МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Машинное обучение»

Тема: Понижение размерности пространства признаков

Студент гр. 6307 _____ Михайлов И. Т.

Преподаватель _____ Жангиров Т. Р.

Санкт-Петербург

2020

Цель работы:

Ознакомиться с методами понижения размерности данных из библиотеки Scikit Learn.

Ход выполнения работы:

Загружены данные из датасета “dataset_group.csv”.

Список всех id покупателей, которые есть в файле:

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

```
1139
```

Список всех товаров, которые есть в файле:

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

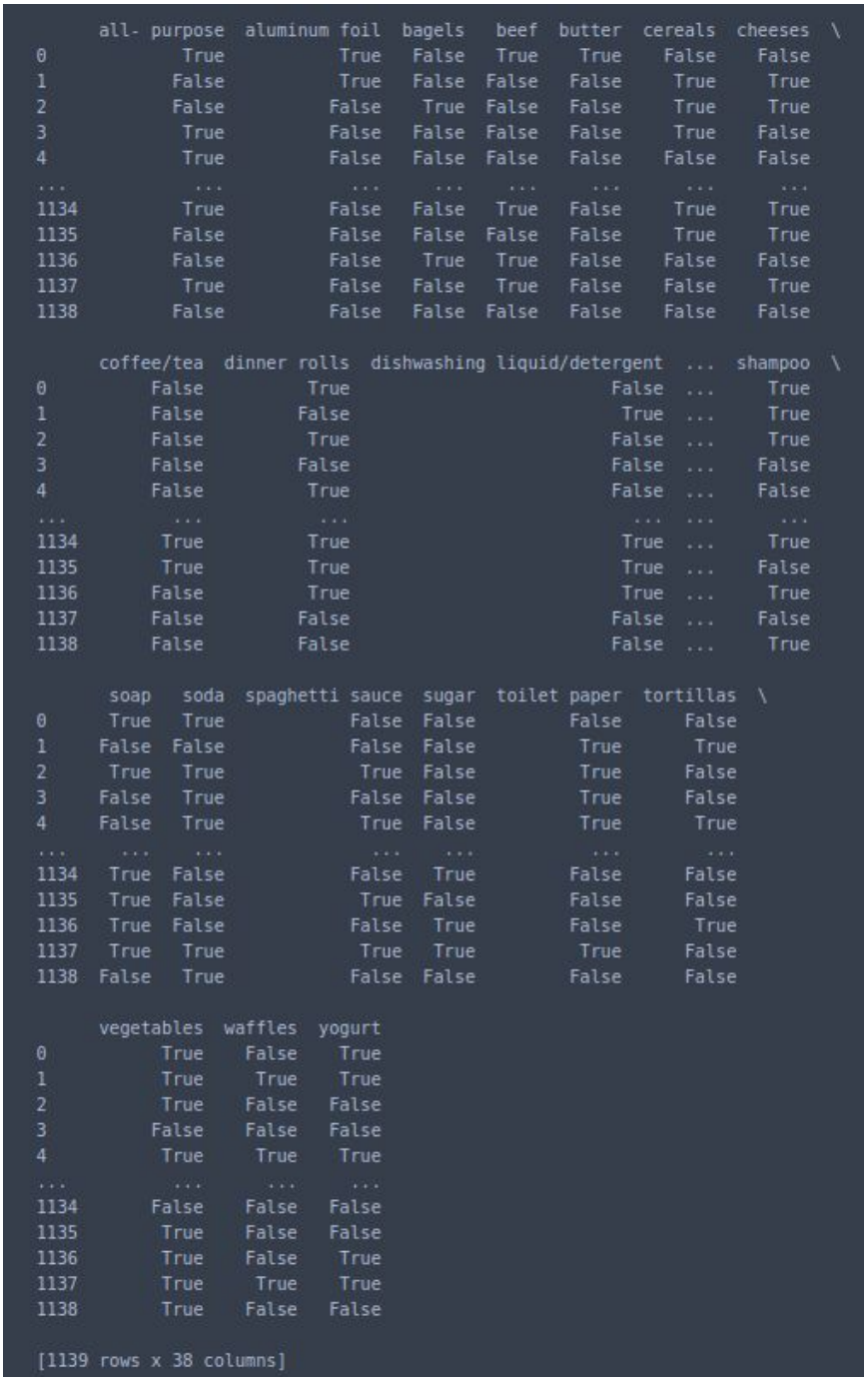
```
38
```

Создан 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]
```

Подготовка данных

Результат кодирования данных с помощью TransationEncoder:



	all- purpose	aluminum foil	bagels	beef	butter	cereals	cheeses	\
0	True	True	False	True	True	False	False	
1	False	True	False	False	False	True	True	
2	False	False	True	False	False	True	True	
3	True	False	False	False	False	True	False	
4	True	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	
1134	True	False	False	True	False	True	True	
1135	False	False	False	False	False	True	True	
1136	False	False	True	True	False	False	False	
1137	True	False	False	True	False	False	True	
1138	False	False	False	False	False	False	False	

	coffee/tea	dinner rolls	dishwashing liquid/detergent	...	shampoo	\
0	False	True	False	...	True	
1	False	False	True	...	True	
2	False	True	False	...	True	
3	False	False	False	...	False	
4	False	True	False	...	False	
...	...	...	...	...	...	
1134	True	True	True	...	True	
1135	True	True	True	...	False	
1136	False	True	True	...	True	
1137	False	False	False	...	False	
1138	False	False	False	...	True	

	soap	soda	spaghetti sauce	sugar	toilet paper	tortillas	\
0	True	True	False	False	False	False	
1	False	False	False	False	True	True	
2	True	True	True	False	True	False	
3	False	True	False	False	True	False	
4	False	True	True	False	True	True	
...	...	...	...	...	...	...	
1134	True	False	False	True	False	False	
1135	True	False	True	False	False	False	
1136	True	False	False	True	False	True	
