Лабораторная работ 6.
Ассоциативный анализ.

Цель: ознакомиться с методами FPGrowth и FPmax ассоциативного анализа библиотеки MLxtend.
Задачи:

1. Загрузить пред обработанные данные;
2. Переформировать новые данные и преобразовать к формату;
3. Провести ассоциативный анализ;
4. Визуализировать результаты в виде графа.

Пример выполнения лабораторной работы.

Выведем первые пять строк из подготовленного датасета.

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import fpgrowth
from math import isnan
df = pd.read_csv('lab6.csv',encoding='cp1251')
print(df.head(5))
```

Результат:

	id	gender	age	income	product
0	1	Male	19	15	cigarettes
1	2	Male	21	15	milk
2	3	Female	20	16	chocolate
3	4	Female	23	16	milk
4	5	Female	31	17	bread

В нашем случае нас интересуют столбей с продуктами. Переформеруем данные.

```
df = pd.read_csv('lab6.csv',encoding='cp1251')
np_df = df.to_numpy()
np_df = [[elem for elem in row[4:] if isinstance(elem,str)] for row in
np_df]
print(np_df)
```

Результат:

```
[['cigarettes'], ['milk'], ['chocolate'], ['milk'], ['bread'], ['bread'], ['milk'], ['water '], ['water '], ['bread'], ['wate
r '], ['water '], ['chocolate'], ['milk'], ['water '], ['oil'], ['chocolate'], ['oil'], ['chocolate'], ['chocolate'], ['cigar
ettes'], ['flakes '], ['cigarettes'], ['tangerines'], ['tangerines'], ['cigarettes'], ['tangerines'], ['cigarettes'], ['water
'], ['flakes '], ['tangerines'], ['cigarettes'], ['cigarettes'], ['milk'], ['chocolate'], ['milk'], ['bread'], ['bread'], ['m
ilk'], ['water '], ['water '], ['bread'], ['water '], ['water '], ['chocolate'], ['milk'], ['water '], ['oil'], ['chocolate']
, ['oil'], ['chocolate'], ['chocolate'], ['cigarettes'], ['flakes '], ['cigarettes'], ['tangerines'], ['tangerines'], ['cigar
ettes'], ['tangerines'], ['cigarettes'], ['water '], ['flakes '], ['tangerines'], ['cigarettes'], ['cigarettes'], ['milk'], [
'chocolate'], ['milk'], ['bread'], ['bread'], ['milk'], ['water '], ['water '], ['bread'], ['water '], ['water '], ['chocolat
e'], ['milk'], ['water '], ['oil'], ['chocolate'], ['oil'], ['chocolate'], ['chocolate'], ['cigarettes'], ['flakes '], ['ciga
rettes'], ['tangerines'], ['tangerines'], ['cigarettes'], ['tangerines'], ['cigarettes'], ['water '], ['flakes '], ['tangerin
es'], ['cigarettes'], ['cigarettes'], ['milk'], ['chocolate'], ['milk'], ['bread'], ['bread'], ['milk'], ['water '], ['water
'], ['bread'], ['water '], ['water '], ['chocolate'], ['milk'], ['water '], ['oil'], ['chocolate'], ['oil'], ['chocolate'], [
'chocolate'], ['cigarettes'], ['flakes '], ['cigarettes'], ['tangerines'], ['tangerines'], ['cigarettes'], ['tangerines'], ['
cigarettes'], ['water '], ['flakes '], ['tangerines'], ['cigarettes'], ['cigarettes'], ['milk'], ['chocolate'], ['milk'], ['b
read'], ['bread'], ['milk'], ['water '], ['water '], ['bread'], ['water '], ['water '], ['chocolate'], ['milk'], ['water '],
['oil'], ['chocolate'], ['oil'], ['chocolate'], ['chocolate'], ['cigarettes'], ['flakes '], ['cigarettes'], ['tangerines'], [
'tangerines'], ['cigarettes'], ['tangerines'], ['cigarettes'], ['water '], ['flakes '], ['tangerines'], ['cigarettes'], ['cig
arettes'], ['milk'], ['chocolate'], ['milk'], ['bread'], ['bread'], ['milk'], ['water '], ['water '], ['bread'], ['water '],
['water '], ['chocolate'], ['milk'], ['water '], ['oil'], ['chocolate'], ['oil'], ['chocolate'], ['chocolate'], ['cigarettes'
], ['flakes '], ['cigarettes'], ['tangerines'], ['tangerines'], ['cigarettes'], ['tangerines'], ['cigarettes'], ['water '], [
'flakes '], ['tangerines'], ['cigarettes'], ['cigarettes'], ['milk'], ['chocolate'], ['milk'], ['bread'], ['bread']]
```

Выведем уникальные названия продуктов.

```
df = pd.read_csv('lab5.csv',encoding='cp1251')
np_df = df.to_numpy()
np_df = [[elem for elem in row[4:] if isinstance(elem,str)] for row in
np_df]
unique_items = set()
for row in np_df:
    for elem in row:
        unique_items.add(elem)
print(unique_items)
```

Результат:

```
{'bread', 'milk', 'chocolate', 'water ', 'oil', 'flakes ', 'tangerines', 'cigarettes'}
```

И преобразуем данные к нужному формату для дальнейшего анализа.

