

Lista 6

Questão 01 – ItemSets e Regras (suporte $\geq 0,3$, confiança $\geq 0,8$)

ItemSets frequentes:

- Tamanho 1: {Leite}, {Café}, {Manteiga}, {Pão} (4 conjuntos)
- Tamanho 2: {Café, Manteiga}, {Café, Pão}, {Manteiga, Pão} (3 conjuntos)
- Tamanho 3: {Café, Manteiga, Pão} (1 conjunto)

Total de regras geradas com confiança $\geq 0,8$: 7

Regras:

- {Café} \rightarrow {Manteiga} (conf = 1,0)
- {Café} \rightarrow {Pão} (1,0)
- {Manteiga} \rightarrow {Pão} (1,0)
- {Pão} \rightarrow {Manteiga} (0,83)
- {Café} \rightarrow {Manteiga, Pão} (1,0)
- {Café, Manteiga} \rightarrow {Pão} (1,0)
- {Café, Pão} \rightarrow {Manteiga} (1,0)

Questão 02 –

Podemos ordenar estas regras por uma métrica desejada

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

✓ 0.0s

	Antecedente	Consequente	suporte	confianca	lift
30	[Cafe, Leite]	[Pao]	0.3	1.000000	1.428571
2	[Feijao]	[Arroz]	0.3	0.600000	1.200000
1	[Arroz]	[Feijao]	0.3	0.600000	1.200000
19	[Arroz]	[Cafe, Manteiga]	0.3	0.600000	1.200000
15	[Feijao]	[Cafe, Arroz]	0.3	0.600000	1.200000
18	[Cafe, Feijao]	[Arroz]	0.3	0.600000	1.200000
16	[Cafe, Arroz]	[Feijao]	0.3	0.600000	1.200000
14	[Arroz]	[Cafe, Feijao]	0.3	0.600000	1.200000
22	[Cafe, Manteiga]	[Arroz]	0.3	0.600000	1.200000
26	[Feijao]	[Cafe, Pao]	0.4	0.800000	1.142857
11	[Feijao]	[Pao]	0.4	0.800000	1.142857
33	[Cafe, Manteiga]	[Pao]	0.4	0.800000	1.142857
27	[Cafe, Feijao]	[Pao]	0.4	0.800000	1.142857
0	[Arroz]	[Cafe]	0.5	1.000000	1.111111
25	[Arroz, Pao]	[Cafe]	0.3	1.000000	1.111111
21	[Arroz, Manteiga]	[Cafe]	0.3	1.000000	1.111111
6	[Feijao]	[Cafe]	0.5	1.000000	1.111111
17	[Arroz, Feijao]	[Cafe]	0.3	1.000000	1.111111

Questão 03 –

```
dados_itemsets = []
for registro in resultados:
    itemset = list(registro.items)
    suporte = registro.support
    dados_itemsets.append({
        'Itemset': itemset,
        'Suporte': suporte
    })

df_itemsets = pd.DataFrame(dados_itemsets)
df_itemsets = df_itemsets.sort_values(by='Suporte', ascending=False)

print("\nItemsets frequentes e seus suportes:")
print(df_itemsets.reset_index(drop=True))
```

Itemsets frequentes e seus suportes:

	Itemset	Suporte
0	[Cafe]	0.9
1	[Pao]	0.7
2	[Pao, Cafe]	0.7
3	[Manteiga]	0.6
4	[Arroz, Cafe]	0.5
5	[Cafe, Feijao]	0.5
6	[Pao, Cafe, Feijao]	0.4
7	[Pao, Feijao]	0.4
8	[Pao, Cafe, Manteiga]	0.4
9	[Arroz, Manteiga]	0.3
10	[Arroz, Feijao]	0.3
11	[Pao, Leite]	0.3
12	[Arroz, Cafe, Manteiga]	0.3
13	[Arroz, Feijao, Cafe]	0.3
14	[Pao, Arroz, Cafe]	0.3
15	[Pao, Cafe, Leite]	0.3

Regras de associação:

	Antecedente	Consequente	Suporte	Confiança	Lift
0	[Cafe, Leite]	[Pao]	0.3	1.000000	1.428571
1	[Feijao]	[Arroz]	0.3	0.600000	1.200000
2	[Arroz]	[Feijao]	0.3	0.600000	1.200000
...					
23	[Leite]	[Pao, Cafe]	0.3	0.750000	1.071429
24	[Leite]	[Pao]	0.3	0.750000	1.071429
25	[Arroz]	[Manteiga]	0.3	0.600000	1.000000
26	[Arroz, Cafe]	[Manteiga]	0.3	0.600000	1.000000

Questão 04 – Geração de regras com negação

```
# Função para criar itens de negação
def criar_itens_negados(df):
    negado_df = df.copy()
    for coluna in negado_df.columns:
        nova_coluna = f'NÃO_{coluna}'
        negado_df[nova_coluna] = negado_df[coluna].map({'Sim': 'Nao', 'Nao': 'Sim'})
    return negado_df

# Criar dataframe com itens negados
base_negada = criar_itens_negados(base)

# Construir transações considerando presença e ausência
transacoes = []
for _, linha in base_negada.iterrows():
    transacao = []
    for coluna in base_negada.columns:
        if linha[coluna] == 'Sim':
            transacao.append(coluna)
    transacoes.append(transacao)
```

```
# Filtrar regras com negação
regras_negacao = df_regras[df_regras['Regra'].str.contains('NÃO_')]

print("\nItemsets frequentes com negação:")
print(df_itemsets.head(10))

print("\n\nRegras de associação com ausência:")
print(regras_negacao.reset_index(drop=True).head(10))
```

