

Lista 4 - Aprendizado de Máquina

Curso: Ciência de Dados

Disciplina: Aprendizado de Máquina I

Profa. Cristiane Neri Nobre

Data de entrega: 03/11

Valor: 1,5 ponto

Nome Discente: Gustavo Costa

Questão 1

Questão 01

Considere que em um determinado supermercado foram efetuadas as seguintes transações:

Nº	Leite	Café	Cerveja	Pão	Manteiga	Arroz	Feijão
1	Não	Sim	Não	Sim	Sim	Não	Não
2	Sim	Não	Sim	Sim	Sim	Não	Não
3	Não	Sim	Não	Sim	Sim	Não	Não
4	Sim	Sim	Não	Sim	Sim	Não	Não
5	Não	Não	Sim	Não	Não	Não	Não
6	Não	Não	Não	Não	Sim	Não	Não
7	Não	Não	Não	Sim	Não	Não	Não
8	Não	Não	Não	Não	Não	Não	Sim
9	Não	Não	Não	Não	Não	Sim	Sim
10	Não	Não	Não	Não	Não	Sim	Não

Utilizando-se o algoritmo Apriori, um suporte mínimo aceitável de **0.3** e confiança de **0.8**, o número de **ItensSets 1, 2, 3** e de **regras** a partir desta base de dados são:

Passo 1: Gerar Itemset 1

- Leite: 2/10 = 0.2 ✗
- Café: 3/10 = 0.3 ✓
- Cerveja: 2/10 = 0.2 ✗
- Pão: 5/10 = 0.5 ✓
- Manteiga: 5/10 = 0.5 ✓
- Arroz: 2/10 = 0.2 ✗
- Feijão: 2/10 = 0.2 ✗

Passo 2: Gerar Itemset 2

- Café e Pão: 3/10 = 0.3 ✓
- Café e Manteiga: 3/10 = 0.3 ✓
- Pão e Manteiga: 4/10 = 0.4 ✓

Passo 3: Gerar Itemset 3

- Café, Pão e Manteiga: 3/10 = 0.3 ✓

Passo 4: Construção das Implicações Lógicas

1. **Café -> Pão:** 3/3 = 1 ✓
2. **Pão -> Café:** 3/5 = 0.6 ✗
3. **Café -> Manteiga:** 3/3 = 1 ✓
4. **Manteiga -> Café:** 3/5 = 0.75 ✗
5. **Pão -> Manteiga:** 4/5 = 0.8 ✓
6. **Manteiga -> Pão:** 4/5 = 0.8 ✓
7. **Café -> Pão e Manteiga:** 3/3 = 1 ✓
8. **Pão e Manteiga -> Café:** 3/4 = 0.75 ✗
9. **Pão -> Café e Manteiga:** 3/5 = 0.6 ✗
10. **Café e Manteiga -> Pão:** 3/3 = 1 ✓
11. **Manteiga -> Café e Pão:** 3/5 = 0.6 ✗
12. **Café e Pão -> Manteiga:** 3/3 = 1 ✓

Passo 5: Regras de Associação Escolhidas com confiança > 0.8

1. Café -> Pão
2. Café -> Manteiga
3. Pão -> Manteiga
4. Manteiga -> Pão
5. Café -> Pão e Manteiga
6. Café e Manteiga -> Pão
7. Café e Pão -> Manteiga

Questão 2

Os resultados coincidem com as regras encontradas no algoritmo feito no 'papel'.

RegrasFinais.sort_values(by='lift', ascending =False)

