 'frozen chicken',
'brandy', 'preservation products', 'cocoa drinks', 'female sanitary products',
'processed cheese', 'cat food', 'tropical fruit', 'rubbing alcohol', 'make up remover',
'yogurt', 'waffles', 'frankfurter', 'other vegetables', 'chocolate', 'newspapers',
'UHT-milk', 'dog food', 'syrup', 'instant coffee', 'cream', 'jam', 'canned beer',
'packaged fruit/vegetables', 'frozen fruits', 'hard cheese', 'sliced cheese',
'chicken', 'soft cheese', 'organic products', 'bathroom cleaner', 'popcorn',
'fruit/vegetable juice', 'pet care', 'rice', 'cookware', 'seasonal products', 'dishes',
'artif. sweetener', 'coffee', 'flower (seeds)', 'herbs', 'sound storage medium', 'semi-
finished bread', 'specialty cheese', 'snack products', 'sugar', 'mustard', 'pudding
powder', 'frozen dessert', 'hamburger meat', 'white bread', 'root vegetables', 'hair
spray', 'kitchen utensil', 'cleaner', 'meat', 'ready soups', 'pastry', 'ice cream'}
```

FPGrowth и FPMax

1. Данные приведены к виду на рис. 2, удобному для анализа.

	Instant food products	UHT-milk	abrasive cleaner	artif. sweetener	baby cosmetics	baby food	bags	baking powder	bathroom cleaner	beef	...	turkey	vinegar	waffles	whipped/sou cream
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	..
9830	False	False	False	False	False	False	False	False	False	True	...	False	False	False	True
9831	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
9832	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
9833	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
9834	False	False	False	False	False	False	False	False	False	False	...	False	True	False	False

9835 rows × 169 columns

Рисунок 2 – Подготовленные данные

2. Проведен ассоциативный анализ с помощью алгоритма *FPGrowth* при уровне поддержки 0.03. Результат приведен на рис. 3.

	support	itemsets
5	0.255516	(whole milk)
8	0.193493	(other vegetables)
11	0.183935	(rolls/buns)
19	0.174377	(soda)
2	0.139502	(yogurt)
...	...	...
43	0.031012	(onions)
61	0.030605	(sausage, rolls/buns)
44	0.030503	(whole milk, citrus fruit)
42	0.030402	(specialty chocolate)
50	0.030097	(whole milk, pip fruit)

63 rows × 2 columns

Рисунок 3 – Результат *FPGrowth*

3. Аналогичный пункту 2 ассоциативный анализ проведен с помощью алгоритма *FPMax*. Результат представлен на рис. 4.

	support	itemsets
35	0.098526	(shopping bags)
31	0.080529	(bottled beer)
30	0.079817	(newspapers)
29	0.077682	(canned beer)