Itemsets frequentes com negação:

	Itemset	Suporte
9	[NÃO_Cerveja, Manteiga]	0.5
37	[Cafe, NÃO_Cerveja, Manteiga]	0.5
10	[NÃO_Feijao, Manteiga]	0.4
16	[Arroz, NÃO_Leite, Cafe]	0.4
1	[Arroz, NÃO_Leite]	0.4
42	[Pao, Cafe, NÃO_Arroz]	0.4
39	[Cafe, NÃO_Leite, Manteiga]	0.4
69	[Pao, NÃO_Cerveja, Manteiga]	0.4
68	[NÃO_Leite, NÃO_Cerveja, Manteiga]	0.4
11	[NÃO_Feijao, NÃO_Arroz]	0.3

Regras de associação com ausência:

	Regra	Suporte	Confiança	Lift
0	Arroz, Cerveja → NÃO_Manteiga	0.2	1.0	2.5
1	Cafe, Cerveja → NÃO_Manteiga	0.3	1.0	2.5
2	Cerveja → Arroz, NÃO_Manteiga	0.2	0.5	2.5
3	Arroz, NÃO_Manteiga → Cerveja	0.2	1.0	2.5
4	NÃO_Manteiga → Arroz, Cerveja	0.2	0.5	2.5
5	NÃO_Manteiga → Cerveja, Pao	0.2	0.5	2.5
6	Cerveja → NÃO_Leite, NÃO_Manteiga	0.2	0.5	2.5
7	NÃO_Leite, NÃO_Manteiga → Cerveja	0.2	1.0	2.5
8	Cerveja, NÃO_Leite → NÃO_Manteiga	0.2	1.0	2.5
9	NÃO_Manteiga → Cerveja, NÃO_Leite	0.2	0.5	2.5

Questão 05 – Biblioteca mlxtend

- ``apriori(df, min_support, use_colnames=True)`` calcula conjuntos frequentes em um DataFrame onehot.
- ``association_rules(frequent_itemsets, metric='confidence', min_threshold=0.8)`` gera regras e métricas como lift, leverage e conviction. Vantagens: implementação vetorizada em pandas, várias métricas; Limitação: não gera negações, mas pode-se usar o mesmo truque da geração de regras com negação.

Questão 06 – Resenha do artigo

O artigo de Iztok Fister Jr. e colaboradores oferece uma análise abrangente das técnicas de visualização aplicadas à mineração de regras de associação (ARM - Association Rule Mining), um campo essencial dentro da mineração de dados e aprendizado de máquina. O foco está em como representar visualmente, de maneira eficaz, grandes conjuntos de regras geradas a partir de bases

transacionais, permitindo que os usuários compreendam melhor os padrões descobertos. O principal objetivo do artigo é reunir, classificar e avaliar métodos de visualização desenvolvidos para ARM, desde os mais tradicionais até as abordagens recentes. Além disso, os autores propõem uma taxonomia das técnicas, discutem os desafios e limitações atuais, e apontam direções promissoras para pesquisas futuras — especialmente sob a ótica da Inteligência Artificial Explicável (XAI).

Entre as principais contribuições do artigo, destaca-se a apresentação da evolução dos métodos de visualização de ARM, com a comparação de suas características, vantagens e desvantagens. Cada método é relacionado ao seu potencial de fornecer explicações mais compreensíveis, e novas ideias são introduzidas, como diagramas de Sankey, mapas de metrô e representações moleculares. Também é proposta uma taxonomia baseada em diferentes critérios, como tipo de dado (categórico, numérico, binário), número de medidas de interesse exibidas, tamanho do conjunto de regras e interatividade.

A metodologia utilizada envolveu uma revisão sistemática baseada em artigos encontrados nas bases ACM Digital Library, IEEE Xplore e Google Scholar, considerando critérios de inclusão como relevância ao tema, revisão por pares e idioma inglês. Os métodos de visualização discutidos incluem os tradicionais, como scatter plots, gráficos de matriz, gráficos em mosaico e gráficos em rede, além dos inovadores, como diagramas de Ishikawa (espinha de peixe), representações moleculares, mapas de metrô, diagramas de Sankey, plots em fita (ribbon) e visualizações baseadas em glifos. Esses métodos são avaliados quanto à capacidade de exibir múltiplas medidas (como suporte, confiança e lift), interatividade e tipos de dados suportados.

O artigo também destaca diversos desafios e problemas em aberto, como a dificuldade em lidar com conjuntos de regras muito grandes, a ausência de interatividade em muitos métodos inovadores, a falta de padronização para comparar métodos diferentes, limitações na visualização de atributos numéricos e temporais, e a pouca aplicação de conceitos de XAI no campo de ARM. Como direções futuras, os autores sugerem a integração com técnicas de aprendizado profundo e XAI, visualizações adaptativas baseadas no perfil do usuário, abordagens interativas e em tempo real, além de técnicas híbridas que combinem múltiplas formas de visualização.

Concluindo, o artigo é um recurso valioso para pesquisadores e profissionais que trabalham com mineração de dados, pois consolida décadas de avanços em um panorama organizado e crítico. Sua contribuição é ainda mais relevante por conectar o tema à crescente demanda por interpretabilidade e transparência na inteligência artificial.