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

ОТЧЕТ

по лабораторной работе №4

по дисциплине «Машинное обучение»

Тема: Ассоциативный анализ

Студент гр. 6304 Доброхвалов М. О.

Преподаватель Жангиров Т. Р.

Санкт-Петербург

2020

Цель работы

Ознакомиться с методами ассоциативного 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('groceries - groceries.csv')
```

2. Были переформированы данные, а также удалены все значения NaN

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

3. Был получен список всех уникальных товаров

```
unique_items = set(np.unique(np.concatenate(np_data)))
```

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]
```

5. Список содержит 169 товаров (рис. 1).

```
print(unique_items)
print(len(unique_items))
```

```
print(unique_items)
{'tropical fruit', 'prosecco', 'sweet spreads', 'frozen fish', 'cream cheese', 'curd cheese', 'turkey', 'grapes', 'brandy', 'org
anic products', 'pip fruit', 'whipped/sour cream', 'zwieback', 'cereals', 'liquor (appetizer)', 'salad dressing', 'onions', 'can
dles', 'vinegar', 'flour', 'butter milk', 'skin care', 'soft cheese', 'root vegetables', 'snack products', 'kitchen towels', 'co
ffee', 'roll products', 'specialty cheese', 'liver loaf', 'female sanitary products', 'cat food', 'softener', 'salt', 'canned fi
sh', 'mayonnaise', 'nut snack', 'pudding powder', 'chocolate', 'baking powder', 'bottled water', 'waffles', 'newspapers', 'long
life bakery product', 'cake bar', 'rum', 'popcorn', 'pastry', 'pork', 'pasta', 'yogurt', 'butter', 'dessert', 'specialty bar',
'cocoa drinks', 'flower soil/fertilizer', 'specialty fat', 'pickled vegetables', 'toilet cleaner', 'hygiene articles', 'liquor',
'make up remover', 'oil', 'semi-finished bread', 'frozen dessert', 'cookware', 'chewing gum', 'abrasive cleaner', 'specialty cho
colate', 'whole milk', 'cream', 'salty snack', 'frozen vegetables', 'bottled beer', 'hair spray', 'liqueur', 'frozen fruits', 'p
rocessed cheese', 'frozen meals', 'honey', 'meat spreads', 'baby food', 'soups', 'potted plants', 'dishes', 'domestic eggs', 'ke
tchup', 'spices', 'hard cheese', 'dish cleaner', 'citrus fruit', 'herbs', 'canned vegetables', 'beverages', 'condensed milk', 'p
et care', 'ham', 'cleaner', 'jam', 'tidbits', 'cling film/bags', 'spread cheese', 'fruit/vegetable juice', 'seasonal products',
'syrup', 'bathroom cleaner', 'meat', 'chicken', 'dental care', 'whisky', 'sliced cheese', 'artif. sweetener', 'canned beer', 'ic
e cream', 'organic sausage', 'napkins', 'sound storage medium', 'soda', 'brown bread', 'frozen chicken', 'frankfurter', 'shoppin
g bags', 'frozen potato products', 'misc. beverages', 'bags', 'mustard', 'instant coffee', 'hamburger meat', 'specialty vegetabl
es', 'sugar', 'chocolate marshmallow', 'detergent', 'potato products', 'ready soups', 'rubbing alcohol', 'sauces', 'fish', 'soa
p', 'UHT-milk', 'dog food', 'flower (seeds)', 'other vegetables', 'berries', 'canned fruit', 'curd', 'white bread', 'decalcifie
r', 'photo/film', 'preservation products', 'candy', 'cooking chocolate', 'finished products', 'rice', 'light bulbs', 'nuts/prune
s', 'red/blush wine', 'sparkling wine', 'sausage', 'packaged fruit/vegetables', 'house keeping products', 'white wine', 'tea',
```

Рис. 1 — Список товаров

FPGrowth и FPMax

1. Данные были преобразованы к виду, удобному для анализа

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

2. Был проведен ассоциативный анализ используя алгоритмы FPGrowth и FPMax при уровне поддержки 0.03

```
result_fpgrowth = fpgrowth(data, min_support=0.03, use_colnames
= True)
result_fpgrowth['length'] = np.fromiter(map(len,
result_fpgrowth['itemsets']),dtype=int)
result_fpmax = fpmax(data, min_support=0.03, use_colnames =
True)
result_fpmax['length'] = np.fromiter(map(len,
result_fpmax['itemsets']),dtype=int)
```

3. Были проанализированы получившиеся варианты.