49	0.074835	(whole milk, other vegetables)
27	0.072293	(fruit/vegetable juice)
25	0.064870	(brown bread)
24	0.063447	(domestic eggs)
23	0.058973	(frankfurter)
22	0.058566	(margarine)
21	0.058058	(coffee)
20	0.057651	(pork)
48	0.056634	(whole milk, rolls/buns)
43	0.056024	(whole milk, yogurt)
19	0.055414	(butter)
18	0.053279	(curd)
17	0.052466	(beef)
16	0.052364	(napkins)
15	0.049619	(chocolate)
39	0.048907	(root vegetables, whole milk)
14	0.048094	(frozen vegetables)
38	0.047382	(root vegetables, other vegetables)
42	0.043416	(yogurt, other vegetables)
13	0.042908	(chicken)
47	0.042603	(rolls/buns, other vegetables)
37	0.042298	(whole milk, tropical fruit)
12	0.042095	(white bread)
46	0.040061	(whole milk, soda)
11	0.039654	(cream cheese)
10	0.038434	(waffles)
45	0.038332	(rolls/buns, soda)
9	0.037824	(salty snack)
8	0.037417	(long life bakery product)
7	0.037112	(dessert)
36	0.035892	(tropical fruit, other vegetables)
40	0.034367	(bottled water, whole milk)
41	0.034367	(yogurt, rolls/buns)
6	0.033859	(sugar)
5	0.033452	(UHT-milk)
33	0.033249	(whole milk, pastry)
3	0.033249	(berries)
4	0.033249	(hamburger meat)
2	0.032944	(hygiene articles)
44	0.032740	(other vegetables, soda)
26	0.032232	(whipped/sour cream, whole milk)
1	0.031012	(onions)
34	0.030605	(sausage, rolls/buns)
32	0.030503	(whole milk, citrus fruit)
0	0.030402	(specialty chocolate)
28	0.030097	(whole milk, pip fruit)

Рисунок 4 – Результат *FPMax*

4. Минимальные и максимальные значения уровня поддержки для набора из 1, 2 и т.д. объектов представлены в табл. 1.

Таблица 1 – Уровни поддержки

Длина набора	*FPGrowth*	*FPMax*
1	[0.0304, 0.25552]	[0.0304, 0.09853]
2	[0.0301, 0.07483]	[0.0301, 0.07483]

Из табл. 1 видно, что отличается только максимальный уровень поддержки для длины набора 1. Это так, потому что в *FPMax* набор не может быть частью другого набора большей длины. Наиболее часто встречающиеся наборы длины 1 вошли в наборы длины 2.

5. Гистограмма для 10 самых встречаемых товаров представлена на рис. 5. Данная гистограмма напрямую коррелирует с результатами ассоциативного анализа. Первые 10 самых встречаемых наборов товаров, полученных с помощью *FPGrowth*, продемонстрированы на рис. 6.

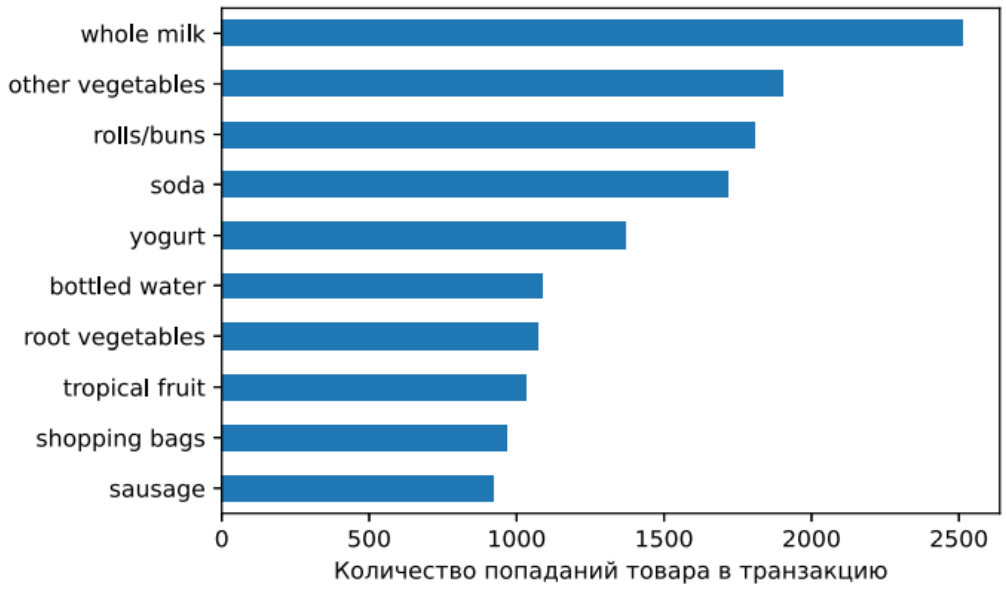

Рисунок 5 – 10 самых встречаемых товаров

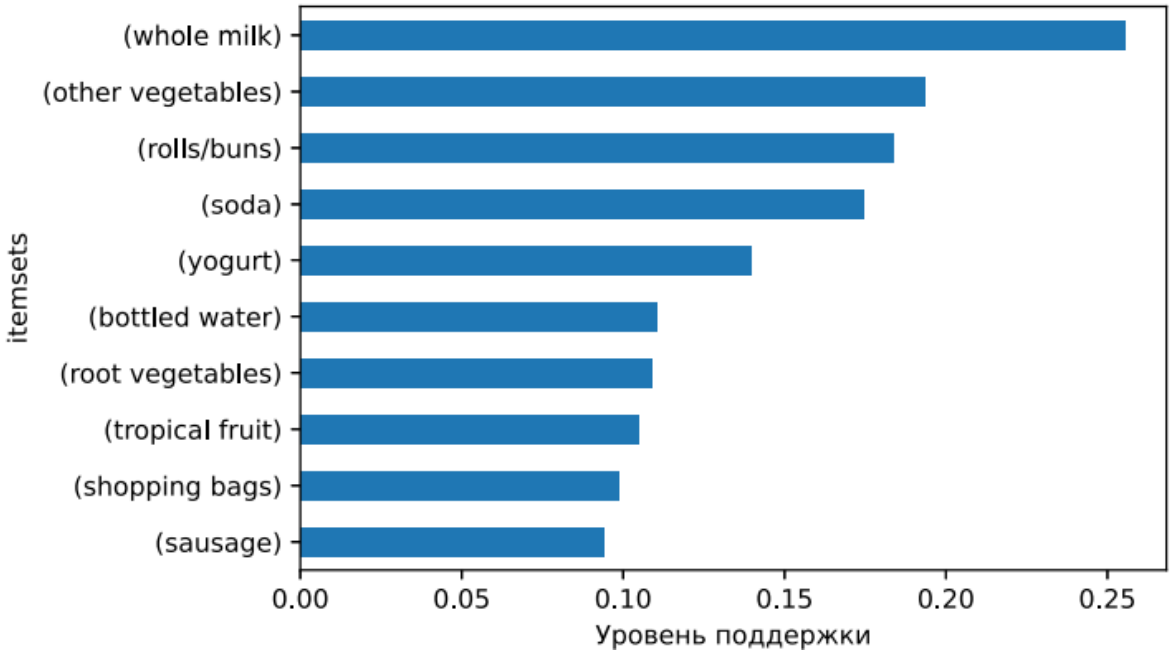

Рисунок 6 – Результат *FPGrowth*

6. Данные преобразованы и содержат только ограниченный набор товаров.
