DATA SCIENCE

AULA 8 - Aprendizagem Não- supervisionada IV

Prof. Gabriel Resende Machado

gabrielmachado@unifeso.edu.com

https://www.linkedin.com/in/machadogabriel

https://github.com/UNIFESO-Gabriel/data-science

Motivação

- **Cenário de exemplo:** como aumentar as vendas de um supermercado?
  - Induzir os clientes a comprarem produtos relacionados, colocando-os em prateleiras próximas.
  - Exemplos são: (i) pão e manteiga; (ii) carvão, carne e cerveja, (iii) fralda e cerveja entre outros;
  - **Como saber quais produtos estão relacionados?**

Algoritmo Apriori

- **Algoritmo pertencente à classe de algoritmos de associação;**

- Seu objetivo é encontrar elementos que implicam na presença de outros elementos em uma mesma transação. Ou seja, encontrar **relacionamentos** ou padrões frequentes entre conjuntos de dados;
- A compra de um produto quando outro é comprado, por exemplo, representa uma **regra de associação**;
- Regras de associação são frequentemente utilizadas em áreas como *marketing*, mas também em outras áreas, como descrever falhas em linhas de comunicação, ações na interface do usuário, entre outros.

Algoritmo Apriori

- **Tomemos como exemplo o seguinte *dataset*:**
  - As linhas remetem à transações e o número 1 que o respectivo item foi comprado.

ID	Beer	Diaper	Gum	Soda	Snack
1	1	1	1	1	0
2	1	1	0	0	0
3	1	1	1	0	1
4	1	1	0	1	1
5	0	1	0	1	0
6	0	1	0	0	0
7	0	1	0	0	0
8	0	0	0	1	1
9	0	0	0	1	1

Regras de Associação

- Regras de associação nada mais são do que um **método** para explorar itens em um conjunto de dados;

- Define-se os seguintes termos para as regras de associação:

  - **I (Itens):** conjunto dos seus n atributos $\{i_1, i_2, \ldots, i_n\}$;

  - **D (Database):** conjunto das *m* transações $\{t_1, t_2, \ldots, t_m\}$;

  - Toda transação $\mathbf{t_i}$ **é única em D** e consiste em um subconjunto dos Itens I;

  - Define-se a **regra de associação** (X -> Y) , onde X e Y são subconjuntos de I. Eles não podem ter nenhum elemento em comum.

  - **X é chamado de antecedente e Y da consequência da Regra.**

Regras de Associação

- Exemplo de transação é a regra de ID = 2: **{Beer} -> {Diaper}.**
  - Exprime a ideia de quem comprou cerveja, também comprou fralda.
- Mesmo um conjunto pequeno pode gerar uma quantidade muito grande de regras.
- Como identificar as regras mais relevantes?
  - *Support* (Suporte);
  - *Confidence* (Confiança);
  - *Lift* (Alavancagem);
  - *Conviction* (Convicção).

Métricas

- ***Support* (Suporte):**
    - Este número é a popularidade do item no conjunto de dados estudado. Sendo assim esse número é definido pela quantidade de vezes que o item apareceu em uma transação, dividido pela quantidade de transações existentes no dataset;
    - **supp(x)** = (número de transações que X aparece) / (número total de transações);
    - Ou seja, **Supp({Beer}) = 4/9 = 0,4444.**
- Algo importante a ser citado neste momento é o support threshold. Este número irá expressar a quantidade mínima que o suporte de um item (ou conjunto de itens) deve aparecer para ele se tornar significante.

Métricas

- ***Confidence* (Confiança):**
  - A confiança é um número que expressa a possibilidade de um item ser comprado quando outro item correlato é comprado;
  - Por exemplo, qual a confiança que um cliente irá comprar um hambúrguer considerando que ele já comprou cebolas e batatas;
  - A confiança é calculada por meio da seguinte equação:

$$conf(X \rightarrow Y) = supp(X \cup Y)/supp(X)$$

  - Ou seja, **Conf({Beer}=>{Gum}) = (2/9)/(4/9) = 1/2.**

Métricas

- ***Lift* (Alavancagem):**
  - Esta medida também calcula a possibilidade de um item ser comprado em relação a outro item.
  - Porém, esta medida considera a **popularidade de ambos os itens**.
  - A alavancagem é calculada por meio da seguinte equação:

$$lift(X \rightarrow Y) = supp(X \cup Y)/supp(X) * supp(Y)$$

  - No caso da regra, **({Beer}=>{Gum}), Lift({Beer}=>{Gum}) = (2/9)/((4/9)*(2/9)) = 2/3.**

Métricas

- ***Conviction* (Convicção):**
  - Essa medida está interessada em calcular a frequência que X ocorre e Y não ocorre, ou seja, ela está interessada em quando a Regra falha.

$$conv(X \rightarrow Y) = 1 - supp(X)/1 - conf(X \rightarrow Y)$$

  - Essa medida varia entre [0, inf). Se Conf(X->Y) for igual a 1, então o denominador da fórmula é zerado e o resultado da Convicção é definido como infinito.
  - Já se o Supp(Y) for igual a 1, ou seja, Y é presente em todas as transações, então Convicção é igual a 0 — não há erro.
- **X = {Soda, Snack} e Y = {Beer}**
  - Conv(X=>Y) = ((9/9) — (4/9)) / ((9/9) — (3/9)) = 5/6

Algoritmo Apriori

- O Apriori trabalha com o conceito de itens frequentes que são os itens do seu conjunto **I** que têm a pontuação do Support superior a de um limiar (fornecido como hiperparâmetro);

- É preciso calcular o *Support* de todas as combinações de itens e extrair um subconjunto de itens frequentes.