	Количество элементов	FPGrowth	FPMax
Min	1	0.030	0.030
	2	0.030	0.030
Max	1	0.256	0.099
	2	0.075	0.075

4. FP-Max - это вариант FP-Growth, который фокусируется на получении максимальных наборов предметов. Набор элементов X называется максимальным, если X является частым и не существует частого супер-шаблона, содержащего X. Другими словами, частый шаблон X не может быть под-шаблоном более частого шаблона, чтобы соответствовать определению максимального набора элементов.

5. Частота встречаемости товаров пропорционально значению уровня поддержки для конкретного товара (рис. 2).

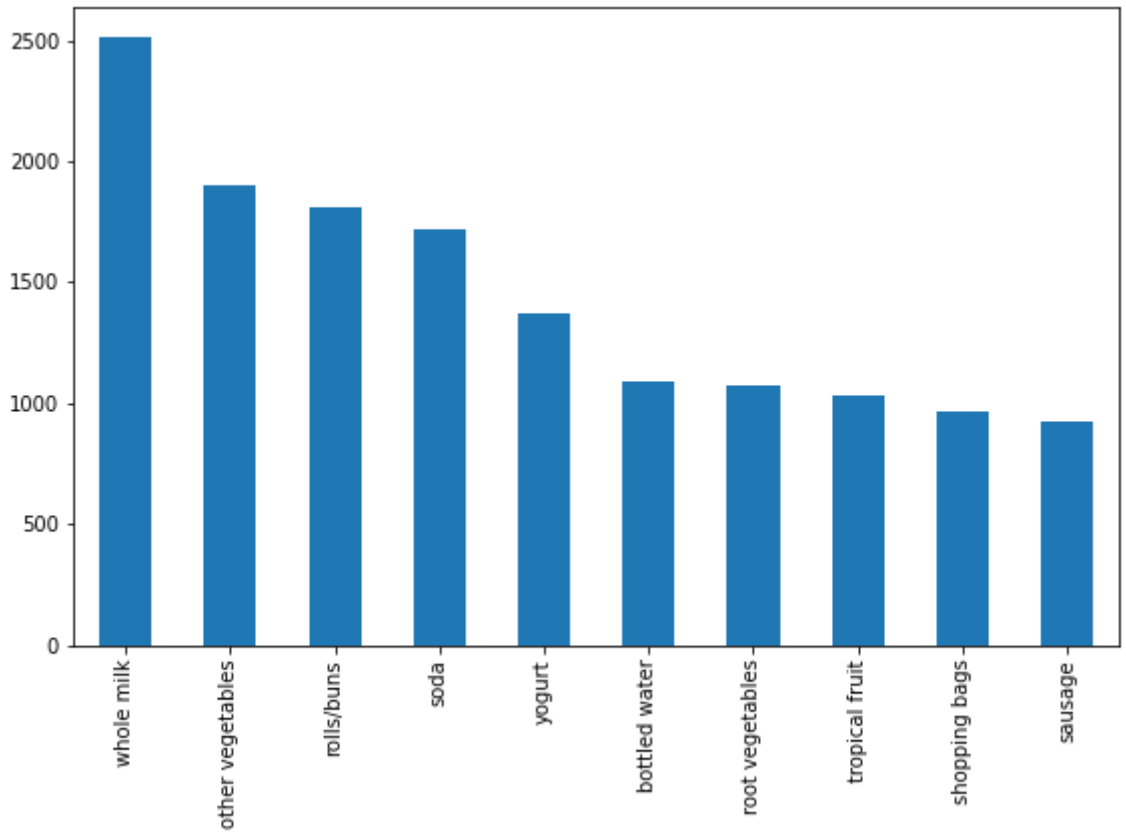

Рис. 2 — 10 самых часто встречающихся товаров

6. Был преобразуем набор данных, чтобы он содержал ограниченный набор товаров

```
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_f = all_data.to_numpy()
np_data_f = [[elem for elem in row[1:] if isinstance(elem,str)
and elem in items] for row in np_data_f]
```