```
te = TransactionEncoder()
te_ary = te.fit(np_df).transform(np_df)
df_new = pd.DataFrame(te_ary, columns=te.columns_)
print(df_new)
```

Результат:

	bread	chocolate	cigarettes	flakes	milk	oil	tangerines	water
0	False	False	True	False	False	False	False	False
1	False	False	False	False	True	False	False	False

Используем алгоритм FPGrowth при уровне поддержки 0.03 и проведем ассоциативный анализ.

```
te = TransactionEncoder()
te_ary = te.fit(np_df).transform(np_df)
df_new = pd.DataFrame(te_ary, columns=te.columns_)
fpg = fpgrowth(df_new, min_support=0.03, use_colnames = True)
print(fpg)
```

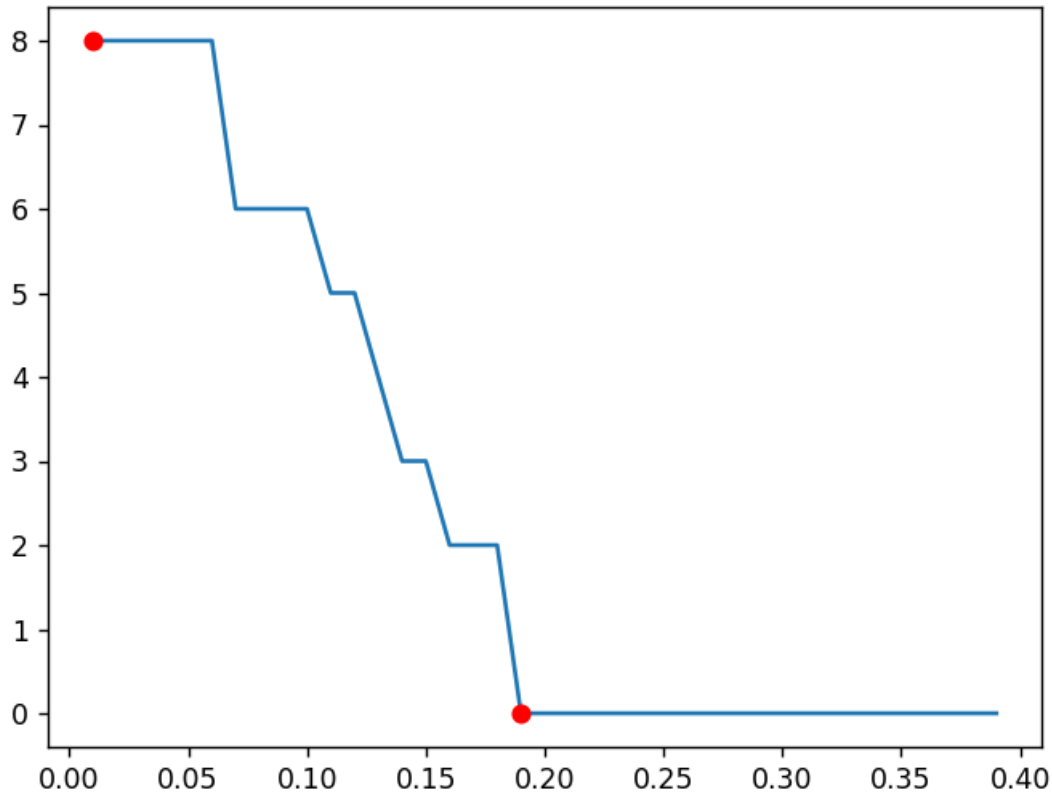
Результат:

	support	itemsets
0	0.186869	(cigarettes)
1	0.131313	(milk)
2	0.156566	(chocolate)
3	0.101010	(bread)
4	0.181818	(water)
5	0.060606	(oil)
6	0.060606	(flakes)
7	0.121212	(tangerines)

Определим минимальное и максимальное значения для уровня поддержки и визуализируем получившийся результат.

```
min_support_range = np.arange(0.01, 0.4, 0.01)
itemsets_lengths = []
threshold_supports = []
threshold_lengths = []
last_itemset_len = len(df_new.columns)
for min_support in min_support_range:
    fpg = fpgrowth(df_new, min_support=min_support, use_colnames=True)
    itemsets_lengths.append(len(fpg))
    fpg['length'] = fpg['itemsets'].apply(lambda x: len(x))
    current_itemset_max_len = fpg['length' ].max()
    if isnan(current_itemset_max_len):
        current_itemset_max_len = 0
    if current_itemset_max_len < last_itemset_len:
        last_itemset_len = current_itemset_max_len
        threshold_supports.append(min_support)
        threshold_lengths.append(len(fpg))
plt.figure()
plt.plot(min_support_range.tolist(), itemsets_lengths)
plt.plot(threshold_supports, threshold_lengths, 'ro')
plt.show()
```

Результат:



В ходе лабораторной работы мы провели ассоциативный анализ с помощью методов FPGrowth и FPmax в результате ….

Задание.

1. Загрузить пред обработанные данные, вывести первые 5 строк из датасета;
2. Переформировать нужную колонку;
3. Вывести уникальное количество значений;
4. Преобразовать данные к формату воспользовавшись TransactionEncoder;
5. Провести ассоциативный анализ с помощью алгоритмов FPGrowth и FPmax.
6. Визуализировать результаты двух методов.
7. Описать разницу методов и результат.

Формат отчета.

Протокол лабораторной работы в формате PDF, который должен содержать, поэтапное выполнение всех задач с текстовым описанием ваших действий и экранными формами, отображающими данные действия.