实验二 关联规则分析-Apriori算法

班级：信息安全193班 学号：8003119100 姓名：丁俊

1、导入数据

In [2]:

```
import os
import pandas as pd
ds_file = os.path.join(os.path.curdir, 'mushroom.dat')
df_data = pd.read_csv(ds_file, sep ='\\s+', header=None)
```

In [3]:

```
# 查看数据集
df_data
```

Out[3]:

	0	1	2	3	4	5	6	7	8	9	...	13	14	15	16	17	18	19	20	21	22
0	1	3	9	13	23	25	34	36	38	40	...	63	67	76	85	86	90	93	98	107	113
1	2	3	9	14	23	26	34	36	39	40	...	63	67	76	85	86	90	93	99	108	114
2	2	4	9	15	23	27	34	36	39	41	...	63	67	76	85	86	90	93	99	108	115
3	1	3	10	15	23	25	34	36	38	41	...	63	67	76	85	86	90	93	98	107	113
4	2	3	9	16	24	28	34	37	39	40	...	63	67	76	85	86	90	94	99	109	114
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8119	2	7	9	13	24	28	35	36	39	50	...	63	73	83	85	88	90	93	106	112	119
8120	2	3	9	13	24	28	35	36	39	50	...	63	73	83	85	87	90	93	106	110	119
8121	2	6	9	13	24	28	35	36	39	41	...	63	73	83	85	88	90	93	106	112	119
8122	1	7	10	13	24	31	34	36	38	48	...	66	67	76	85	86	90	94	102	110	119
8123	2	3	9	13	24	28	35	36	39	50	...	63	73	83	85	88	90	93	104	112	119

8124 rows × 23 columns

2、Apriori算法

2.1 初始化

In [4]:

```
import numpy as np
min_s = 0.3
min_c = 0.6
p_d = len(df_data)
num_min_s = np.round(min_s * p_d, 0)
```

In [5]:

```
num_min_s
```

Out[5]:

```
2437.0
```

2.2 创建频繁1项集

In [6]:

```
from collections import Counter
c1_sel = Counter(df_data.values.reshape(-1))
c1_fre = {k:v for k,v in c1_sel.items() if v>=num_min_s}
```

In [7]:

```
c1_sel
```

Out[7]:

```
Counter({1: 3916,
         3: 3656,
         9: 2556,
         13: 2284,
         23: 3376,
         25: 256,
         34: 7914,
         36: 6812,
         38: 2512,
         40: 408,
         52: 3516,
         54: 1120,
         59: 5176,
         63: 4936,
         67: 4464,
         76: 4384,
         85: 8124,
         86: 7924,
```

In [10]:

```
c1_fre
```

Out[10]:

```
{1: 3916,
 3: 3656,
 9: 2556,
 23: 3376,
 34: 7914,
 36: 6812,
 38: 2512,
 52: 3516,
 59: 5176,
 63: 4936,
 67: 4464,
 76: 4384,
 85: 8124,
 86: 7924,
 90: 7488,
 93: 3968,
 2: 4208,
 39: 5612,
 10: 3244,
 24: 4748,
 28: 3528,
 53: 4608,
 94: 2776,
 110: 4040,
 6: 3152,
 56: 3776,
 116: 3148,
 58: 2480}
```

2.3 创建关于有毒的频繁1项集

In [13]:

```
df_data2 = df_data[df_data[0]==2]
df_data2 = df_data2.iloc[:,1:]
ls_data2 = [set(i) for i in df_data2.values] # 对频繁项集进行去重
def gen_1(df_data2, num_min_s):
    ls_c1_sel = Counter(df_data2.values.reshape(-1))
    dic_c1_fre = {k:v for k,v in ls_c1_sel.items() if v >= num_min_s}
    ls_c1_fre = [set([k]) for k in dic_c1_fre.keys()]
    return ls_c1_fre, dic_c1_fre
```

In [14]:

```
ls_c1_fre, dic_c1_fre = gen_1(df_data2, num_min_s)
```

In [15]:

```
df_data2
```

Out[15]:

	1	2	3	4	5	6	7	8	9	10	...	13	14	15	16	17	18	19	20	21	22
1	3	9	14	23	26	34	36	39	40	52	...	63	67	76	85	86	90	93	99	108	114
2	4	9	15	23	27	34	36	39	41	52	...	63	67	76	85	86	90	93	99	108	115
4	3	9	16	24	28	34	37	39	40	53	...	63	67	76	85	86	90	94	99	109	114
5	3	10	14	23	26	34	36	39	41	52	...	63	67	76	85	86	90	93	98	108	114
6	4	9	15	23	26	34	36	39	42	52	...	63	67	76	85	86	90	93	98	108	115
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8115	3	9	13	24	28	35	36	39	50	52	...	63	73	83	85	88	90	93	104	110	119
8119	7	9	13	24	28	35	36	39	50	52	...	63	73	83	85	88	90	93	106	112	119
8120	3	9	13	24	28	35	36	39	50	52	...	63	73	83	85	87	90	93	106	110	119
8121	6	9	13	24	28	35	36	39	41	52	...	63	73	83	85	88	90	93	106	112	119

2.4 创建关于有毒的频繁k项集

In [16]:

```
def gen_sel(ls_ck_fre):
    ls_ck2_sel = []
    li = len(ls_ck_fre)
    pi = 0
    for si in ls_ck_fre[0:li]:
        pi = pi + 1
        for sj in ls_ck_fre[pi : li]:
            st = si | sj
            if len(st) > len(si) and st not in ls_ck2_sel:
                ls_ck2_sel.append(st)
    return ls_ck2_sel
```

In [17]:

```
ls_ck2_sel = gen_sel(ls_c1_fre)
ls_ck2_sel
```

Out[17]:

```
[{23, 34},
 {23, 36},
 {23, 39},
 {23, 59},
 {23, 63},
 {23, 67},
 {23, 76},
 {23, 85},
 {23, 86},
 {23, 90},
 {23, 93},
 {23, 28},
 {23, 53},
 {34, 36},
 {34, 39},
 {34, 59},
 {34, 63},
 {34, 67},
 {34, 76},
 {34, 85},
 {34, 86},
 {34, 90},
 {34, 93},
 {28, 34},
 {34, 53},
 {36, 39},
 {36, 59},
 {36, 63},
 {36, 67},
 {36, 76},
 {36, 85},
 {36, 86},
 {36, 90},
 {36, 93},
 {28, 36},
 {36, 53},
 {39, 59},
 {39, 63},
 {39, 67},
 {39, 76},
 {39, 85},
 {39, 86},
 {39, 90},
 {39, 93},
 {28, 39},
 {39, 53},
 {59, 63},
 {59, 67},
 {59, 76},
 {59, 85},
 {59, 86},
 {59, 90},
 {59, 93},
 {28, 59},
```

```
 {53, 59},
 {63, 67},
 {63, 76},
 {63, 85},
 {63, 86},
 {63, 90},
 {63, 93},
 {28, 63},
 {53, 63},
 {67, 76},
 {67, 85},
 {67, 86},
 {67, 90},
 {67, 93},
 {28, 67},
 {53, 67},
 {76, 85},
 {76, 86},
 {76, 90},
 {76, 93},
 {28, 76},
 {53, 76},
 {85, 86},
 {85, 90},
 {85, 93},
 {28, 85},
 {53, 85},
 {86, 90},
 {86, 93},
 {28, 86},
 {53, 86},
 {90, 93},
 {28, 90},
 {53, 90},
 {28, 93},
 {53, 93},
 {28, 53}]
```

In [18]:

```
def gen_fre(ls_ck2_sel, ls_data2, num_min_s):
    dic_ck2_fre = {}
    ls_ck2_fre = []
    for i in ls_ck2_sel:
        c = 0
        for j in ls_data2:
            if i & j == i:
                c = c  + 1
        if c >= num_min_s:
            dic_ck2_fre[tuple(i)] = c
            ls_ck2_fre.append(i)
    return ls_ck2_fre, dic_ck2_fre
```

