

Lista de IA #6

Questão 1) Suporte = 0.3 / Confiança = 0.8

ItemSet 1

Leite	Café	Cerveja	Pão	Manteiga	Arroz	Feijão
0.2	0.3	0.2	0.5	0.5	0.2	0.2

ItemSet 2

Produtos	Café e Pão	Café e Manteiga	Manteiga e Pão
Suporte	0.3	0.3	0.4
Confiança	Café->Pão = 1 Pão->Café = 0.6	Café->Manteiga = 1 Manteiga->Café = 0.6	Pão->Manteiga = 0.8 Manteiga->Pão = 0.8

ItemSet 3

Produtos	Confiança
Café e Pão -> Manteiga	1
Café e Manteiga -> Pão	1
Manteiga e Pão -> Café	0.75
Café -> Pão e Manteiga	1
Pão -> Café e Manteiga	0.6
Manteiga -> Café e Pão	0.6

Os que foram marcados com a cor vermelha serão os que não serão regras!

Questão 2 e 3)

```
[1] !pip install apyori

[2] import pandas as pd
    from apyori import apriori

[3] #encoding = "cp1252", encoding='utf-8', encoding='latin1'etc
    base= pd.read_csv('/content/sample_data/MercadoSim.csv', sep=';', encoding='cp1252', header=None)
    base
```

```
[5] transacoes = []
    for i in range(len(base)):
        #print(i)
        #print(base_mercado1.values[i, 0])
        transacoes.append([str(base.values[i, j]) for j in range(base.shape[1])])
```

transacoes

```
[['nan', 'Cafe', 'nan', 'Pao', 'Manteiga', 'nan', 'nan'],
 ['Leite', 'nan', 'Cerveja', 'Pao', 'Manteiga', 'nan', 'nan'],
 ['nan', 'Cafe', 'nan', 'Pao', 'Manteiga', 'nan', 'nan'],
 ['Leite', 'Cafe', 'nan', 'Pao', 'Manteiga', 'nan', 'nan'],
 ['nan', 'nan', 'Cerveja', 'nan', 'nan', 'nan', 'nan'],
 ['nan', 'nan', 'nan', 'nan', 'Manteiga', 'nan', 'nan'],
 ['nan', 'nan', 'nan', 'Pao', 'nan', 'nan', 'nan'],
 ['nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'Feijao'],
 ['nan', 'nan', 'nan', 'nan', 'nan', 'Arroz', 'Feijao'],
 ['nan', 'nan', 'nan', 'nan', 'nan', 'Arroz', 'nan']]
```

Vamos chamar executar o algoritmo apriori e armazenar as regras obtidas

```
[16] regras = apriori(transacoes, min_support = 0.3, min_confidence = 0.8)
      saida = list(regras)
      print(len(saida))
      print(saida)
```

```
Antecedente = []
Consequente = []
suporte = []
confianca = []
lift = []
RegrasFinais = []
```

```
for resultado in saida:
    s = resultado[1]
    result_rules = resultado[2]
    for result_rule in result_rules:
        a = list(result_rule[0])
        b = list(result_rule[1])
        c = result_rule[2]
        l = result_rule[3]
        if 'nan' in a or 'nan' in b: continue
        if len(a) == 0 or len(b) == 0: continue
        Antecedente.append(a)
        Consequente.append(b)
        suporte.append(s)
        confianca.append(c)
        lift.append(l)
RegrasFinais = pd.DataFrame({'Antecedente': Antecedente, 'Consequente': Consequente, 'suporte': suporte, 'confianca': confianca, 'lift': li
```

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



	Antecedente	Consequente	suporte	confianca	lift
4	[Cafe]	[Pao, Manteiga]	0.3	1.0	2.5
1	[Cafe]	[Pao]	0.3	1.0	2.0
0	[Cafe]	[Manteiga]	0.3	1.0	2.0
6	[Pao, Cafe]	[Manteiga]	0.3	1.0	2.0
5	[Manteiga, Cafe]	[Pao]	0.3	1.0	2.0
2	[Manteiga]	[Pao]	0.4	0.8	1.6
3	[Pao]	[Manteiga]	0.4	0.8	1.6



```
print("Todos os Itemsets Frequentes:")  
for itemset in todos_itemsets:  
    print(itemset)
```



```
Todos os Itemsets Frequentes:  
['nan']  
['Manteiga', 'Cafe']  
['Pao', 'Cafe']  
['nan', 'Cafe']  
['Pao', 'Manteiga']  
['nan', 'Manteiga']  
['Pao', 'nan']  
['Pao', 'Manteiga', 'Cafe']  
['nan', 'Manteiga', 'Cafe']  
['Pao', 'nan', 'Cafe']  
['Pao', 'nan', 'Manteiga']  
['Pao', 'nan', 'Manteiga', 'Cafe']
```

Questão 4)

```
transacoes_nova = transacoes.copy()
for l in range(len(transacoes_nova)):
    if 'nan' in transacoes_nova[l][0]:
        transacoes_nova[l][0] = 'Nao_Leite'
    for l in range(len(transacoes_nova)):
        if 'nan' in transacoes_nova[l][1]:
            transacoes_nova[l][1] = 'Nao_Cafe'
    for l in range(len(transacoes_nova)):
        if 'nan' in transacoes_nova[l][2]:
            transacoes_nova[l][2] = 'Nao_Cerveja'
    for l in range(len(transacoes_nova)):
        if 'nan' in transacoes_nova[l][3]:
            transacoes_nova[l][3] = 'Nao_Pao'
    for l in range(len(transacoes_nova)):
        if 'nan' in transacoes_nova[l][4]:
            transacoes_nova[l][4] = 'Nao_Manteiga'
    for l in range(len(transacoes_nova)):
        if 'nan' in transacoes_nova[l][5]:
            transacoes_nova[l][5] = 'Nao_Arroz'
    for l in range(len(transacoes_nova)):
        if 'nan' in transacoes_nova[l][6]:
            transacoes_nova[l][6] = 'Nao_Feijao'
transacoes_nova

[62] regras_nova = apriori(transacoes_nova, min_support = 0.5, min_confidence = 0.9)
saida_nova = list(regras_nova)
```