7. Был проведен анализ FPGrowth и FPMax для нового набора данных. Проанализируйте, что изменилось

   Максимальные значения уровня поддержки не изменились. Минимальные значения изменились. Причиной является то, изменились товары. Как следствие, тот товар, уровень значения которого был минимален ранее, теперь удален. Следовательно,

значение стало другим. Значения уровней поддержки товаров, которые остались - не изменилось.

	Количество элементов	FPGrowth	FPMax
Min	1	0.058	0.058
	2	0.031	0.031
Max	1	0.256	0.099
	2	0.075	0.075

8. Было проведено исследование изменение количества получаемых правил от уровня поддержки (рис. 3)

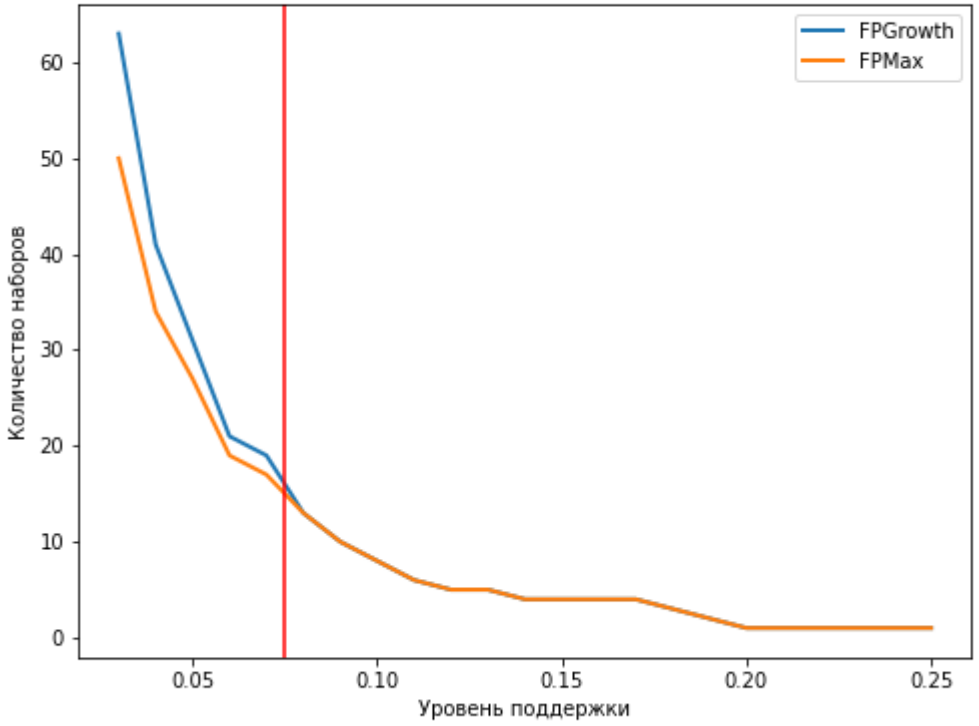

Рис. 3 — Зависимость количества наборов от уровня поддержки

Ассоциативные правила

1. Был сформирован набор данных из определенных товаров и так, чтобы размер транзакции был 2 и более. После чего были получены частоты наборов используя алгоритм FPGrowth.

```
np_data = all_data.to_numpy()
np_data = [[elem for elem in row[1:] if isinstance(elem,str) and
elem in
items] for row in np_data]
np_data = [row for row in np_data if len(row) > 1]
result = fpgrowth(data, min_support=0.05, use_colnames = True)
```