7. Для ограниченного набора товаров проведен анализ с помощью *FPGrowth* и *FPMax*. Результаты представлены на рис. 7-8 и табл. 2.

	support	itemsets
3	0.255516	(whole milk)
4	0.193493	(other vegetables)
5	0.183935	(rolls/buns)
8	0.174377	(soda)
1	0.139502	(yogurt)
7	0.110524	(bottled water)
10	0.108998	(root vegetables)
2	0.104931	(tropical fruit)
13	0.098526	(shopping bags)
12	0.093950	(sausage)
9	0.088968	(pastry)
0	0.082766	(citrus fruit)
6	0.080529	(bottled beer)
11	0.077682	(canned beer)
22	0.074835	(whole milk, other vegetables)
14	0.071683	(whipped/sour cream)
15	0.057651	(pork)
24	0.056634	(whole milk, rolls/buns)
17	0.056024	(whole milk, yogurt)
31	0.048907	(root vegetables, whole milk)
30	0.047382	(root vegetables, other vegetables)
19	0.043416	(yogurt, other vegetables)
23	0.042603	(rolls/buns, other vegetables)
21	0.042298	(whole milk, tropical fruit)
27	0.040061	(whole milk, soda)
26	0.038332	(rolls/buns, soda)
20	0.035892	(tropical fruit, other vegetables)
18	0.034367	(yogurt, rolls/buns)
25	0.034367	(bottled water, whole milk)
29	0.033249	(whole milk, pastry)
28	0.032740	(other vegetables, soda)
33	0.032232	(whipped/sour cream, whole milk)
32	0.030605	(sausage, rolls/buns)
16	0.030503	(whole milk, citrus fruit)

Рисунок 7 – Результат *FPGrowth* для ограниченного набора

	support	itemsets
7	0.098526	(shopping bags)
3	0.080529	(bottled beer)
2	0.077682	(canned beer)
21	0.074835	(whole milk, other vegetables)
0	0.057651	(pork)
20	0.056634	(whole milk, rolls/buns)
15	0.056024	(whole milk, yogurt)
11	0.048907	(root vegetables, whole milk)
10	0.047382	(root vegetables, other vegetables)
14	0.043416	(yogurt, other vegetables)
19	0.042603	(rolls/buns, other vegetables)
9	0.042298	(whole milk, tropical fruit)
18	0.040061	(whole milk, soda)
17	0.038332	(rolls/buns, soda)
8	0.035892	(tropical fruit, other vegetables)
13	0.034367	(yogurt, rolls/buns)
12	0.034367	(bottled water, whole milk)
5	0.033249	(whole milk, pastry)
16	0.032740	(other vegetables, soda)
1	0.032232	(whipped/sour cream, whole milk)
6	0.030605	(sausage, rolls/buns)
4	0.030503	(whole milk, citrus fruit)

Рисунок 7 – Результат *FPMax* для ограниченного набора

Таблица 2 – Уровни поддержки

Длина набора	*FPGrowth*	*FPMax*
1	[0.05765, 0.25552]	[0.05765, 0.09853]
2	[0.0305, 0.07483]	[0.0305, 0.07483]