1137	True	True	True	True	True	False	
1138	False	True	False	False	False	False	

	vegetables	waffles	yogurt
0	True	False	True
1	True	True	True
2	True	False	False
3	False	False	False
4	True	True	True
...	...	...	...
1134	False	False	False
1135	True	False	False
1136	True	False	True
1137	True	True	True
1138	True	False	False

[1139 rows x 38 columns]

Данные представлены в виде матрицы, таким образом, что для каждого id покупателя указано приобретал ли он товар (True/False).

Ассоциативный анализ с использованием алгоритма Apriori

1. Применим алгоритм Априори:

	support	itemsets	length
0	0.374890	(all- purpose)	1
1	0.384548	(aluminum foil)	1
2	0.385426	(bagels)	1
3	0.374890	(beef)	1
4	0.367867	(butter)	1
5	0.395961	(cereals)	1
6	0.390694	(cheeses)	1
7	0.379280	(coffee/tea)	1
8	0.388938	(dinner rolls)	1
9	0.388060	(dishwashing liquid/detergent)	1
10	0.389816	(eggs)	1
11	0.352941	(flour)	1
12	0.370500	(fruits)	1
13	0.345917	(hand soap)	1
14	0.398595	(ice cream)	1
15	0.375768	(individual meals)	1
16	0.376646	(juice)	1
17	0.371378	(ketchup)	1
18	0.378402	(laundry detergent)	1
19	0.395083	(lunch meat)	1
20	0.380158	(milk)	1
21	0.375768	(mixes)	1
22	0.362599	(paper towels)	1
23	0.371378	(pasta)	1
24	0.355575	(pork)	1
25	0.421422	(poultry)	1
26	0.367867	(sandwich bags)	1
27	0.349429	(sandwich loaves)	1
28	0.368745	(shampoo)	1
29	0.379280	(soap)	1
30	0.390694	(soda)	1
31	0.373134	(spaghetti sauce)	1
32	0.360843	(sugar)	1
33	0.378402	(toilet paper)	1
34	0.369622	(tortillas)	1
35	0.739245	(vegetables)	1
36	0.394205	(waffles)	1
37	0.384548	(yogurt)	1
38	0.310799	(vegetables, aluminum foil)	2
39	0.300263	(vegetables, bagels)	2
40	0.310799	(vegetables, cereals)	2
41	0.309043	(vegetables, cheeses)	2
42	0.308165	(vegetables, dinner rolls)	2
43	0.306409	(vegetables, dishwashing liquid/detergent)	2
44	0.326602	(eggs, vegetables)	2
45	0.302897	(vegetables, ice cream)	2
46	0.309043	(laundry detergent, vegetables)	2
47	0.311677	(vegetables, lunch meat)	2
48	0.331870	(vegetables, poultry)	2
49	0.305531	(soda, vegetables)	2
50	0.315189	(vegetables, waffles)	2
51	0.319579	(vegetables, yogurt)	2

