**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 _____ Иванов Д.В.

Преподаватель _____ Жангиров Т. Р.

Санкт-Петербург

2020

**Цель работы**

Ознакомиться с методами ассоциативного анализа из библиотеки MLxtend

**Загрузка данных**

1. Датасет скачан и загружен в датафрейм.

```
import pandas as pd
import numpy as np
all_data = pd.read_csv('groceries - groceries.csv')
```

2. Данные переформированы, а также из них 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()
for row in np_data:
    for elem in row:
        unique_items.add(elem)
```

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))
```

```
{'processed cheese', 'specialty fat', 'coffee', 'red/blush wine', 'baby cosmetics', 'soda', 'rubbing alcohol', 'canned b
eer', 'spices', 'hair spray', 'bathroom cleaner', 'soups', 'rolls/buns', 'instant coffee', 'canned vegetables', 'salty s
nack', 'chicken', 'salt', 'condensed milk', 'house keeping products', 'UHT-milk', 'ham', 'fruit/vegetable juice', 'hard
cheese', 'berries', 'flower (seeds)', 'snack products', 'butter', 'frozen chicken', 'sausage', 'kitchen utensil', 'jam',
'margarine', 'roll products', 'sauces', 'specialty vegetables', 'toilet cleaner', 'misc. beverages', 'flour', 'curd chee
se', 'candy', 'frozen meals', 'dog food', 'butter milk', 'pet care', 'cling film/bags', 'preservation products', 'oil',
'vinegar', 'syrup', 'pork', 'popcorn', 'specialty bar', 'soap', 'dental care', 'candles', 'organic sausage', 'fish', 'ho
ney', 'long life bakery product', 'mustard', 'detergent', 'newspapers', 'whipped/sour cream', 'cooking chocolate', 'liqu
eur', 'canned fish', 'softener', 'packaged fruit/vegetables', 'sound storage medium', 'photo/film', 'specialty cheese',
'canned fruit', 'curd', 'frozen vegetables', 'beverages', 'cream cheese', 'semi-finished bread', 'shopping bags', 'past
a', 'dishes', 'spread cheese', 'tropical fruit', 'citrus fruit', 'finished products', 'cereals', 'female sanitary produc
ts', 'pastry', 'cat food', 'frozen dessert', 'chewing gum', 'whole milk', 'other vegetables', 'nut snack', 'ketchup', 'l
iquor (appetizer)', 'abrasive cleaner', 'grapes', 'specialty chocolate', 'frankfurter', 'yogurt', 'sparkling wine', 'coo
kware', 'frozen fish', 'bags', 'bottled water', 'waffles', 'napkins', 'organic products', 'male cosmetics', 'sliced chee
se', 'cake bar', 'liver loaf', 'tea', 'nuts/prunes', 'mayonnaise', 'soft cheese', 'rice', 'chocolate', 'seasonal product
s', 'dessert', 'flower soil/fertilizer', 'hygiene articles', 'frozen fruits', 'zwieback', 'cleaner', 'cream', 'decalcifi
er', 'domestic eggs', 'meat', 'whisky', 'hamburger meat', 'make up remover', 'chocolate marshmallow', 'prosecco', 'herb
s', 'kitchen towels', 'root vegetables', 'cocoa drinks', 'baby food', 'light bulbs', 'baking powder', 'pickled vegetable
s', 'rum', 'turkey', 'salad dressing', 'skin care', 'brandy', 'pip fruit', 'Instant food products', 'bottled beer', 'tid
bits', 'sweet spreads', 'ready soups', 'dish cleaner', 'meat spreads', 'ice cream', 'white bread', 'liquor', 'brown brea
d', 'beef', 'frozen potato products', 'white wine', 'pudding powder', 'potted plants', 'sugar', 'potato products', 'arti
f. sweetener', 'onions'}
169
```

Рис. 1 — Список товаров

**FPGrowth и FPMax**

1. Данные 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. Проведен ассоциативный анализ с использованием алгоритмов 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. Проанализированы получившиеся результаты.

| Количество элементов | Min/Max значение уровня | FPGrowth | FPMax |
|---|---|---|---|
| 1 | Min | 0.0304 | 0.0304 |
| | Max | 0.2555 | 0.0985 |
| 2 | Min | 0.0300 | 0.0300 |
| | Max | 0.0748 | 0.0748 |

4. FP-Max – это вариант FP-Growth, фокусирующийся на получении максимальных наборов предметов. Набор элементов X называется максимальным, если X является частым и не существует частого супер-шаблона, содержащего X. Т.е. 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).