2. Был проведен ассоциативный анализ. По умолчанию расчет проводится на основе метрики *Confidence.*

```
rules = association_rules(result, min_threshold = 0.3) print(rules)
```

Confidence(Уверенность) - вероятность увидеть консеквент в транзакции при условии, что оно также содержит антецедент. Метрика не является симметричной или направленной; например, *Confidence* для $A \rightarrow B$ отличается от достоверности для $B \rightarrow A$. Достоверность равна 1 (максимальная) для правила $A \rightarrow B$, если консеквент и антецедент всегда встречаются вместе.

$$confidence(A \rightarrow B) = \frac{support(AB)}{support(A)} = \frac{rules[support]}{rules[antecedent\ support]}$$

Lift(подъем) - насколько чаще предшествующее и последующее действие правила $A \rightarrow B$ встречается вместе, чем мы ожидали бы, если бы они были статистически независимыми. Если A и B независимы, оценка *Lift* будет ровно 1.

$$lift(A \rightarrow B) = \frac{confidence(A \rightarrow B)}{support(B)} = \frac{rules[confidence]}{rules[consequent\ support]}$$

Leverage(действие рычага) - разница между наблюдаемой частотой появления A и B вместе и частотой, которую можно было бы ожидать, если бы A и B были независимыми. Значение *Leverage* = 0 указывает на независимость.

$$levarage(A \rightarrow B) = support(A \rightarrow B) - support(A) \times support(B) = rules[support] - rules[antecedent\ support] \times rules[consequent\ support]$$

Conviction(убеждение) - насколько консеквент сильно зависит от антецедента. Как и в случае с *Lift*, если предметы независимы, *Conviction* равна 1.

$$conviction(A \to B) = \frac{1 - support(B)}{1 - confidence(A \to B)} = \frac{1 - rules[consequent\ support]}{1 - rules[confidence]}$$

3. Было проведено построение ассоциативных правил для различных метрик. Минимальное значение было выбрано на основе того, чтобы выводилось не менее 10 правил.

```
association_rules_res = association_rules(result_fpgrowth,
metric='confidence', min_threshold = 0.33)
association_rules(result_fpgrowth, metric='lift', min_threshold
= 1.65)
association_rules(result_fpgrowth, metric='leverage',
min_threshold=0.016)
association_rules(result_fpgrowth, metric='conviction',
min_threshold=1.18)
```

4. Были рассчитаны описательные статистики для метрик

```
association_rules_res.iloc[:,2:].describe()
```

	antecedent support	consequent support	support	confidence	lift	leverage	conviction
mean	0,107	0,238	0,042	0,394	1,675	0,017	1,261
std	0,034	0,031	0,014	0,040	0,228	0,006	0,079
min	0,072	0,184	0,030	0,326	1,442	0,009	1,179
25%	0,086	0,225	0,031	0,371	1,535	0,012	1,212
50%	0,105	0,256	0,036	0,398	1,578	0,015	1,236
75%	0,109	0,256	0,048	0,419	1,764	0,021	1,299
max	0,193	0,256	0,075	0,450	2,247	0,026	1,427

5. Был построен граф. Каждая вершина графа отображает набор товаров. Граф ориентирован от антецедента к консеквенту. Ширина ребра отображает уровень support, а подпись на ребре отображает confidence.

