Apriori

library python:

-Numpy

-Pandas

- mlxtend

Dataset

id	Wine	Chips	Bread	Butter	Milk	Apple
1	Yes	Yes	Yes	Yes	Yes	Yes
2	Yes	No	Yes	Yes	Yes	No
3	No	No	Yes	Yes	Yes	No
4	No	Yes	No	No	No	Yes
5	Yes	Yes	Yes	Yes	Yes	Yes
6	Yes	Yes	No	No	Yes	No
7	Yes	Yes	Yes	Yes	No	Yes
8	Yes	Yes	No	No	Yes	No
9	No	No	Yes	No	No	Yes
10	Yes	No	Yes	Yes	Yes	No
11	No	Yes	Yes	Yes	No	Yes
12	Yes	No	No	Yes	Yes	Yes
13	Yes	Yes	Yes	Yes	Yes	No
14	Yes	No	Yes	No	Yes	Yes
15	Yes	No	Yes	Yes	Yes	Yes
16	Yes	Yes	No	Yes	Yes	Yes
17	No	Yes	Yes	Yes	Yes	Yes
18	No	Yes	No	No	Yes	Yes
19	Yes	Yes	Yes	Yes	Yes	Yes
20	Yes	No	Yes	Yes	Yes	Yes
21	Yes	Yes	Yes	No	Yes	Yes
22	No	Yes	No	No	No	No

CSV:

```
id,Wine,Chips,Bread,Butter,Milk,Apple
1,Yes,Yes,Yes,Yes,Yes,Yes
2,Yes,No,Yes,Yes,Yes,No
3,No,No,Yes,Yes,Yes,No
4,No,Yes,No,No,No,Yes
5,Yes,Yes,Yes,Yes,Yes,Yes
6,Yes,Yes,No,No,Yes,No
7,Yes,Yes,Yes,Yes,No,Yes
8,Yes,Yes,No,No,Yes,No
9,No,No,Yes,No,No,Yes
10,Yes,No,Yes,Yes,Yes,No
11,No,Yes,Yes,Yes,No,Yes
12,Yes,No,No,Yes,IYes,Yes
13,Yes,Yes,Yes,Yes,Yes,No
14,Yes,No,Yes,No,Yes,Yes
15,Yes,No,Yes,Yes,Yes,Yes
16,Yes,Yes,No,Yes,Yes,Yes
17,No,Yes,Yes,Yes,Yes,Yes
18,No,Yes,No,No,Yes,Yes
19,Yes,Yes,Yes,Yes,Yes,Yes
20,Yes,No,Yes,Yes,Yes,Yes
21,Yes,Yes,Yes,No,Yes,Yes
22,No,Yes,No,No,No,No
```

Weka

```
Best rules found:

 1. Wine=Yes Chips=No 6 ==> Milk=Yes 6    <conf:(1)> lift:(1.29) lev:(0.06) [1] conv:(1.36)
 2. Chips=No Butter=Yes 6 ==> Milk=Yes 6    <conf:(1)> lift:(1.29) lev:(0.06) [1] conv:(1.36)
 3. Wine=Yes Apple=No 5 ==> Milk=Yes 5    <conf:(1)> lift:(1.29) lev:(0.05) [1] conv:(1.14)
 4. Bread=No Butter=No 5 ==> Chips=Yes 5    <conf:(1)> lift:(1.57) lev:(0.08) [1] conv:(1.82)
 5. Wine=Yes Chips=No Bread=Yes 5 ==> Milk=Yes 5    <conf:(1)> lift:(1.29) lev:(0.05) [1] conv:(1.14)
 6. Wine=Yes Chips=No Butter=Yes 5 ==> Milk=Yes 5    <conf:(1)> lift:(1.29) lev:(0.05) [1] conv:(1.14)
 7. Chips=No Bread=Yes Butter=Yes 5 ==> Milk=Yes 5    <conf:(1)> lift:(1.29) lev:(0.05) [1] conv:(1.14)
 8. Wine=Yes 15 ==> Milk=Yes 14    <conf:(0.93)> lift:(1.21) lev:(0.11) [2] conv:(1.7)
 9. Wine=Yes Bread=Yes 11 ==> Milk=Yes 10    <conf:(0.91)> lift:(1.18) lev:(0.07) [1] conv:(1.25)
10. Wine=Yes Butter=Yes 11 ==> Milk=Yes 10    <conf:(0.91)> lift:(1.18) lev:(0.07) [1] conv:(1.25)
```