Максимальные значения уровня поддержки не изменились, но изменились минимальные значения, т.к. из данных были удалены товары, которые имели наименьший уровень поддержки. Ограничение товаров изменило количество наборов в результатах алгоритмов, но не изменило уровень поддержки для оставшихся наборов.

8. Построен график изменения количества получаемых правил от уровня поддержки, представленный на рис. 8. На графике отдельно отображены кривые для набора товаров длины 1, 2 и т.д.

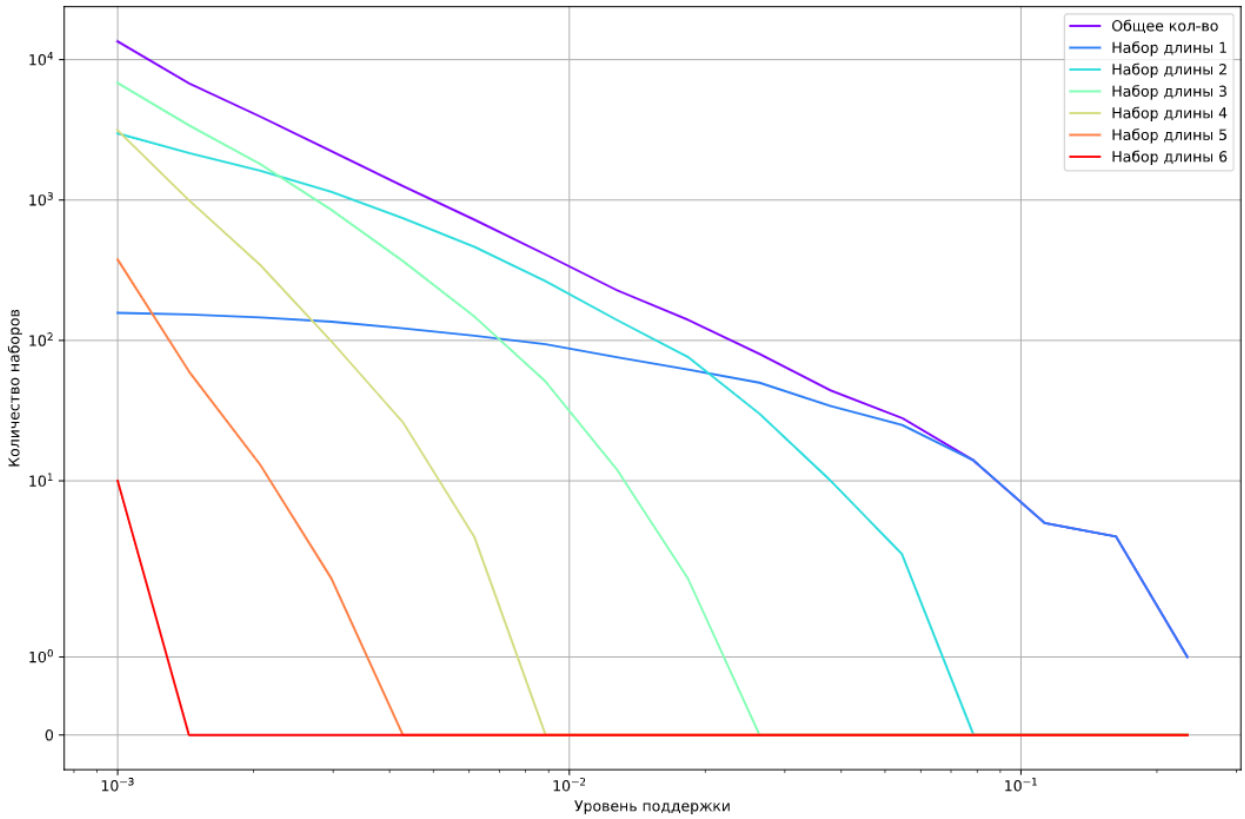

Рисунок 8 – Зависимость количества получаемых правил от уровня поддержки

Количество наборов уменьшается с увеличением уровня минимальной поддержки. Наборы большей длины перестают генерироваться раньше.

Ассоциативные правила

1. Набор данных сформирован так, чтобы размер транзакции был 2 и более.
2. Получены частоты наборов с помощью *FPGrowth*. Результат представлен на рис. 9.

	support	itemsets
0	0.241240	(yogurt)
1	0.185864	(tropical fruit)
2	0.421869	(whole milk)
3	0.335079	(other vegetables)
4	0.296214	(rolls/buns)
5	0.113371	(bottled beer)
6	0.185461	(bottled water)
7	0.146395	(citrus fruit)
8	0.267217	(soda)
9	0.196335	(root vegetables)
10	0.082763	(canned beer)
11	0.167539	(sausage)
12	0.166935	(shopping bags)
13	0.124245	(whipped/sour cream)
14	0.099476	(pork)
15	0.150624	(pastry)
16	0.110954	(whole milk, yogurt)
17	0.054168	(yogurt, soda)
18	0.068063	(yogurt, rolls/buns)
19	0.085985	(yogurt, other vegetables)
20	0.057994	(yogurt, tropical fruit)
21	0.071083	(tropical fruit, other vegetables)
22	0.083770	(whole milk, tropical fruit)
23	0.148208	(whole milk, other vegetables)
24	0.084374	(rolls/buns, other vegetables)
25	0.112163	(whole milk, rolls/buns)
26	0.068063	(bottled water, whole milk)
27	0.057390	(bottled water, soda)
28	0.060411	(whole milk, citrus fruit)
29	0.057189	(citrus fruit, other vegetables)
30	0.075916	(rolls/buns, soda)
31	0.079340	(whole milk, soda)
32	0.064841	(other vegetables, soda)
33	0.093838	(root vegetables, other vegetables)
34	0.096859	(root vegetables, whole milk)
35	0.051148	(root vegetables, yogurt)
36	0.060612	(sausage, rolls/buns)
37	0.059203	(whole milk, sausage)
38	0.053363	(sausage, other vegetables)
39	0.063834	(whipped/sour cream, whole milk)
40	0.057189	(whipped/sour cream, other vegetables)
41	0.065848	(whole milk, pastry)

Рисунок 9 – Результат *FPGrowth* для транзакций длины 2 и более