	Antecedente_nova	Consequente_nova	suporte_nova	confianca_nova	lift_nova
18	[Nao_Pao]	[Nao_Cafe, Nao_Leite]	0.5	1.0	1.666667
15	[Nao_Manteiga]	[Nao_Cafe, Nao_Leite]	0.5	1.0	1.666667
8	[Manteiga]	[Nao_Arroz, Nao_Feijao]	0.5	1.0	1.428571
3	[Nao_Manteiga]	[Nao_Cafe]	0.5	1.0	1.428571
4	[Nao_Pao]	[Nao_Cafe]	0.5	1.0	1.428571
20	[Nao_Pao, Nao_Leite]	[Nao_Cafe]	0.5	1.0	1.428571
17	[Nao_Leite, Nao_Manteiga]	[Nao_Cafe]	0.5	1.0	1.428571
11	[Pao]	[Nao_Arroz, Nao_Feijao]	0.5	1.0	1.428571
0	[Manteiga]	[Nao_Arroz]	0.5	1.0	1.250000
1	[Manteiga]	[Nao_Feijao]	0.5	1.0	1.250000
2	[Pao]	[Nao_Arroz]	0.5	1.0	1.250000
6	[Nao_Manteiga]	[Nao_Leite]	0.5	1.0	1.250000
5	[Pao]	[Nao_Feijao]	0.5	1.0	1.250000
12	[Pao, Nao_Arroz]	[Nao_Feijao]	0.5	1.0	1.250000
10	[Manteiga, Nao_Feijao]	[Nao_Arroz]	0.5	1.0	1.250000
9	[Nao_Arroz, Manteiga]	[Nao_Feijao]	0.5	1.0	1.250000
7	[Nao_Pao]	[Nao_Leite]	0.5	1.0	1.250000
16	[Nao_Cafe, Nao_Manteiga]	[Nao_Leite]	0.5	1.0	1.250000
14	[Nao_Cerveja, Nao_Cafe]	[Nao_Leite]	0.5	1.0	1.250000
13	[Pao, Nao_Feijao]	[Nao_Arroz]	0.5	1.0	1.250000
19	[Nao_Pao, Nao_Cafe]	[Nao_Leite]	0.5	1.0	1.250000

Questão 5)

```
transacoes = []
for i in range(len(base)):
    transacoes.append([str(base.values[i, j]) for j in range(base.shape[1])])
```

```
te = TransactionEncoder()
te_ary = te.fit(transacoes).transform(transacoes)
df = pd.DataFrame(te_ary, columns=te.columns_)
df.drop(columns=['nan'], inplace=True)
df
```

	Arroz	Cafe	Cerveja	Feijao	Leite	Manteiga	Pao
0	False	True	False	False	False	True	True
1	False	False	True	False	True	True	True
2	False	True	False	False	False	True	True
3	False	True	False	False	True	True	True
4	False	False	True	False	False	False	False
5	False	False	False	False	False	True	False
6	False	False	False	False	False	False	True
7	False	False	False	True	False	False	False
8	True	False	False	True	False	False	False
9	True	False	False	False	False	False	False

```
from mlxtend.frequent_patterns import apriori
apriori(df, min_support=0.3, use_colnames=True)
```

	support	itemsets
0	0.3	(Cafe)
1	0.5	(Manteiga)
2	0.5	(Pao)
3	0.3	(Manteiga, Cafe)
4	0.3	(Pao, Cafe)
5	0.4	(Manteiga, Pao)
6	0.3	(Manteiga, Pao, Cafe)

```
frequent_itemsets = apriori(df, min_support=0.3, use_colnames=True)
frequent_itemsets['length'] = frequent_itemsets['itemsets'].apply(lambda x: len(x))
frequent_itemsets
```

	support	itemsets	length
0	0.3	(Cafe)	1
1	0.5	(Manteiga)	1
2	0.5	(Pao)	1
3	0.3	(Manteiga, Cafe)	2
4	0.3	(Pao, Cafe)	2
5	0.4	(Manteiga, Pao)	2
6	0.3	(Manteiga, Pao, Cafe)	3

Questão 6)

O artigo, "A comprehensive review of visualization methods for association rule" oferece uma avaliação metódica e minuciosa dos métodos de visualização utilizados na mineração de regras de associação (ARM). O artigo destaca que, devido ao grande volume de regras geradas por algoritmos como Apriori e seus sucessores, a visualização é essencial para simplificar a compreensão e aplicação prática dos resultados, além de auxiliar na explicabilidade em inteligência artificial (XAI). A pesquisa analisa tanto técnicas convencionais, tais como gráficos de scatter, matriz-based plots, mosaic e double decker, que ilustram as relações entre medidas de suporte, confiança e lift, quanto métodos inovadores de "nova geração", como diagramas de Ishikawa (fishbone), representações moleculares, mapas de metro, diagramas de Sankey, gráficos de ribbon e gráficos baseados em glicídios, além de gráficos baseados em glicídios. Além disso, a revisão estabelece uma taxonomia que categoriza os métodos com base em critérios como a quantidade de medidas de interesse apresentadas, a extensão do conjunto de regras, a interatividade e o foco da visualização (itens, estrutura ou dados temporais). Além disso, propõe estudos futuros para aprimorar técnicas híbridas e melhorar a compreensão e usabilidade dos resultados de ARM.

