Universidade Federal do Maranhão
Programa de Pós Graduação Em Ciência da Computação

Aluno
CARLOS EDUARDO NASCIMENTO CAJADO

Professor
Prof. Dr. LUCIANO REIS COUTINHO

CCOM0002 - APRENDIZAGEM DE MÁQUINA

Lista de Atividades 2 - Questão 4
Análise de associações
Apriori/FP-Growth

São Luís, MA
Julho- 2024

1. Introdução

1.1 Contexto

Segundo Facelli et al. [1], os dois principais algoritmos para a descoberta de regras de associação são o Apriori e o FP-Growth. O algoritmo Apriori calcula conjuntos de itens frequentes por meio de várias iterações, utilizando o conhecimento sobre conjuntos de itens infrequentes obtido de iterações anteriores para reduzir o conjunto de itens. Esse processo de poda elimina conjuntos de itens candidatos que não podem ser frequentes, com base na observação de que, se um conjunto de itens é frequente, todos os seus subconjuntos também devem ser. Antes de entrar na segunda etapa, o algoritmo descarta todos os conjuntos de itens candidatos que possuem um subconjunto infrequente [2].

O FP-Growth, por sua vez, é um algoritmo desenvolvido para extrair conjuntos de itens frequentes de forma mais eficiente, servindo como uma alternativa ao algoritmo Apriori [3]. De acordo com Han et al. [4], o FP-Growth, possui como característica principal não utilizar a geração de conjuntos de itens candidatos, utilizando em vez disso uma estrutura de dados compacta chamada FP-tree, o que resulta em um desempenho significativamente melhor em muitos casos.

1.2 problema

Neste trabalho, os algoritmos Apriori e FP-Growth serão empregados para gerar conjuntos de padrões frequentes, variando o grau de suporte. Em seguida, serão geradas regras de associação a partir desses padrões. Por fim, será realizada uma análise dos padrões e das regras obtidas.

1.3 objetivos

- Implementar os algoritmos Apriori e FP-Growth em bases de dados selecionadas.
- Avaliar o desempenho e a eficiência de ambos os algoritmos na geração de conjuntos de padrões frequentes, variando o grau de suporte mínimo.
- Gerar regras de associação a partir dos conjuntos de padrões frequentes obtidos.

1.4 Configuração do Ambiente

Para desenvolvimento do projeto, temos as seguintes configurações:

1.3.1 Software:

- Windows 11
- Linguagem python versão 3.12.2
- Jupyter Notebook
- Bibliotecas mlxtend, scikitlearn, pandas, numpy, matplotlib, seaborn.

1.3.2 Hardware:

- Processador: Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz
- Memória RAM: 16,0 GB (utilizável: 15,9 GB)

2. Bases de Dados

2.1 Descrição

- Título da base : Cesta de compras

Desenvolvido a partir das principais técnicas usadas por grandes varejistas para descobrir associações entre itens. Funciona procurando combinações de itens que ocorrem juntos com frequência nas transações. Em outras palavras, permite que os varejistas identifiquem relações entre os itens que as pessoas compram. Funciona procurando combinações de itens que ocorrem juntos com frequência nas transações. Em outras palavras, permite que os varejistas identifiquem relações entre os itens que as pessoas compram. Disponível em kaggle [5].

2.2 Estatísticas

O conjunto de dados possui 38.765 linhas de pedidos de compra de pessoas em supermercados. Essas ordens podem ser analisadas e regras de associação podem ser geradas usando Market Basket Analysis por algoritmos como o Algoritmo Apriori [5].

3. Preparação (Pré-processamento)

3.1 Seleção

Foi desenvolvido um algoritmo que se utiliza a biblioteca “mlxtend” para preparar dados transacionais no formato adequado para a aplicação de algoritmos de mineração de regras de associação.