3. Проведен ассоциативный анализ. Расчет произведен на основе метрики *confidence*. Результат представлен на рис. 10. Значения метрик приведены в табл. 3. A – антецедент, C – консеквент.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(yogurt)	(whole milk)	0.241240	0.421869	0.110954	0.459933	1.090228	0.009183	1.070481
1	(yogurt)	(other vegetables)	0.241240	0.335079	0.085985	0.356427	1.063713	0.005150	1.033172
2	(tropical fruit)	(yogurt)	0.185864	0.241240	0.057994	0.312026	1.293423	0.013156	1.102890
3	(tropical fruit)	(other vegetables)	0.185864	0.335079	0.071083	0.382449	1.141370	0.008804	1.076706
4	(tropical fruit)	(whole milk)	0.185864	0.421869	0.083770	0.450704	1.068352	0.005359	1.052495
5	(whole milk)	(other vegetables)	0.421869	0.335079	0.148208	0.351313	1.048449	0.006849	1.025026
6	(other vegetables)	(whole milk)	0.335079	0.421869	0.148208	0.442308	1.048449	0.006849	1.036649
7	(rolls/buns)	(whole milk)	0.296214	0.421869	0.112163	0.378654	0.897564	-0.012801	0.930450
8	(bottled water)	(whole milk)	0.185461	0.421869	0.068063	0.366992	0.869921	-0.010177	0.913309
9	(bottled water)	(soda)	0.185461	0.267217	0.057390	0.309446	1.158033	0.007832	1.061153
10	(citrus fruit)	(whole milk)	0.146395	0.421869	0.060411	0.412655	0.978159	-0.001349	0.984313
11	(citrus fruit)	(other vegetables)	0.146395	0.335079	0.057189	0.390646	1.165836	0.008135	1.091192
12	(root vegetables)	(other vegetables)	0.196335	0.335079	0.093838	0.477949	1.426378	0.028050	1.273671
13	(root vegetables)	(whole milk)	0.196335	0.421869	0.096859	0.493333	1.169400	0.014031	1.141049
14	(sausage)	(rolls/buns)	0.167539	0.296214	0.060612	0.361779	1.221342	0.010985	1.102730
15	(sausage)	(whole milk)	0.167539	0.421869	0.059203	0.353365	0.837619	-0.011477	0.894062
16	(sausage)	(other vegetables)	0.167539	0.335079	0.053363	0.318510	0.950552	-0.002776	0.975687
17	(whipped/sour cream)	(whole milk)	0.124245	0.421869	0.063834	0.513776	1.217858	0.011419	1.189023
18	(whipped/sour cream)	(other vegetables)	0.124245	0.335079	0.057189	0.460292	1.373683	0.015557	1.232002
19	(pastry)	(whole milk)	0.150624	0.421869	0.065848	0.437166	1.036260	0.002304	1.027179

Рисунок 10 – Результат ассоциативного анализа

Таблица 3 – Метрики ассоциативного анализа

Метрика	Значение	Формула	Диапазон
antecedent support	Поддержка A	$\sup(A)$	[0,1]
consequent support	Поддержка C	$\sup(C)$	[0,1]
support	Поддержка набора из A и C	$\sup(A \rightarrow C) = \sup(A \cup C)$	[0,1]

confidence	Вероятность увидеть C в транзакции, содержащей A. Confidence 1, если A и C всегда находятся вместе в транзакциях.	$\text{confidence}(A \to C) = \frac{\sup(A \to C)}{\sup(A)}$	[0,1]