- Os passos do algoritmo Apriori são:

    - Dentro dos Itens **I,** extraia um subconjunto (**I_freq**) dos itens que têm o seu *Support* maior que o limiar;

    - Dentro de **I_freq**, itere para formar as combinações dos **I_freq** com **I**, aplique o limiar e acumule em **I_freq;**

    - Pare quando não sobrar nenhum item após aplicar o limiar.

Algoritmo Apriori

- Por exemplo, vamos calcular o *Support* para grupos de um item, com limiar de 0,4:

support	itemsets	length
0.444444	(Beer)	1
0.777778	(Diaper)	1
0.222222	(Gum)	1
0.555556	(Soda)	1
0.444444	(Snack)	1

- Agora, são removidos os Itens com *Support* menor que o limiar, ou seja, o chiclete é removido:

support	itemsets	length
0.444444	(Beer)	1
0.777778	(Diaper)	1
0.555556	(Soda)	1
0.444444	(Snack)	1

Algoritmo Apriori

- Depois disso, é realizado o mesmo cálculo para os pares de Itens, como por exemplo (*Diaper, Beer*):

support	itemsets	length
0.444444	(Diaper, Beer)	2
0.222222	(Gum, Beer)	2
0.222222	(Soda, Beer)	2
0.222222	(Snack, Beer)	2
0.222222	(Gum, Diaper)	2
0.333333	(Soda, Diaper)	2
0.222222	(Snack, Diaper)	2
0.111111	(Gum, Soda)	2
0.111111	(Gum, Snack)	2
0.333333	(Soda, Snack)	2

- Agora, aplicando o limiar, sobra apenas *(Diaper, Beer),* que deve ser acumulado em **I_freq**, como mostrado a seguir:

support	itemsets	length
0.444444	(Beer)	1
0.777778	(Diaper)	1
0.555556	(Soda)	1
0.444444	(Snack)	1
0.444444	(Diaper, Beer)	2

Algoritmo Apriori

- Depois disso, é realizado o mesmo cálculo para grupos de três Itens, como por exemplo (*Diaper, Beer*):

support	itemsets	length
0.222222	(Gum, Diaper, Beer)	3
0.222222	(Soda, Diaper, Beer)	3
0.222222	(Snack, Diaper, Beer)	3
0.111111	(Gum, Soda, Beer)	3
0.111111	(Gum, Snack, Beer)	3
0.111111	(Soda, Snack, Beer)	3
0.111111	(Gum, Diaper, Soda)	3
0.111111	(Gum, Diaper, Snack)	3
0.111111	(Soda, Diaper, Snack)	3
0.000000	(Gum, Snack, Soda)	3

- Após aplicar o limiar de 0,4, percebe-se que não sobra nenhum item. Logo, o algoritmo para, com **I_freq** formado a seguir:

support	itemsets	length
0.444444	(Beer)	1
0.777778	(Diaper)	1
0.555556	(Soda)	1
0.444444	(Snack)	1
0.444444	(Diaper, Beer)	2

→ Vale a pena deixar as fraldas e cervejas em prateleiras próximas?

Algoritmo Apriori - Outro exemplo

- Considere os seguintes itens comprados por um cliente em um supermercado:
  - 0 representa *não comprou* e 1 representa *comprou*;
  - define-se um limiar menor ou igual 50% para o *support*.

ID	CEBOLA	BATATA	HAMBÚRGUER	LEITE	CERVEJA
t1	1	1	1	0	0
t2	0	1	1	1	0
t3	0	0	0	1	1
t4	1	1	0	1	0
t5	1	1	1	0	1
t6	1	1	1	1	1

Algoritmo Apriori - Outro exemplo

- Cria-se uma tabela que representa a frequência de cada item e aplica-se o limiar pré-definido:

Item	Suporte
Cebola	4/6
Batata	5/6
Hambúrguer	4/6
Leite	4/6
Cerveja	3/6

Item	Suporte
Cebola	4/6
Batata	5/6
Hambúrguer	4/6
Leite	4/6

Algoritmo Apriori - Outro exemplo

- Agora é preciso realizar todas as combinações possíveis dentre os três itens e cria-se uma tabela contendo a combinação e quantas vezes essa combinação aparece nas transações:
    - Elimina-se os itens que estão abaixo do limiar pré-definido.

Item	Suporte
Cebola + Batata	4/6
Cebola + Hambúrguer	3/6
Cebola + Leite	2/6
Batata + Hambúrguer	4/6
Batata + Leite	3/6
Hambúrguer + Leite	2/6

Item	Suporte
Cebola + Batata + Hambúrguer	3/6
Batata + Leite + Hambúrguer	2/6

Algoritmo Apriori - Exemplo

- Encontre as principais regras de associação entre os principais produtos de um supermercado. Utilize o algoritmo Apriori. Notebook em https://shorturl.at/bxAJS.

Algoritmo Apriori - Exercício

- Encontre as principais regras de associação entre os principais produtos de um supermercado. Utilize o algoritmo Apriori. Notebook em https://shorturl.at/dwBSX.

DATA SCIENCE

AULA 7 - Aprendizagem Não-Supervisionada III

Dúvidas e/ou perguntas?

gabrielmachado@unifeso.edu.com

https://www.linkedin.com/in/machadogabriel

https://github.com/UNIFESO-Gabriel/data-science