Лабораторная работа №3

Частотный анализ

Цель:

Ознакомиться с методами частотного анализа из библиотеки MLxtend

Выполнение:

Для выполнения лаб. работы необходимо установить библиотеку MLxtend

Для этого в терминале введите:

```
pip install mlxtend
```

Загрузка данных

1. Загрузить датасет по ссылке: https://www.kaggle.com/acostasg/random-shopping-cart . Данные представлены в виде csv таблицы. Данные представляют собой информацию о том, какой покупатель что и когда покупал. В данной лаб. работе не будем использовать данные о дате покупки.
2. Создать Python скрипт. Загрузить данные в датафрейм.

```
import pandas as pd
import numpy as np

all_data = pd.read_csv('dataset_group.csv',header=None)
#В файле нет строки с названием столбцов, поэтому параметр header равен
None.
#Интерес представляет информация об id покупателя - столбец с названием 1
#Название купленного товара хранится в столбце с названием 2
```

3. Получим список всех id покупателей, которые есть в файле.

```
unique_id = list(set(all_data[1]))
print(len(unique_id)) #Выведем количество id
```

4. Получим список всех товаров, которые есть в файле.

```
items = list(set(all_data[2]))
print(len(items)) #Выведем количество товаров
```

5. Далее необходимо сформировать датасет подходящий для частотного анализа. Для этого надо слить все товары одного покупателя в один список. Для дальнейшего частотного анализа id покупателя будет не нужен

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

Подготовка данных

1. Так как полученные датасет не пригоден для анализа напрямую, так как каждый список пользователя может содержать разное количество товаров. Поэтому данные надо закодировать так, чтобы их можно было представить в виде матрицы. Для кодированния данных используем TransactionEncoder

```
from mlxtend.preprocessing import TransactionEncoder

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

2. Выведите полученный dataframe и объясните, как стали представляться данные

```
print(df)
```

Ассоциативный анализ с использованием алгоритма Apriori

1. Применим алгоритм apriori с минимальным уровнем поддержки 0.3

```
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)) #добавление
размера набора
print(results)
```

   Объясните полученный результат

2. Применим алгоритм apriori с тем же уровнем поддержки, но ограничим максимальный размер набора единицей

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

3. Применим алгоритм apriori и выведем только те наборы, которые имеют размер 2, а также количество таких наборов

```
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))
```

4. Посчитайте количество наборов при различных уровнях поддержки. Начальное значение поддержки 0.05, шаг 0.01. Постройте график зависимости количества наборов от уровня поддержки
5. Определите значение уровня поддержки при котором перестают генерироваться наборы размера 1,2,3, и.т.д. Отметьте полученные уровне поддержки на графике построенном в пункте 4
6. Построим датасет только из тех элементов, которые попадают в наборы размером 1 при уровне поддержки 0.38

```
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]
```

7. Приведите полученный датасет к формату, который можно обработать
8. Проведите ассоциативный анализ при уровне поддержки 0.3 для нового датасета. Опишите в чем сходства и различия
9. Проведите ассоциативный анализ при уровне поддержки 0.15 для нового датасета. Выведите все наборы размер которых больше 1 и в котором есть 'yogurt' или 'waffles'
10. Постройте датасет, из тех элементов, которые не попали в датасет в п. 6 и приведите его к удобному для анализа виду
11. Проведите анализ aprioti для полученного датасета
12. Напишите правило, для вывода всех наборов, в которых хотя бы два элемента начинаются на 's'
13. Напишите правило, для вывода всех наборов, для которых уровень поддержки изменяется от 0.1 до 0.25