

Lista 5 – IA

1\_

```
[(('Café',), ('Pão',), 0.3, 1.0),  
 (('Café',), ('Manteiga',), 0.3, 1.0),  
 (('Pão',), ('Manteiga',), 0.4, 0.8),  
 (('Manteiga',), ('Pão',), 0.4, 0.8),  
 (('Café',), ('Manteiga', 'Pão'), 0.3, 1.0),  
 (('Café', 'Pão'), ('Manteiga',), 0.3, 1.0),  
 (('Café', 'Manteiga'), ('Pão',), 0.3, 1.0)]
```

ItensSets 1: 3

ItensSets 2: 3

ItensSets 3: 1

Regras: 7

2\_

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

3\_

```
itemssets = []
suportes = []

for resultado in saida:
    s = resultado[1]
    itemset = resultado[0]
    itemssets.append(itemset)
    suportes.append(s)

itemssets_df = pd.DataFrame({'Itemset': itemssets, 'Suporte': suportes})

# Imprimir os itemsets gerados e seus respectivos suportes
print(itemssets_df)

# Estrutura para as regras finais
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': lift})
```

	Itemset	Suporte
0	(nan)	1.0
1	(Cafe, Manteiga)	0.3
2	(Cafe, Pao)	0.3
3	(Cafe, nan)	0.3
4	(Manteiga, Pao)	0.4
5	(nan, Manteiga)	0.5
6	(nan, Pao)	0.5
7	(Cafe, Manteiga, Pao)	0.3
8	(Cafe, Manteiga, nan)	0.3
9	(Cafe, Pao, nan)	0.3
10	(nan, Manteiga, Pao)	0.4
11	(Cafe, Manteiga, Pao, nan)	0.3

4\_

	Antecedente	Consequente	suporte	confianca	lift
0	leva Cafe	leva Manteiga	0.3	1.00	2.0
1	leva Manteiga	leva Cafe	0.3	0.60	2.0
2	leva Cafe	leva Pao	0.3	1.00	2.0
3	leva Pao	leva Cafe	0.3	0.60	2.0
4	leva Manteiga	leva Pao	0.4	0.80	1.6
5	leva Pao	leva Manteiga	0.4	0.80	1.6
6	leva Cafe	leva Manteiga e leva Pao	0.3	1.00	2.5
7	leva Manteiga	leva Cafe e leva Pao	0.3	0.60	2.0
8	leva Pao	leva Cafe e leva Manteiga	0.3	0.60	2.0
9	leva Cafe e leva Manteiga	leva Pao	0.3	1.00	2.0
10	leva Cafe e leva Pao	leva Manteiga	0.3	1.00	2.0
11	leva Manteiga e leva Pao	leva Cafe	0.3	0.75	2.5

5\_

A biblioteca mlxtend gera de regras de associação, utilizando principalmente o algoritmo Apriori. O processo inicia com a preparação dos dados, organizando-os em um DataFrame do Pandas onde cada linha representa uma transação e cada coluna um item, com valores binários indicando a presença ou ausência de itens. Em seguida, a função `apriori()` identifica itemsets frequentes, requerendo um parâmetro `min_support` para determinar a frequência mínima de ocorrência. Com os itemsets, a função `association_rules()` gera as regras de associação, permitindo a especificação de métricas como 'confidence' ou 'lift' e um limiar mínimo para elas. O resultado é um DataFrame com as regras, que os usuários podem analisar para descobrir padrões relevantes, como combinações de produtos frequentemente adquiridos.

6\_

Resenha sobre Mineração de Regras de Associação (ARM) e sua Visualização

A Mineração de Regras de Associação (ARM) é uma técnica de mineração de dados amplamente reconhecida por sua capacidade de descobrir relacionamentos ocultos em bancos de dados transacionais. Este campo, que se insere na área de Aprendizado de Máquina (ML), se concentra em identificar associações interessantes entre itens em grandes conjuntos de dados, permitindo que as relações sejam expressas através de regras de associação.

A história da ARM teve início com o trabalho seminal de Agrawal e Srikant em 1994, que estabeleceram as bases teóricas para essa técnica e propuseram o algoritmo Apriori, que ainda é considerado um dos principais algoritmos na mineração de dados. Com o passar dos

anos, muitos algoritmos baseados em métodos determinísticos, como Eclat e FP-Growth, foram desenvolvidos, embora métodos estocásticos baseados em população tenham ganhado destaque recentemente.

A ARM viu um aumento significativo de interesse dentro da comunidade de aprendizado de máquina, com aplicações práticas em diversas áreas, como análise de mercado, diagnósticos médicos e análise de dados censitários. O processo de ARM é complexo e envolve três etapas principais: pré-processamento, execução da ARM e pós-processamento. Essas etapas garantem que os dados sejam limpos e adequados para a mineração, resultando em um grande conjunto de regras de associação que requerem visualização para uma melhor interpretação.

Um aspecto emergente nesse campo é a explicabilidade em Inteligência Artificial (XAI). A XAI visa tornar os modelos de ML mais transparentes e compreensíveis, permitindo que usuários não especializados entendam as decisões tomadas pelos sistemas de IA. As técnicas de visualização desempenham um papel crucial nesse contexto, pois facilitam a compreensão dos resultados da ARM. Apesar da abundância de métodos visuais propostos, até o momento não havia uma revisão sistemática das abordagens de visualização sob a perspectiva da XAI.

Este artigo se propõe a preencher essa lacuna, apresentando e discutindo técnicas de visualização da ARM desenvolvidas ao longo do tempo. Os principais objetivos incluem a apresentação da evolução das técnicas de visualização, a definição das características de cada método, e a discussão de suas vantagens e desvantagens, acompanhada de exemplos práticos. A análise é baseada em publicações de fontes reconhecidas, como a ACM Digital Library e IEEEExplore, e é estruturada de forma a abordar a definição matemática da visualização da ARM, métodos tradicionais, novas ideias e taxonomias.

Em conclusão, a revisão proposta não apenas contribui para a literatura sobre visualização na ARM, mas também sugere direções para pesquisas futuras, visando aprimorar a interpretação dos resultados obtidos por algoritmos de mineração de regras de associação. Este estudo é particularmente relevante em um mundo onde a compreensão e a confiança nos sistemas de IA são cada vez mais necessárias.