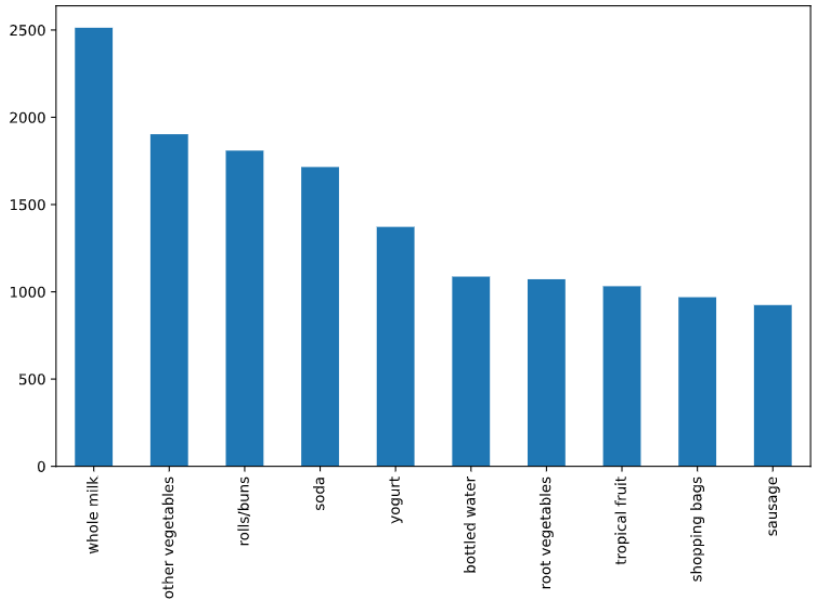


Рис. 2 — 10 самых часто встречающихся товаров

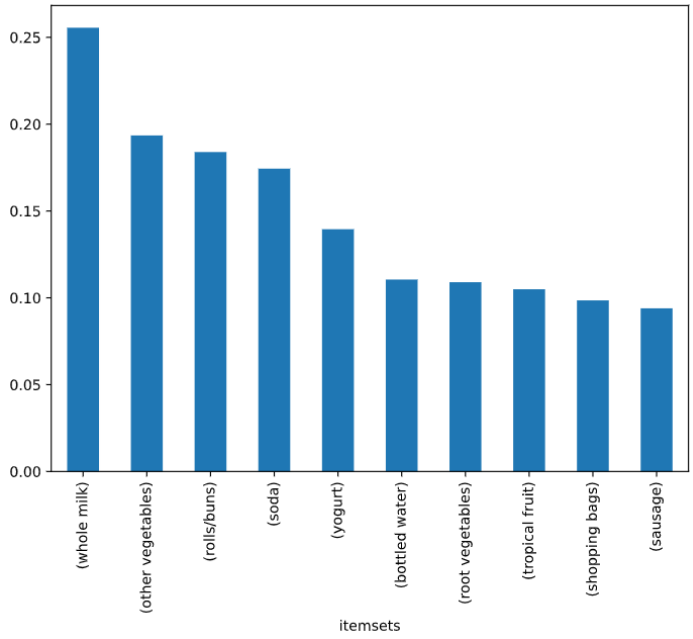


Рис. 3 — 10 наборов с максимальным уровнем поддержки

6. Набор данных преобразован так, 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. Проведен анализ FPGrowth и FPMax для нового набора данных. 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 | Min/Max значение уровня | FPGrowth | FPMax |
|---|---|---|---|
| 1 | Min | 0.0576 | 0.0576 |
| | Max | 0.2555 | 0.0985 |
| 2 | Min | 0.0305 | 0.0305 |
| | Max | 0.0748 | 0.0748 |

8. Исследовано изменение количества получаемых правил от уровня поддержки (рис. 4)