Figura 1: Pré-processamento

```
from mlxtend.preprocessing import TransactionEncoder

# Transformando os dados para uma lista de listas
transactions = Dataset.apply(lambda x: x.dropna().tolist(), axis=1).tolist()

# Transformar os dados para o formato binário
te = TransactionEncoder()
te_ary = te.fit(transactions).transform(transactions)
df_binary = pd.DataFrame(te_ary, columns=te.columns_)
df_binary = df_binary.astype(int)
```

Autor: próprio

3.2 Limpeza dos dados

Inicialmente, os dados são transformados em uma lista de listas, onde cada sub-lista contém os itens de uma transação, removendo valores nulos. Em seguida, se converte esses dados em um formato binário (one-hot encoding), onde cada coluna representa um item e cada linha uma transação, com valores 1 indicando a presença do item e 0 a ausência.

4 Experimentos e Resultados

4.1 Padrões frequentes com Apriori

Partindo das funções apriori e association_rules do módulo mlxtend.frequent_patterns, será utilizado para descobrir padrões frequentes e gerar regras de associação a partir dos dados pré-processados.Onde, inicialmente, a função “apriori” é utilizada para identificar conjuntos de itens que aparecem frequentemente juntos no dataset, com um suporte mínimo de 5%. Esses conjuntos de itens frequentes são então passados para a função “association_rules”, que gera regras de associação baseadas nesses conjuntos.

As regras são criadas com base na métrica de confiança, que, basicamente, mede a probabilidade de ocorrência dos itens dado que os itens antecedentes já ocorreram, nos experimentos será utilizado um limiar mínimo de 20%.

Figura 2: Padrões frequentes com Apriori

```python
from mlxtend.frequent_patterns import apriori, association_rules

# Gerar padrões frequentes com Apriori
frequent_itemsets_apriori = apriori(df_binary, min_support=0.05, use_colnames=True)

# Gerar regras de associação
rules_apriori = association_rules(frequent_itemsets_apriori, metric="confidence", min_threshold=0.2)

print("Padrões frequentes usando Apriori:")
print(frequent_itemsets_apriori)
print("Regras de associação usando Apriori:")
print(rules_apriori)
```

Autor: próprio

Os resultados podem ser analisados de forma mais detalhada observando a Figura 3, que apresenta a lista dos padrões frequentes identificados pelo algoritmo Apriori. Nessa figura, podemos visualizar os itens frequentemente comprados juntos, como leite integral (whole milk), vegetais diversos (other vegetables), pão (rolls/buns), refrigerante (soda), e frutas tropicais (tropical fruit), cada um com seu respectivo suporte indicando a frequência de ocorrência no conjunto de dados.