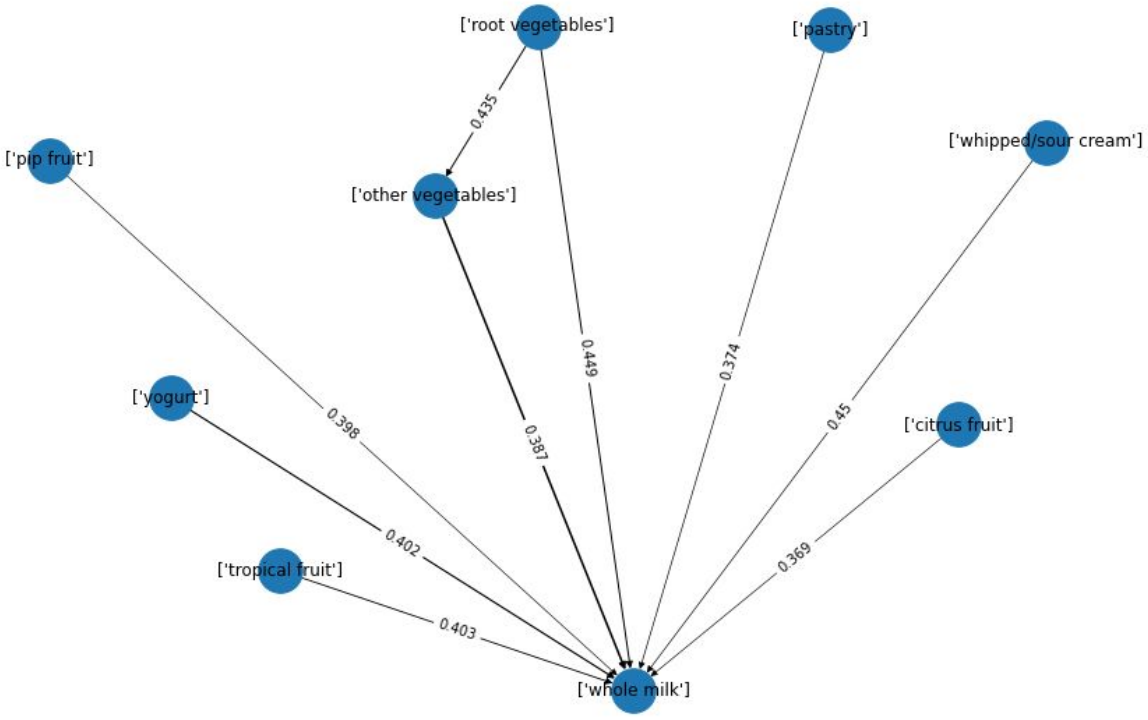

Рис. 4 — Граф набора товаров

6. Из графа можно сделать выводы вроде: если в транзакции есть предметы вроде cirtus fruit, pastry и т.п., то с высокой вероятностью также окажется whole milk.

7. В качестве альтернативы можно использовать heatmap (рис. 5)

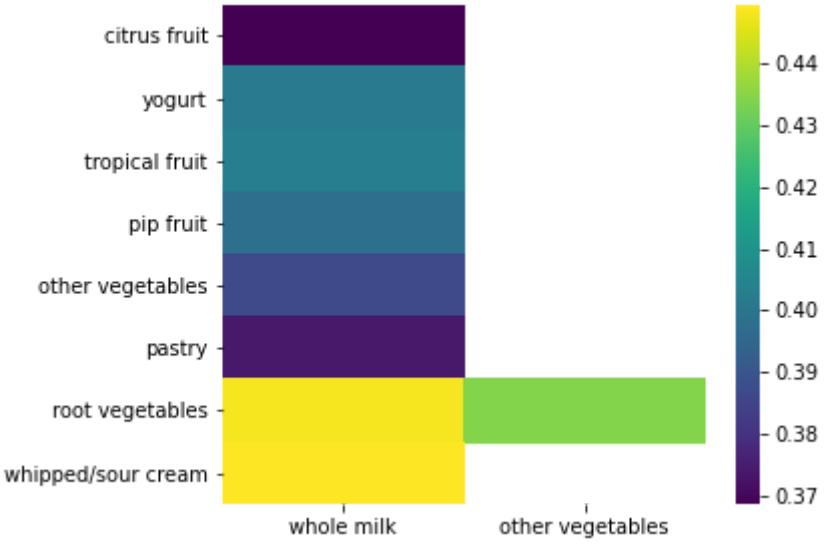

Рис. 4 — Heatmap значений confidence

Вывод

Были изучены методы ассоциативного анализа из библиотеки MLxtend. Рассмотренными алгоритмами является FPGrowth, FPMax, а также построение ассоциативных правил с помощью association_rules. Возможными вариантами применения этих алгоритмов является построение рекомендаций.