Результаты в которых поддержка меньше 0.3 не попали в итоговую выборку.

2. Применим алгоритм Априори с тем же уровнем поддержки, но выведем только одноэлементные наборы:

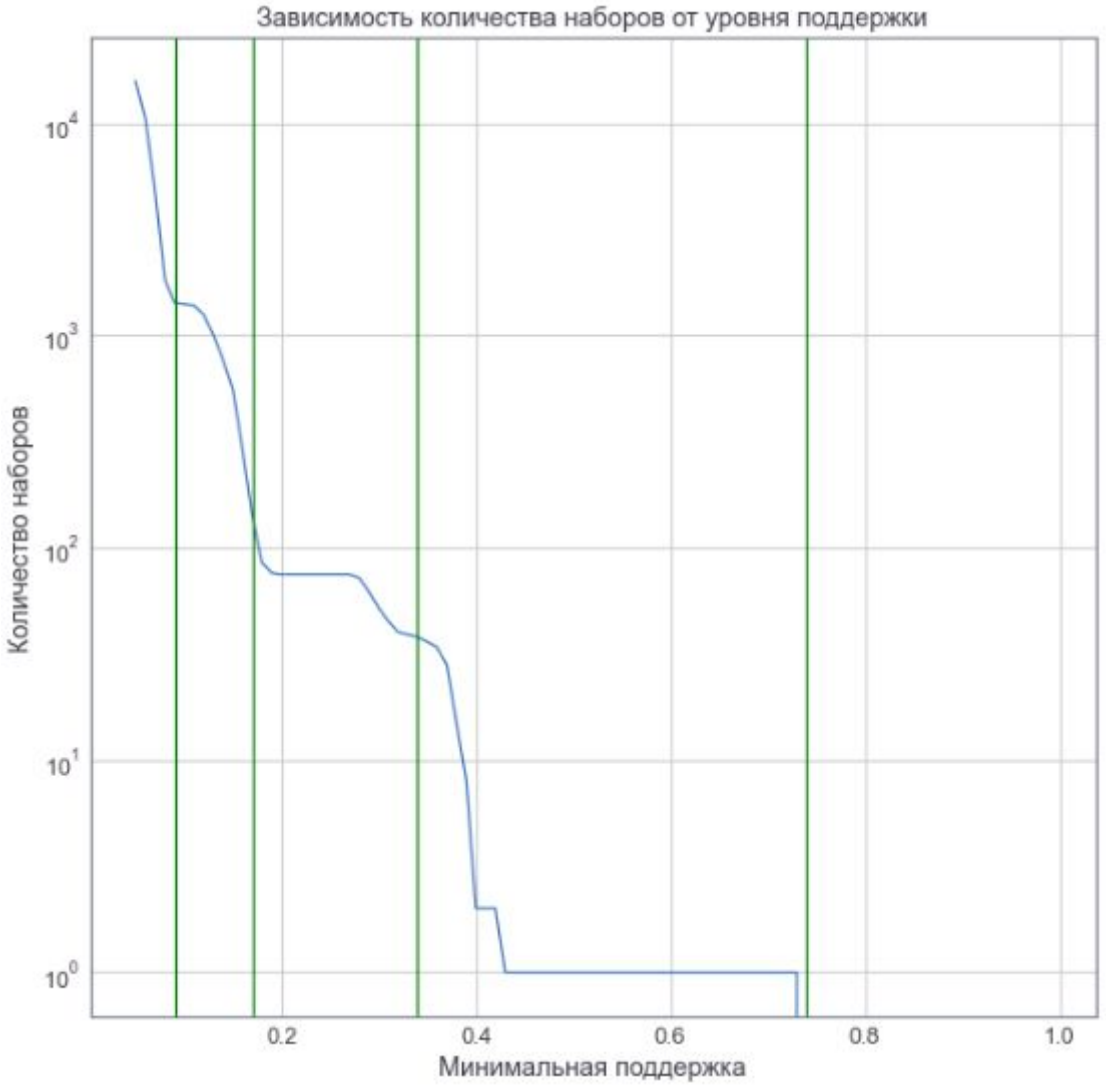
	support	itemsets
0	0.374890	(all- purpose)
1	0.384548	(aluminum foil)
2	0.385426	(bagels)
3	0.374890	(beef)
4	0.367867	(butter)
5	0.395961	(cereals)
6	0.390694	(cheeses)
7	0.379280	(coffee/tea)
8	0.388938	(dinner rolls)
9	0.388060	(dishwashing liquid/detergent)
10	0.389816	(eggs)
11	0.352941	(flour)
12	0.370500	(fruits)
13	0.345917	(hand soap)
14	0.398595	(ice cream)
15	0.375768	(individual meals)
16	0.376646	(juice)
17	0.371378	(ketchup)
18	0.378402	(laundry detergent)
19	0.395083	(lunch meat)
20	0.380158	(milk)
21	0.375768	(mixes)
22	0.362599	(paper towels)
23	0.371378	(pasta)
24	0.355575	(pork)
25	0.421422	(poultry)
26	0.367867	(sandwich bags)
27	0.349429	(sandwich loaves)
28	0.368745	(shampoo)
29	0.379280	(soap)
30	0.390694	(soda)
31	0.373134	(spaghetti sauce)
32	0.360843	(sugar)
33	0.378402	(toilet paper)
34	0.369622	(tortillas)
35	0.739245	(vegetables)
36	0.394205	(waffles)
37	0.384548	(yogurt)

3. Двухэлементные наборы:

	support	itemsets	length
38	0.310799	(vegetables, aluminum foil)	2
39	0.300263	(vegetables, bagels)	2
40	0.310799	(vegetables, cereals)	2
41	0.309043	(vegetables, cheeses)	2
42	0.308165	(vegetables, dinner rolls)	2
43	0.306409	(vegetables, dishwashing liquid/detergent)	2
44	0.326602	(eggs, vegetables)	2
45	0.302897	(vegetables, ice cream)	2
46	0.309043	(laundry detergent, vegetables)	2
47	0.311677	(vegetables, lunch meat)	2
48	0.331870	(vegetables, poultry)	2
49	0.305531	(soda, vegetables)	2
50	0.315189	(vegetables, waffles)	2
51	0.319579	(vegetables, yogurt)	2

Count of result itemstes = 14

4-5. График зависимости количества наборов от уровня поддержки (начальное значение поддержки 0.05, шаг 0.01.):



Зеленым обозначены линии, обозначающие значения уровня поддержки, при которых перестают генерироваться наборы наборы размера 4, 3, 2, 1.

6. 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. Новый датасет приведен к формату, необходимому для использования метода Apriori.