	Lhs	Rhs	suporte	confianca	lift
25	[nan, Cafe]	[Manteiga, Pao]	0.3	1.0	2.5
22	[Cafe] [nan, Manteiga, Pao]		0.3	1.0	2.5
8	[Cafe]	[Manteiga, Pao]	0.3	1.0	2.5
14	[Cafe]	[nan, Pao]	0.3	1.0	2.0
11	[Cafe]	[nan, Manteiga]	0.3	1.0	2.0
27	[nan, Manteiga, Cafe]	[Pao]	0.3	1.0	2.0
24	[Pao, Cafe]	[nan, Manteiga]	0.3	1.0	2.0
23	[Manteiga, Cafe]	[nan, Pao]	0.3	1.0	2.0
16	[nan, Cafe]	[Pao]	0.3	1.0	2.0
1	[Cafe]	[Manteiga]	0.3	1.0	2.0
13	[nan, Cafe]	[Manteiga]	0.3	1.0	2.0
28	[nan, Pao, Cafe]	[Manteiga]	0.3	1.0	2.0
9	[Manteiga, Cafe]	[Pao]	0.3	1.0	2.0
2	[Cafe]	[Pao]	0.3	1.0	2.0
10	[Pao, Cafe]	[Manteiga]	0.3	1.0	2.0
20	[nan, Manteiga]	[Pao]	0.4	0.8	1.6
5	[Pao]	[Manteiga]	0.4	0.8	1.6
17	[Manteiga]	[nan, Pao]	0.4	0.8	1.6
18	[Pao]	[nan, Manteiga]	0.4	0.8	1.6
21	[nan, Pao]	[Manteiga]	0.4	0.8	1.6
4	[Manteiga]	[Pao]	0.4	0.8	1.6
26	[Manteiga, Pao, Cafe]	[nan]	0.3	1.0	1.0
3	[Cafe]	[nan]	0.3	1.0	1.0
12	[Manteiga, Cafe]	[nan]	0.3	1.0	1.0
6	[Manteiga]	[nan]	0.5	1.0	1.0
19	[Manteiga, Pao]	[nan]	0.4	1.0	1.0
7	[Pao]	[nan]	0.5	1.0	1.0
15	[Pao, Cafe]	[nan]	0.3	1.0	1.0
0	[]	[nan]	1.0	1.0	1.0

Questão 3

Feito.



```
# Impressão dos itemsets e suportes
print("Itemsets gerados e seus respectivos suportes:")
for resultado in saida:
    itemset = list(resultado.items)
    suporte = resultado.support
    print(f"Itemset: {itemset}, Suporte: {suporte}")
```



```
Itemsets gerados e seus respectivos suportes:
Itemset: ['nan'], Suporte: 1.0
Itemset: ['Manteiga', 'Cafe'], Suporte: 0.3
Itemset: ['Pao', 'Cafe'], Suporte: 0.3
Itemset: ['nan', 'Cafe'], Suporte: 0.3
Itemset: ['Manteiga', 'Pao'], Suporte: 0.4
Itemset: ['nan', 'Manteiga'], Suporte: 0.5
Itemset: ['nan', 'Pao'], Suporte: 0.5
Itemset: ['Manteiga', 'Pao', 'Cafe'], Suporte: 0.3
Itemset: ['nan', 'Manteiga', 'Cafe'], Suporte: 0.3
Itemset: ['nan', 'Pao', 'Cafe'], Suporte: 0.3
Itemset: ['nan', 'Manteiga', 'Pao'], Suporte: 0.4
Itemset: ['nan', 'Manteiga', 'Pao', 'Cafe'], Suporte: 0.3
```

Questão 4

Com essa alteração feita, de 12 regras ele gerará 449 regras. Isso porque o algoritmo considerará o "não levar" determinado item como um evento possível e pra isso criará todas as combinações existentes.



```
RegrasFinais.sort_values(by='lift', ascending=False)
```



	Lhs	Rhs	suporte	confianca	lift
448	[não Arroz, não Cerveja, não Feijão, Manteiga,...]	[Cafe]	0.3	1.0	3.333333
430	[não Cerveja, Manteiga, Pao]	[não Feijão, não Arroz, Cafe]	0.3	1.0	3.333333
363	[não Feijão, não Cerveja, Manteiga, Pao]	[Cafe]	0.3	1.0	3.333333
358	[não Cerveja, Manteiga, Pao]	[não Feijão, Cafe]	0.3	1.0	3.333333
414	[Cafe]	[não Arroz, não Cerveja, não Feijão, Manteiga,...]	0.3	1.0	3.333333
...
276	[não Feijão, não Cerveja, não Leite]	[não Arroz]	0.4	0.8	1.000000
275	[não Feijão, não Arroz, não Leite]	[não Cerveja]	0.4	0.8	1.000000
274	[não Arroz, não Cerveja, não Leite]	[não Feijão]	0.4	0.8	1.000000
273	[não Feijão, não Arroz, não Cerveja]	[não Leite]	0.4	0.8	1.000000
0	[]	[não Arroz]	0.8	0.8	1.000000

449 rows × 5 columns

Questão 5

É também uma biblioteca de Aprendizado de Máquina, ela também possui as Regras de Associação inclusas juntamente com o algoritmo apriori, mas os dados precisam estar em um formato específico que é uma matriz booleana, é o equivalente à 'Sim' e 'Nao' por exemplo.

Questão 6

O Artigo citado não está disponível no Canvas.