Figura 3: Lista dos padrões frequentes

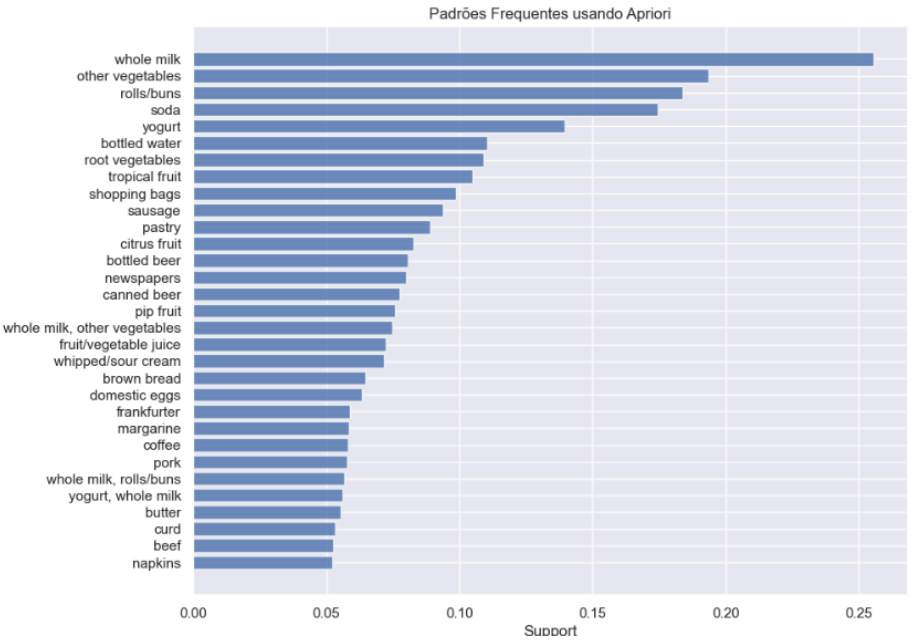
```
Padrões frequentes usando Apriori:
     support                        itemsets
0   0.052466                          (beef)
1   0.080529                   (bottled beer)
2   0.110524                  (bottled water)
3   0.064870                    (brown bread)
4   0.055414                         (butter)
5   0.077682                   (canned beer)
6   0.082766                  (citrus fruit)
7   0.058058                         (coffee)
8   0.053279                           (curd)
9   0.063447                  (domestic eggs)
10  0.058973                    (frankfurter)
11  0.072293          (fruit/vegetable juice)
12  0.058566                       (margarine)
13  0.052364                         (napkins)
14  0.079817                    (newspapers)
15  0.193493               (other vegetables)
16  0.088968                         (pastry)
17  0.075648                       (pip fruit)
18  0.057651                            (pork)
19  0.183935                      (rolls/buns)
20  0.108998                (root vegetables)
21  0.093950                        (sausage)
22  0.098526                  (shopping bags)
23  0.174377                            (soda)
24  0.104931                (tropical fruit)
25  0.071683             (whipped/sour cream)
26  0.255516                       (whole milk)
27  0.139502                          (yogurt)
28  0.074835  (whole milk, other vegetables)
29  0.056634        (whole milk, rolls/buns)
30  0.056024           (yogurt, whole milk)
```

Autor: próprio

Ademais, na Figura 4, é apresentado um gráfico que ilustra a distribuição dos itens da

cesta de compras de acordo com seu suporte. Objetivando uma comparação rápida entre os diferentes itens em termos de popularidade, destacando aqueles que têm maior e menor suporte dentro do conjunto de dados analisado.

Figura 4: Gráfico distribuição dos itens da cesta de compras/suporte



Autor: próprio

4.2 Padrões frequentes com FP-Growth

Na utilização do FP-Growth, tens-se resultados semelhantes , Figura 5 e 6

Figura 6: Lista de frequentes com FP-Growth

Padrões frequentes usando FP-Growth:

	support	itemsets
0	0.082766	(citrus fruit)
1	0.058566	(margarine)
2	0.139502	(yogurt)
3	0.104931	(tropical fruit)
4	0.058058	(coffee)
5	0.255516	(whole milk)
6	0.075648	(pip fruit)
7	0.193493	(other vegetables)
8	0.055414	(butter)
9	0.183935	(rolls/buns)
10	0.080529	(bottled beer)
11	0.110524	(bottled water)
12	0.053279	(curd)
13	0.052466	(beef)
14	0.174377	(soda)
15	0.058973	(frankfurter)
16	0.079817	(newspapers)
17	0.072293	(fruit/vegetable juice)
18	0.088968	(pastry)
19	0.108998	(root vegetables)
20	0.077682	(canned beer)
21	0.093950	(sausage)
22	0.098526	(shopping bags)
23	0.064870	(brown bread)
24	0.052364	(napkins)
25	0.071683	(whipped/sour cream)
26	0.057651	(pork)
27	0.063447	(domestic eggs)
28	0.056024	(yogurt, whole milk)
29	0.074835	(whole milk, other vegetables)
30	0.056634	(whole milk, rolls/buns)