lift	Как часто A и C возникают вместе, чем если бы они были статистически независимы. Если A и C независимы, то lift = 1.	$\text{lift}(A \to C) = \frac{\text{confidence}(A \to C)}{\sup(C)}$	$[0, \infty]$
leverage	Разница между частотой появления A и C вместе с частотой, при независимых A и C. Если A и C независимы, то leverage = 0.	$\text{leverage}(A \to C) = \sup(A \to C) - \sup(A) \times \sup(C)$	[-1,-1]
conviction	Большое значение означает, что C сильно зависит от A. Если A и C независимы, то conviction = 1. При абсолютной зависимости conviction=∞	$\text{conviction}(A \to C) = \frac{1 - \sup(C)}{1 - \text{confidence}(A \to C)}$	$[0, \infty]$

4. Проведено построение ассоциативных правил для различных метрик: *support*, *confidence*, *lift*, *leverage*, *conviction*. Значение *min_threshold* выбиралось такое, чтобы выводилось не менее 10 правил. Средние значения, медианы и СКО для каждой из метрик представлены в табл. 4.

Таблица 4 – Статистика для каждой из метрик

Метрика	Среднее	Медиана	СКО
support	0.07468	0.06696	0.02255
confidence	0.40149	0.38655	0.06209
lift	1.043	1.05608	0.18326
leverage	0.01553	0.01359	0.00606
conviction	1.0172	1.02285	0.08399

5. Построен граф для полученных ассоциативных правил для метрики *confidence* и *min_threshold* равным 0.4. Граф ориентирован от антецедента к консеквенту, ширина ребра отображает уровень поддержки правила, а подпись на ребре отображает *confidence*. Граф представлен на рис. 11.

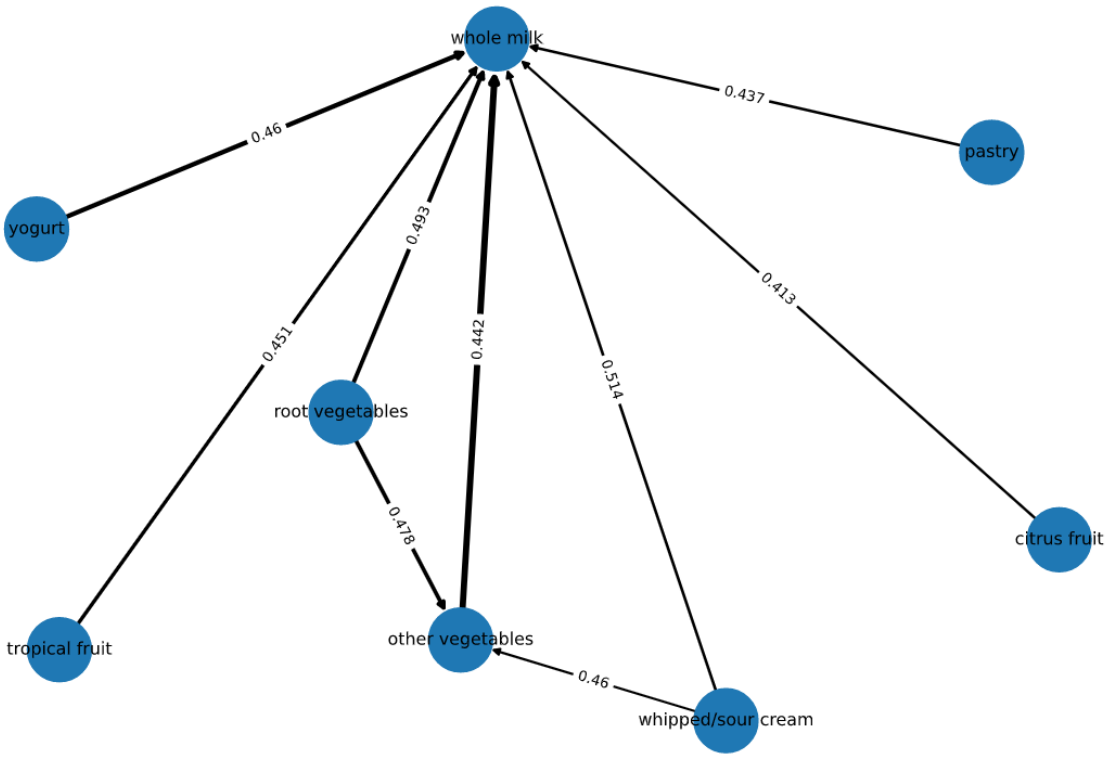

Рисунок 11 – Граф ассоциативных правил

6. Для визуализации ассоциативных правил также можно использовать *heatmap*. Визуализация полученных ранее ассоциативных правил с помощью *heatmap* представлена на рис. 12.

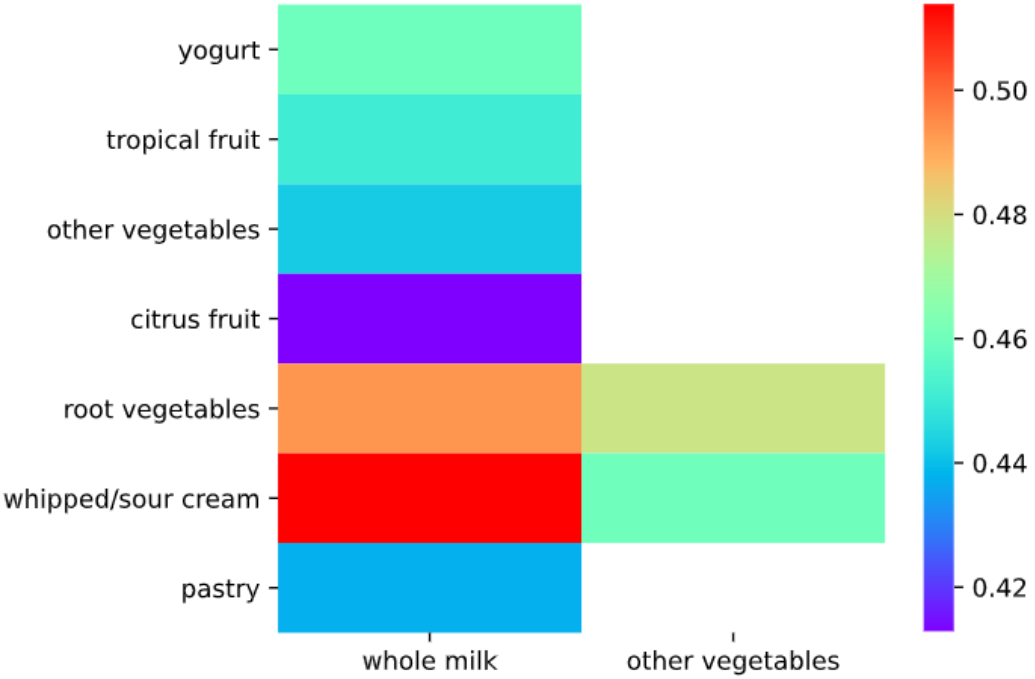

Рисунок 12 – *Heatmap* ассоциативных правил

Выводы

В ходе лабораторной работы изучены методы ассоциативного анализа из библиотеки *MLxtend*: алгоритмы *FPGrowth* и *FPMax* позволяют выделить частовстречающиеся наборы элементов для заданного минимального уровня поддержки. Различие данных алгоритмов заключается в том, что наборы в *FPMax* не могут быть частью других наборов большей длины. Ассоциативные правила можно генерировать с помощью алгоритма *association_rules*, который принимает на вход метрику и ее минимальное значение для расчета.

Приложение А

Код программы на python