```
te = TransactionEncoder()
te_ary = te.fit(new_dataset).transform(new_dataset)
new_df = pd.DataFrame(te_ary, columns=te.columns_)
```

8. Ассоциативный анализ нового датасета при уровне поддержки 0.3:

	support	itemsets	length
0	0.384548	(aluminum foil)	1
1	0.385426	(bagels)	1
2	0.395961	(cereals)	1
3	0.390694	(cheeses)	1
4	0.388938	(dinner rolls)	1
5	0.388060	(dishwashing liquid/detergent)	1
6	0.389816	(eggs)	1
7	0.398595	(ice cream)	1
8	0.395083	(lunch meat)	1
9	0.380158	(milk)	1
10	0.421422	(poultry)	1
11	0.390694	(soda)	1
12	0.739245	(vegetables)	1
13	0.394205	(waffles)	1
14	0.384548	(yogurt)	1
15	0.310799	(vegetables, aluminum foil)	2
16	0.300263	(vegetables, bagels)	2
17	0.310799	(cereals, vegetables)	2
18	0.309043	(cheeses, vegetables)	2
19	0.308165	(vegetables, dinner rolls)	2
20	0.306409	(vegetables, dishwashing liquid/detergent)	2
21	0.326602	(eggs, vegetables)	2
22	0.302897	(vegetables, ice cream)	2
23	0.311677	(vegetables, lunch meat)	2
24	0.331870	(vegetables, poultry)	2
25	0.305531	(soda, vegetables)	2
26	0.315189	(waffles, vegetables)	2
27	0.319579	(yogurt, vegetables)	2

Отличие этого результата анализа состоит в том, что все наборы длины 1 имеют минимальный уровень поддержки больший 0.38.

9. Ассоциативный анализ при уровне поддержки 0.15 для нового датасета. Все наборы размер которых больше 1 и в котором есть 'yogurt' или 'waffles':

	support	itemsets	length
27	0.169447	(waffles, aluminum foil)	2
28	0.177349	(yogurt, aluminum foil)	2
40	0.159789	(waffles, bagels)	2
41	0.162423	(yogurt, bagels)	2
52	0.160667	(waffles, cereals)	2
53	0.172081	(yogurt, cereals)	2
63	0.172959	(waffles, cheeses)	2
64	0.172081	(yogurt, cheeses)	2
73	0.169447	(waffles, dinner rolls)	2
74	0.166813	(yogurt, dinner rolls)	2
82	0.175593	(waffles, dishwashing liquid/detergent)	2
83	0.158033	(yogurt, dishwashing liquid/detergent)	2
90	0.169447	(waffles, eggs)	2
91	0.174715	(yogurt, eggs)	2
97	0.172959	(waffles, ice cream)	2
98	0.156277	(yogurt, ice cream)	2
103	0.184372	(waffles, lunch meat)	2
104	0.161545	(yogurt, lunch meat)	2
108	0.167691	(yogurt, milk)	2
111	0.166813	(waffles, poultry)	2
112	0.180860	(yogurt, poultry)	2
114	0.177349	(waffles, soda)	2
115	0.167691	(yogurt, soda)	2
116	0.315189	(waffles, vegetables)	2
117	0.319579	(yogurt, vegetables)	2
118	0.173837	(yogurt, waffles)	2
119	0.152766	(yogurt, vegetables, aluminum foil)	3
128	0.157155	(yogurt, eggs, vegetables)	3
130	0.157155	(waffles, vegetables, lunch meat)	3
131	0.152766	(yogurt, vegetables, poultry)	3

10-11. Анализ элементов, которые не попали в пункт 6.

	support	itemsets	length
0	0.374890	(all- purpose)	1
1	0.374890	(beef)	1
2	0.367867	(butter)	1
3	0.379280	(coffee/tea)	1
4	0.352941	(flour)	1
5	0.370500	(fruits)	1
6	0.345917	(hand soap)	1
7	0.375768	(individual meals)	1
8	0.376646	(juice)	1
9	0.371378	(ketchup)	1
10	0.378402	(laundry detergent)	1
11	0.375768	(mixes)	1
12	0.362599	(paper towels)	1
13	0.371378	(pasta)	1
14	0.355575	(pork)	1
15	0.367867	(sandwich bags)	1
16	0.349429	(sandwich loaves)	1
17	0.368745	(shampoo)	1
18	0.379280	(soap)	1
19	0.373134	(spaghetti sauce)	1
20	0.360843	(sugar)	1
21	0.378402	(toilet paper)	1
22	0.369622	(tortillas)	1

12. Правило для вывода всех наборов, в которых хотя бы два элемента начинаются на ‘s’:

```
res2 = res2[res2['itemsets'].apply(lambda x: len([item for item in x if item.startswith('s')]) >=2 )]
print(res2)
```

	support	itemsets
675	0.137840	(sandwich loaves, sandwich bags)
676	0.146620	(shampoo, sandwich bags)
677	0.158911	(soap, sandwich bags)
678	0.162423	(soda, sandwich bags)
679	0.147498	(spaghetti sauce, sandwich bags)
...	...	...
15722	0.064091	(yogurt, soda, vegetables, sugar)
15729	0.058824	(spaghetti sauce, toilet paper, vegetables, su...
15730	0.050044	(spaghetti sauce, vegetables, tortillas, sugar)
15731	0.057946	(spaghetti sauce, waffles, vegetables, sugar)
15732	0.061457	(yogurt, spaghetti sauce, vegetables, sugar)

[1275 rows x 2 columns]

13. Правило для вывода всех наборов, для которых уровень поддержки изменяется от 0.1 до 0.25:

```
res3 = res3[(res3['support'] <= 0.25) & (res3['support'] >= 0.1)]
print(res3)
```

```
      support                              itemsets  length
38   0.157155        (aluminum foil, all- purpose)       2
39   0.150132               (bagels, all- purpose)       2
40   0.144864                 (beef, all- purpose)       2
41   0.147498              (all- purpose, butter)       2
42   0.151010            (cereals, all- purpose)       2
...       ...                                 ...     ...
1401 0.135206  (waffles, toilet paper, vegetables)       3
1402 0.130817   (yogurt, toilet paper, vegetables)       3
1403 0.121159     (waffles, vegetables, tortillas)       3
1404 0.130817      (yogurt, vegetables, tortillas)       3
1405 0.146620        (yogurt, waffles, vegetables)       3

[1331 rows x 3 columns]
```

Выводы:

Был изучен алгоритм поиска ассоциативных правил Apriori. Для предподготовки данных использовался TrancactionEncoder из библиотеки MLxtend.

Алгоритм Apriori позволяет выделить часто встречающиеся наборы данных, это может быть использовано для лучшего размещения товаров. Основным параметром, который изменялся в лабораторной работы был минимальный уровень поддержки. Он показывает насколько часто встречается тот или иной набор.