Python

```
import pandas as pd
df=pd.read_csv("Jogan.csv",index_col=0)
df_new=df
df_new=df_new.replace(to_replace = ['Yes','No'],value = ['1','0'])
```

id	Wine	Chips	Bread	Butter	Milk	Apple
1	1	1	1	1	1	1
2	1	0	1	1	1	0
3	0	0	1	1	1	0
4	0	1	0	0	0	1
5	1	1	1	1	1	1
6	1	1	0	0	1	0
7	1	1	1	1	0	1
8	1	1	0	0	1	0
9	0	0	1	0	0	1
10	1	0	1	1	1	0
11	0	1	1	1	0	1
12	1	0	0	1	1	1
13	1	1	1	1	1	0
14	1	0	1	0	1	1
15	1	0	1	1	1	1
16	1	1	0	1	1	1
17	0	1	1	1	1	1
18	0	1	0	0	1	1
19	1	1	1	1	1	1
20	1	0	1	1	1	1
21	1	1	1	0	1	1
22	0	1	0	0	0	0

```
Best rules found:

 1. Wine=Yes Chips=No 6 ==> Milk=Yes 6    <conf:(1)> lift:(1.29) lev:(0.06) [1] conv:(1.36)
 2. Chips=No Butter=Yes 6 ==> Milk=Yes 6    <conf:(1)> lift:(1.29) lev:(0.06) [1] conv:(1.36)
 3. Wine=Yes Apple=No 5 ==> Milk=Yes 5    <conf:(1)> lift:(1.29) lev:(0.05) [1] conv:(1.14)
 4. Bread=No Butter=No 5 ==> Chips=Yes 5    <conf:(1)> lift:(1.57) lev:(0.08) [1] conv:(1.82)
 5. Wine=Yes Chips=No Bread=Yes 5 ==> Milk=Yes 5    <conf:(1)> lift:(1.29) lev:(0.05) [1] conv:(1.14)
 6. Wine=Yes Chips=No Butter=Yes 5 ==> Milk=Yes 5    <conf:(1)> lift:(1.29) lev:(0.05) [1] conv:(1.14)
 7. Chips=No Bread=Yes Butter=Yes 5 ==> Milk=Yes 5    <conf:(1)> lift:(1.29) lev:(0.05) [1] conv:(1.14)
 8. Wine=Yes 15 ==> Milk=Yes 14    <conf:(0.93)> lift:(1.21) lev:(0.11) [2] conv:(1.7)
 9. Wine=Yes Bread=Yes 11 ==> Milk=Yes 10    <conf:(0.91)> lift:(1.18) lev:(0.07) [1] conv:(1.25)
10. Wine=Yes Butter=Yes 11 ==> Milk=Yes 10    <conf:(0.91)> lift:(1.18) lev:(0.07) [1] conv:(1.25)
```

Rekomendasi jika membeli wine dan chips, rekomdensasi yang paling bagus adalah Milk dengan confidential(100%).

Python

```
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
df=pd.read_csv("Jogan.csv",index_col=0)
df_new=df
df_new=df_new.replace(to_replace = ['Yes','No'],value = ['1','0']).astype(int)

frequent_itemsets = apriori(df_new, min_support=0.07, use_colnames=True)
frequent_itemsets.head()
```