```
# To add a new cell, type '# %%'
# To add a new markdown cell, type '# %% [markdown]'

# %%
import pandas as pd
import numpy as np
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import fpgrowth, fpmax, association_rules
import matplotlib.pyplot as plt
import networkx as nx
import seaborn as sns

# %%
all_data = pd.read_csv('groceries - groceries.csv')
all_data #Видно, что датафрейм содержит NaN значения

# %%
np_data = all_data.to_numpy()
np_data = [[elem for elem in row[1:] if isinstance(elem,str)] for row in
np_data]

# %%
unique_items = set()
for row in np_data:
    for elem in row:
        unique_items.add(elem)
print('Количество товаров', len(unique_items), unique_items)

# %%
te = TransactionEncoder()
te_ary = te.fit(np_data).transform(np_data)
data = pd.DataFrame(te_ary, columns=te.columns_)
data

# %%
fpg_result = fpgrowth(data, min_support=0.03, use_colnames = True).sort_values('suppo
rt', ascending=False)
fpg_result

# %%
def printMinMaxSupport(result):
    curr_len = 1
    while True:
        sups = result[result['itemsets'].apply(lambda r: len(r) == curr_len)]['suppor
t']
        if len(sups) == 0:
            break
        print('Длина набора {len}: поддержка [{min}, {max}]'.format(len=curr_len, min
=round(np.min(sups), 5), max=round(np.max(sups), 5)))
        curr_len += 1
printMinMaxSupport(fpg_result)

# %%
fpm_result = fpmax(data, min_support=0.03, use_colnames = True).sort_values('support'
, ascending=False)
fpm_result

# %%
```

```
printMinMaxSupport(fpm_result)

# %%
plt.xlabel('Количество попаданий товара в транзакцию')
data.sum().nlargest(10).sort_values().plot.barh()

# %%
plt.xlabel('Уровень поддержки')
fpg_result.set_index('itemsets')['support'].nlargest(10).sort_values().plot.barh()

# %%
plt.xlabel('Уровень поддержки')
fpm_result.set_index('itemsets')['support'].nlargest(10).sort_values().plot.barh()

# %%
items = ['whole milk', 'yogurt', 'soda', 'tropical fruit', 'shopping bags', 'sausage'
, 'whipped/sour cream', 'rolls/buns', 'other vegetables', 'root vegetables', 'pork',
'bottled water', 'pastry', 'citrus fruit', 'canned beer', 'bottled beer']
np_data_new = all_data.to_numpy()
np_data_new = [[elem for elem in row[1:] if isinstance(elem,str) and elem in items] f
or row in np_data_new]