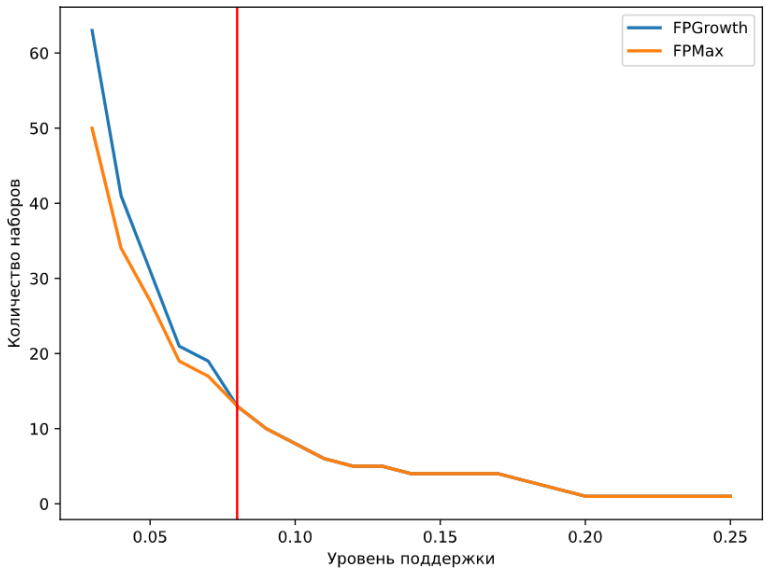


Рис. 4 — Зависимость количества наборов от уровня поддержки

**Ассоциативные правила**

1. Сформирован набор данных из определенных товаров, чтобы размер транзакции был 2 и более. После чего получены частоты наборов с использованием алгоритма 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. Проведен ассоциативный анализ, по умолчанию расчет проводится на основе метрики *Confidence*.

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

| | antecedents | consequents | antecedent support | consequent support | support | confidence | lift | leverage | conviction |
|---|---|---|---|---|---|---|---|---|---|
| 0 | (yogurt) | (whole milk) | 0.139502 | 0.255516 | 0.056024 | 0.401603 | 1.571735 | 0.020379 | 1.244132 |
| 1 | (other vegetables) | (whole milk) | 0.193493 | 0.255516 | 0.074835 | 0.386758 | 1.513634 | 0.025394 | 1.214013 |
| 2 | (rolls/buns) | (whole milk) | 0.183935 | 0.255516 | 0.056634 | 0.307905 | 1.205032 | 0.009636 | 1.075696 |

Рис. 5 — Результаты ассоциативного анализа

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

$$\text{confidence}(\text{A} \rightarrow \text{B}) = \frac{\text{support}(\text{A} \rightarrow \text{B})}{\text{support}(\text{A})}, \text{range: } [0,1]$$

*Lift (Подъем)* – насколько чаще предшествующее и последующее действие правила $A \rightarrow B$ встречается вместе, чем ожидалось, если бы они были статистически независимыми. Если $A$ и $B$ независимы, оценка *Lift* будет равно 1.

$$\text{lift}(\text{A} \rightarrow \text{B}) = \frac{\text{confidence}(\text{A} \rightarrow \text{B})}{\text{support}(\text{B})}, \text{range: } [0, \infty]$$

*Leverage(Рычаг)* – 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.

$$\text{levarage}(\text{A} \rightarrow \text{B}) = \text{support}(\text{A} \rightarrow \text{B}) - \text{support}(\text{A}) \times \text{support}(\text{B}),$$

$$\text{range: } [-1,1]$$

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

$$\text{conviction}(\text{A} \rightarrow \text{B}) = 1 - \frac{\text{support}(\text{B})}{1 - \text{confidence}(\text{A} \rightarrow \text{B})},$$

$$\text{range: } [0, \infty]$$

3. Проведено построение ассоциативных правил для различных метрик. Значение *min_threshold* выбрано на основе того, чтобы выводилось не менее 10 правил.

```
    association_rules_res = association_rules(result_fpgrowth,
metric='confidence', min_threshold = 0.34)
    association_rules(result_fpgrowth, metric='lift', min_threshold =
1.75)
    association_rules(result_fpgrowth, metric='leverage',
min_threshold=0.016)
    association_rules(result_fpgrowth, metric='conviction',
min_threshold=1.18)
```

4. Рассчитаны описательные статистики для метрик

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

| | antecedent support | consequent support | support | confidence | lift | leverage | conviction |
|---|---|---|---|---|---|---|---|
| **count** | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| **mean** | 0.1079 | 0.2431 | 0.0431 | 0.4006 | 1.6655 | 0.0168 | 1.2665 |
| **std** | 0.0360 | 0.0261 | 0.0142 | 0.0353 | 0.2374 | 0.0061 | 0.0816 |
| **min** | 0.0716 | 0.1934 | 0.0300 | 0.3420 | 1.4423 | 0.0093 | 1.1790 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **25%** | 0.0843 | 0.2555 | 0.0324 | 0.3769 | 1.5244 | 0.0115 | 1.2169 |
| **50%** | 0.1049 | 0.2555 | 0.0390 | 0.3997 | 1.5746 | 0.0155 | 1.2402 |
| **75%** | 0.1089 | 0.2555 | 0.0485 | 0.4268 | 1.7588 | 0.0208 | 1.3246 |
| max | 0.1934 | 0.2555 | 0.0748 | 0.4496 | 2.2466 | 0.0262 | 1.4266 |