Index	support	itemsets
0	0.772727	frozenset({'Milk'})
1	0.681818	frozenset({'Wine'})
2	0.681818	frozenset({'Bread'})
3	0.681818	frozenset({'Apple'})
4	0.636364	frozenset({'Chips'})
5	0.636364	frozenset({'Butter'})
6	0.636364	frozenset({'Wine', 'Milk'})
7	0.545455	frozenset({'Milk', 'Bread'})
8	0.545455	frozenset({'Butter', 'Bread'})
9	0.545455	frozenset({'Butter', 'Milk'})
10	0.5	frozenset({'Wine', 'Bread'})
11	0.5	frozenset({'Butter', 'Wine'})
12	0.5	frozenset({'Apple', 'Bread'})
13	0.5	frozenset({'Apple', 'Milk'})
14	0.454545	frozenset({'Apple', 'Wine'})
15	0.454545	frozenset({'Chips', 'Milk'})
16	0.454545	frozenset({'Apple', 'Chips'})
17	0.454545	frozenset({'Butter', 'Apple'})
18	0.454545	frozenset({'Milk', 'Wine', 'Bread'})
19	0.454545	frozenset({'Butter', 'Wine', 'Milk'})
20	0.454545	frozenset({'Butter', 'Milk', 'Bread'})
21	0.409091	frozenset({'Wine', 'Chips'})
22	0.409091	frozenset({'Butter', 'Wine', 'Bread'})
23	0.409091	frozenset({'Apple', 'Wine', 'Milk'})
24	0.363636	frozenset({'Wine', 'Chips', 'Milk'})
25	0.363636	frozenset({'Apple', 'Wine', 'Bread'})

Index	support	itemsets
26	0.363636	frozenset({'Butter', 'Apple', 'Wine'})
27	0.363636	frozenset({'Butter', 'Apple', 'Bread'})
28	0.363636	frozenset({'Apple', 'Milk', 'Bread'})
29	0.363636	frozenset({'Butter', 'Apple', 'Milk'})
30	0.363636	frozenset({'Chips', 'Bread'})
31	0.363636	frozenset({'Butter', 'Chips'})
32	0.363636	frozenset({'Butter', 'Milk', 'Wine', 'Bread'})
33	0.318182	frozenset({'Butter', 'Chips', 'Bread'})
34	0.318182	frozenset({'Apple', 'Chips', 'Bread'})
35	0.318182	frozenset({'Butter', 'Apple', 'Chips'})
36	0.318182	frozenset({'Apple', 'Chips', 'Milk'})
37	0.318182	frozenset({'Apple', 'Milk', 'Wine', 'Bread'})
38	0.318182	frozenset({'Butter', 'Apple', 'Wine', 'Milk'})
39	0.272727	frozenset({'Wine', 'Chips', 'Bread'})
40	0.272727	frozenset({'Butter', 'Wine', 'Chips'})
41	0.272727	frozenset({'Apple', 'Wine', 'Chips'})
42	0.272727	frozenset({'Milk', 'Chips', 'Bread'})
43	0.272727	frozenset({'Butter', 'Chips', 'Milk'})
44	0.272727	frozenset({'Butter', 'Apple', 'Wine', 'Bread'})
45	0.272727	frozenset({'Butter', 'Apple', 'Chips', 'Bread'})
46	0.272727	frozenset({'Butter', 'Apple', 'Milk', 'Bread'})
47	0.227273	frozenset({'Butter', 'Wine', 'Chips', 'Bread'})
48	0.227273	frozenset({'Milk', 'Wine', 'Chips', 'Bread'})
49	0.227273	frozenset({'Apple', 'Wine', 'Chips', 'Bread'})
50	0.227273	frozenset({'Butter', 'Wine', 'Chips', 'Milk'})

Index	support	itemsets
51	0.227273	frozenset({'Butter', 'Apple', 'Wine', 'Chips'})
52	0.227273	frozenset({'Apple', 'Wine', 'Chips', 'Milk'})
53	0.227273	frozenset({'Butter', 'Milk', 'Chips', 'Bread'})
54	0.227273	frozenset({'Apple', 'Milk', 'Chips', 'Bread'})
55	0.227273	frozenset({'Butter', 'Apple', 'Chips', 'Milk'})
56	0.227273	frozenset({'Bread', 'Apple', 'Wine', 'Butter', 'Milk'})
57	0.181818	frozenset({'Bread', 'Wine', 'Butter', 'Chips', 'Milk'})
58	0.181818	frozenset({'Bread', 'Apple', 'Wine', 'Butter', 'Chips'})
59	0.181818	frozenset({'Bread', 'Apple', 'Wine', 'Chips', 'Milk'})
60	0.181818	frozenset({'Apple', 'Wine', 'Butter', 'Chips', 'Milk'})
61	0.181818	frozenset({'Bread', 'Apple', 'Butter', 'Chips', 'Milk'})
62	0.136364	frozenset({'Bread', 'Apple', 'Wine', 'Butter', 'Chips', 'Milk'})

Rekomendasi dari kiri ke kanan.