In [19]:

```
ls_ck2_fre, dic_ck2_fre = gen_fre(ls_ck2_sel, ls_data2, num_min_s)
```

In [20]:

```
dic_ck2_fre # 频繁K项集出现的对应次数
```

Out[20]:

```
{(34, 23): 2752,
 (36, 23): 2656,
 (39, 23): 2656,
 (59, 23): 2752,
 (63, 23): 2560,
 (85, 23): 2752,
 (86, 23): 2752,
 (90, 23): 2528,
 (93, 23): 2560,
 (34, 36): 2816,
 (34, 39): 3728,
 (34, 59): 3448,
 (34, 63): 3208,
 (34, 67): 2752,
 (34, 76): 2704,
 (34, 85): 4016,
 (34, 86): 4016,
 (34, 90): 3488,
 (34, 93): 2960,
 (34, 28): 3216,
 (34, 53): 2592,
 (36, 39): 2864,
 (59, 36): 2992,
 (36, 63): 2752,
 (36, 85): 3008,
 (36, 86): 2816,
 (90, 36): 2768,
 (36, 93): 2768,
 (59, 39): 3376,
 (63, 39): 3184,
 (67, 39): 2464,
 (76, 39): 2464,
 (85, 39): 3920,
 (86, 39): 3728,
 (90, 39): 3392,
 (93, 39): 2960,
 (28, 39): 3216,
 (53, 39): 2496,
 (59, 63): 3124,
 (59, 85): 3640,
 (59, 86): 3448,
 (90, 59): 3272,
 (59, 93): 2992,
 (59, 28): 2840,
 (85, 63): 3400,
 (86, 63): 3208,
 (90, 63): 3032,
 (93, 63): 2800,
 (28, 63): 2792,
 (67, 85): 2752,
 (67, 86): 2752,
 (76, 85): 2704,
 (76, 86): 2704,
 (85, 86): 4016,
```

```
 (90, 85): 3680,
 (93, 85): 3152,
 (28, 85): 3408,
 (53, 85): 2592,
 (90, 86): 3488,
 (93, 86): 2960,
 (28, 86): 3216,
 (53, 86): 2592,
 (90, 93): 2816,
 (90, 28): 2880,
 (90, 53): 2592,
 (28, 53): 2496}
```

2.5 main程序

In [22]:

```
def aprior_fre(df_data, min_s):
    df_data2 = df_data[df_data[0] == 2]
    df_data2 = df_data2.iloc[:,1:]
    ls_data2 = [set(i) for i in df_data2.values]
    ls_c_fre = []
    dic_c_fre = {}
    ls_c1_fre, dic_c1_fre = gen_1(df_data2, min_s)
    ls_c_fre.extend(ls_c1_fre)
    dic_c1_fre.update(dic_c1_fre)
    num_k = len(ls_c1_fre)
    ls_ck2_fre = ls_c1_fre
    for k in range(2, num_k):
        print(k)
        ls_ck2_sel = gen_sel(ls_ck2_fre)
        ls_ck2_fre, dic_ck2_fre = gen_fre(ls_ck2_sel, ls_data2, min_s)
        ls_ck_fre = ls_ck2_fre
        ls_c_fre.extend(ls_ck2_fre)
        dic_c1_fre.update(dic_ck2_fre)
        if len(dic_ck2_fre) == 0:
            break
    return ls_c1_fre, dic_c1_fre
```

In [25]:

```
ls_c1_fre, dic_c_fre = aprior_fre(df_data, num_min_s)
```

```
2
3
4
5
6
7
8
9
```

In [26]:

```
ls_c1_fre
```

Out[26]:

```
[{23},
 {34},
 {36},
 {39},
 {59},
 {63},
 {67},
 {76},
 {85},
 {86},
 {90},
 {93},
 {28},
 {53}]
```

In [ ]:

```

```