5. Построен граф для следующего анализа

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

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

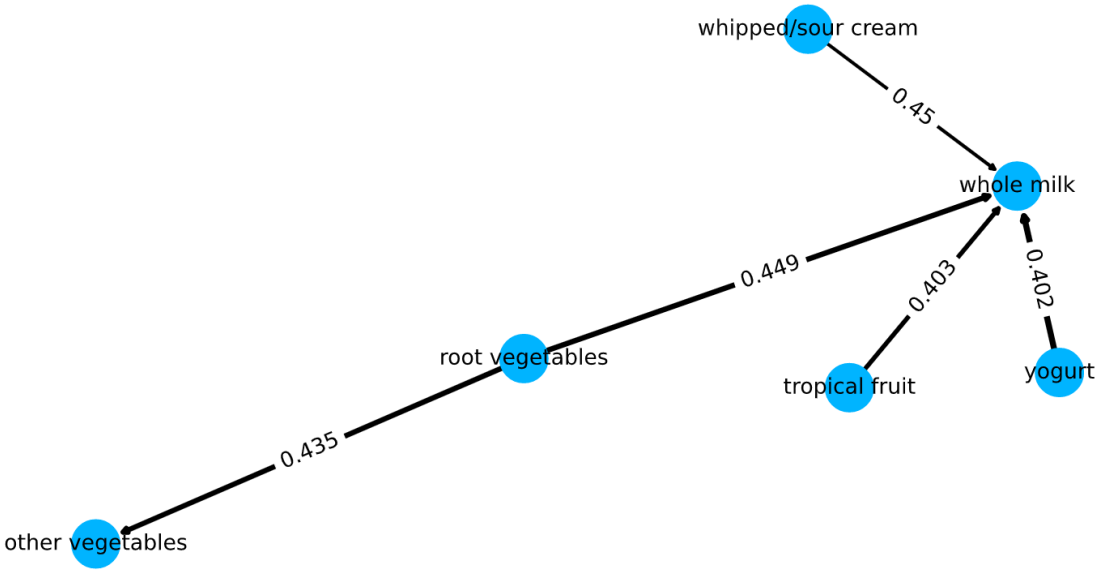


Рис. 6 — Граф набора товаров

6. Из графа можно сделать выводы, что если в транзакции есть предметы вроде *tropical fruit*, *yogurt* и т.п., то с высокой вероятностью в транзакции будет присутствовать *whole milk*; а если есть *root vegetables* – *other vegetables*.

7. Альтернативные способы отображения правил:

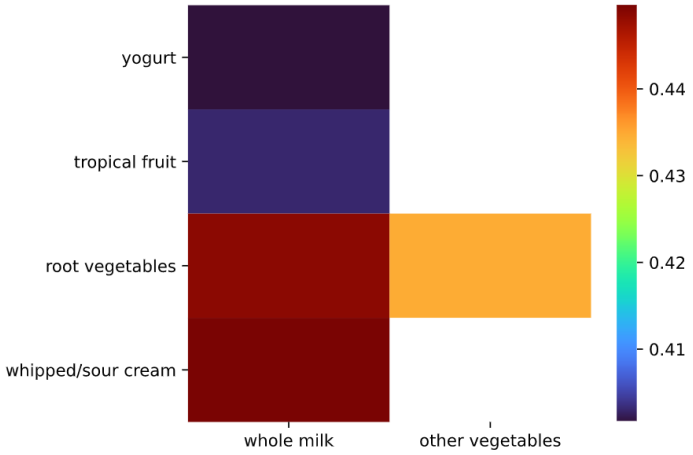


Рис. 7 — Heatmap значений confidence

| | whole milk | other vegetables |
|---|---|---|
| yogurt | 0.401603 | NaN |
| tropical fruit | 0.403101 | NaN |
| root vegetables | 0.448694 | 0.434701 |
| whipped/sour cream | 0.449645 | NaN |

Рис. 8 — Текстовое представление датафрейма правил

**Вывод**

Изучены методы ассоциативного анализа из библиотеки MLxtend. Рассмотрены алгоритмы FPGrowth и FPMax, а также изучено построение ассоциативных правил с помощью *association_rules*. Возможными вариантами применения этих алгоритмов является построение рекомендаций.