# %%
te_new = TransactionEncoder()
te_ary_new = te_new.fit_transform(np_data_new)
data_new = pd.DataFrame(te_ary_new, columns=te_new.columns_)
data_new

# %%
fpg_result_new = fpgrowth(data_new, min_support=0.03, use_colnames = True).sort_value
s('support', ascending=False)
fpg_result_new

# %%
fpm_result_new = fpmax(data_new, min_support=0.03, use_colnames = True).sort_values('
support', ascending=False)
fpm_result_new

# %%
printMinMaxSupport(fpg_result_new)
printMinMaxSupport(fpm_result_new)

# %%
min_supports = np.arange(0.0, 1, 0.01)
sup_data = []

for min_support in np.logspace(-3, 0, num=20):
    results = fpgrowth(data, min_support=min_support, use_colnames=True)
    results['length'] = results['itemsets'].apply(lambda x: len(x))
    max_len_curr = np.max(results['length'])
    if (np.isnan(max_len_curr)):
        break
    grouped_count = results.groupby('length').itemsets.count()

    lens_dict = {
        'Общее кол-во': 0,
        'min_support': min_support
    }
    for i in range(1, len(grouped_count) + 1):
        lens_dict.setdefault(f'Набор длины {i}', grouped_count[i])
    lens_dict['Общее кол-во'] = len(results)
    sup_data.append(lens_dict)
```

```
df_count_by_lens = pd.DataFrame(sup_data).fillna(value=0)
fig, ax = plt.subplots(figsize=(12, 8))
df_count_by_lens.plot(ax=ax, x='min_support', logy='sym', logx=True, colormap='rainbo
w')
ax.set_axisbelow(True)
ax.grid(0.6)
ax.set_ylabel('Количество наборов')
ax.set_xlabel('Уровень поддержки')
fig.tight_layout()

# %% [markdown]
# ## Ассоциативные правила

# %%
np_data_a = all_data.to_numpy()
np_data_a = [[elem for elem in row[1:] if isinstance(elem,str) and elem in items] for
 row in np_data_a]
np_data_a = [row for row in np_data_a if len(row) > 1]
te_a = TransactionEncoder()
te_ary_a = te_a.fit_transform(np_data_a)
data_a = pd.DataFrame(te_ary_a, columns=te_a.columns_)
data_a

# %%
result = fpgrowth(data_a, min_support=0.05, use_colnames = True)
result

# %%
rules_conf = association_rules(result, min_threshold = 0.3)
rules_conf

# %%
rules_sup = association_rules(result, min_threshold = 0.01, metric='support')
rules_sup

# %%
rules_lift = association_rules(result, min_threshold = 0.01, metric='lift')
rules_lift

# %%
rules_leverage = association_rules(result, min_threshold = 0.01, metric='leverage')
rules_leverage

# %%
rules_conviction = association_rules(result, min_threshold = 0.01, metric='conviction
')
rules_conviction

# %%
def getStatistic(rules, metric):
    return f'mean = {round(rules[metric].mean(), 5)}', f'median = {round(rules[metric
].median(), 5)}', f'std = {round(rules[metric].std(), 5)}'

# %%
print('support', getStatistic(rules_sup, 'support'))
print('confidence', getStatistic(rules_conf, 'confidence'))
print('lift', getStatistic(rules_lift, 'lift'))
print('leverage', getStatistic(rules_leverage, 'leverage'))
print('conviction', getStatistic(rules_conviction, 'conviction'))

# %%
```

```
rules_ = association_rules(result, min_threshold=0.4, metric='confidence')
rules_

# %%
digraph = nx.DiGraph()
for rule in rules_.itertuples(index=False):
    digraph.add_edge(rule.antecedents, rule.consequents, weight=rule.support, label=r
ound(rule.confidence, 3))
plt.figure(figsize=(12, 8))
pos = nx.spring_layout(digraph)
nx.draw(digraph, pos,
    labels={node: '\n'.join(node) for node in digraph.nodes()},
    width=[digraph[u][v]['weight']*30 for u,v in digraph.edges()],
    node_size=2000
)
nx.draw_networkx_edge_labels(digraph, pos, edge_labels=nx.get_edge_attributes(digraph
, 'label'))
plt.axis('off')
plt.show()

# %%
rules_pivot = rules_.pivot(index='antecedents', columns='consequents', values='confid
ence')
rules_pivot.index = ['\n'.join(ind) for ind in rules_pivot.index]
rules_pivot.columns = ['\n'.join(col) for col in rules_pivot.columns]
sns.heatmap(rules_pivot, cmap='rainbow')
plt.tight_layout()
plt.show()
```