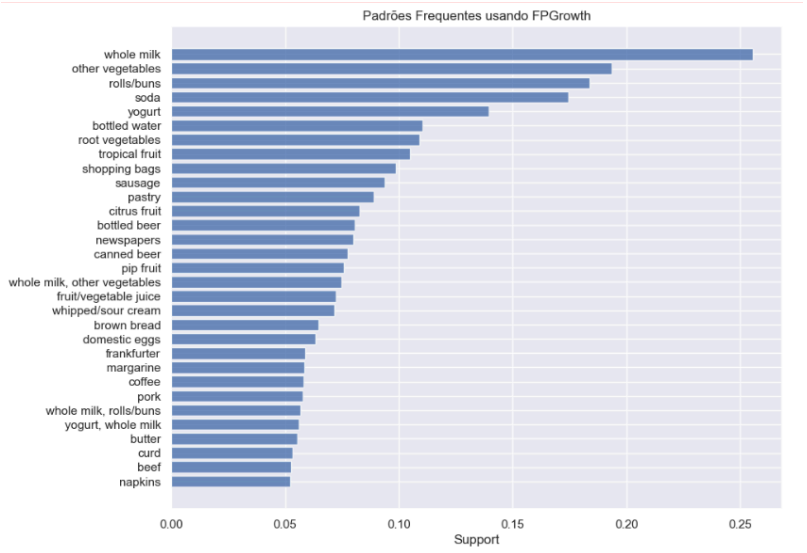
Regras de associação usando FP-Growth:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(yogurt)	(whole milk)	0.139502	0.255516	0.056024	0.401603	1.571735	0.020379	1.244132
1	(whole milk)	(yogurt)	0.255516	0.139502	0.056024	0.219260	1.571735	0.020379	1.102157
2	(whole milk)	(other vegetables)	0.255516	0.193493	0.074835	0.292877	1.513634	0.025394	1.140548
3	(other vegetables)	(whole milk)	0.193493	0.255516	0.074835	0.386758	1.513634	0.025394	1.214013
4	(whole milk)	(rolls/buns)	0.255516	0.183935	0.056634	0.221647	1.205032	0.009636	1.048452
5	(rolls/buns)	(whole milk)	0.183935	0.255516	0.056634	0.307905	1.205032	0.009636	1.075696

Autor: próprio

Tens-se o Gráfico de distribuição :

Figura 8: Gráfico distribuição dos itens da cesta de compras/suporte FP-Growth



Autor: próprio

6. Conclusão

Com base nos estudos utilizando os algoritmos Apriori e FP-Growth utilizados nesta pesquisa, foi possível explorar e comparar a geração de conjuntos de padrões frequentes ao variar o grau de suporte mínimo. Sendo a análise dos resultados revelou a habilidade desses algoritmos em identificar combinações frequentes de itens em conjuntos de dados específicos, proporcionando uma visão detalhada dos padrões gerados pelo Apriori. Este método não apenas demonstrou a eficácia desses métodos na identificação de padrões e na geração de regras de associação, mas também enfatizou a importância de ajustar o suporte mínimo para capturar padrões relevantes e aplicáveis em contextos comerciais e empresariais.

7. Referências

[1] Facelli, K.; Lorena A.C; Gama. J. & Cavalho A.C.P.L.F (2011) Inteligência artificial: uma abordagem de aprendizado de máquina. Rio de Janeiro, RJ: LTC, 2011. xvi, 378 p. ISBN 9788521618805

[2] Kantardzic M. (2011). Data Mining: Concepts, Models, Methods, and Algorithms. Wiley-IEEE Press; 2nd ed. edição. ISBN 9780470890455

[3] DE ARAÚJO, Ramon Batista et al. Regras de associação entre as características dos veículos e os acidentes de trânsito em rodovias federais brasileiras através de aprendizado de máquina. 2022.

[4] Han, J., Pei, J., Yin, Y. et al. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. Data Mining and Knowledge Discovery 8, 53–87 (2004). https://doi.org/10.1023/B:DAMI.0000005258.31418.83

[5] https://www.kaggle.com/datasets/heeraldedhia/groceries-dataset