

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
SISTEMAS DE INFORMAÇÃO

CARLOS DANIEL

ELTON LUNARDI

NICK ISHIDA

PABLO SOAREZ

TRABALHO PRÁTICO DE REGRAS DE ASSOCIAÇÃO

MONTE CARMELO

2021

SUMARIO

1. INTRODUÇÃO.....	3
2. <u>EXECUÇÃO</u> DAS TAREFAS.....	3
2.1. Eliminação das instâncias que não atendem aos valores.....;	3
2.2. Eliminar os atributos de 1 a 23.....	5
2.3. Eliminação dos atributos que não possuam suporte superior ou igual a 10%.....	7
2.4. Execução do Método à Priori.....	7
2.5. Ordenação por Lift.....	8
3. ANÁLISE DOS RESULTADOS.....	8
3.1. Apresentação das regras encontradas.....	8
3.2. Análise dos resultados.....	9

1. Introdução do Problema

Nos foi apresentado o dataset com 1540 instancias e 758 atributos no total com nome de “Mercado.arff” relacionado à um banco de dados de um mercado, contendo informações de todos os produtos vendidos neste mercado, e, a quantidade de suas respectivas vendas.

No entanto, visando o trabalho com as regras de associação, é necessário, primeiramente, uma limpeza desses dados, não só para articulação do programa utilizado para geração das regras o “Weka” na versão 3.8.6, e também para melhor resolução das atividades.

Dessa forma, o dataset possuindo os valores indesejados para nossos objetivos e parâmetros que serão aplicados, fá-lo-emos a análise e definição das regras somente após a apuração e pré-processamento dessas instancias.

2. Execução das Tarefas

Primeiramente, realizamos o pré-processamento no qual foi requisitado.

- Eliminação das instâncias que não atendem aos valores: city_belo_horizonte = 1; city_recife = 1; city_goiania = 1;

Posteriormente, utilizamos um dos filtros já pré-existentes no Weka para a remoção de instancias que compreendem valores nulos, o “RemoveWithValues”.

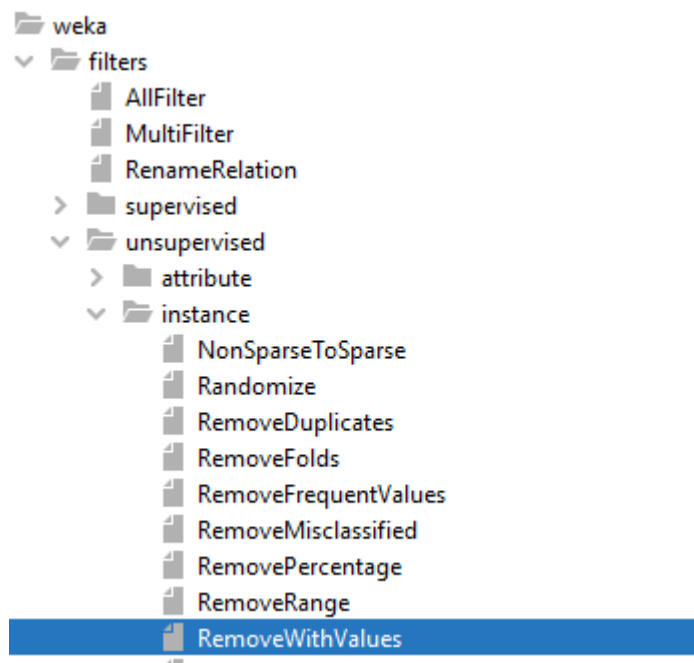


Imagem 1: Filtro utilizado selecionado e sua localização nas pastas da aplicação.

Com este filtro, é possível realizar a limpeza de cada uma das instâncias, adaptando as configurações do filtro para nosso objetivo, como pode ser visto na Imagem 2.

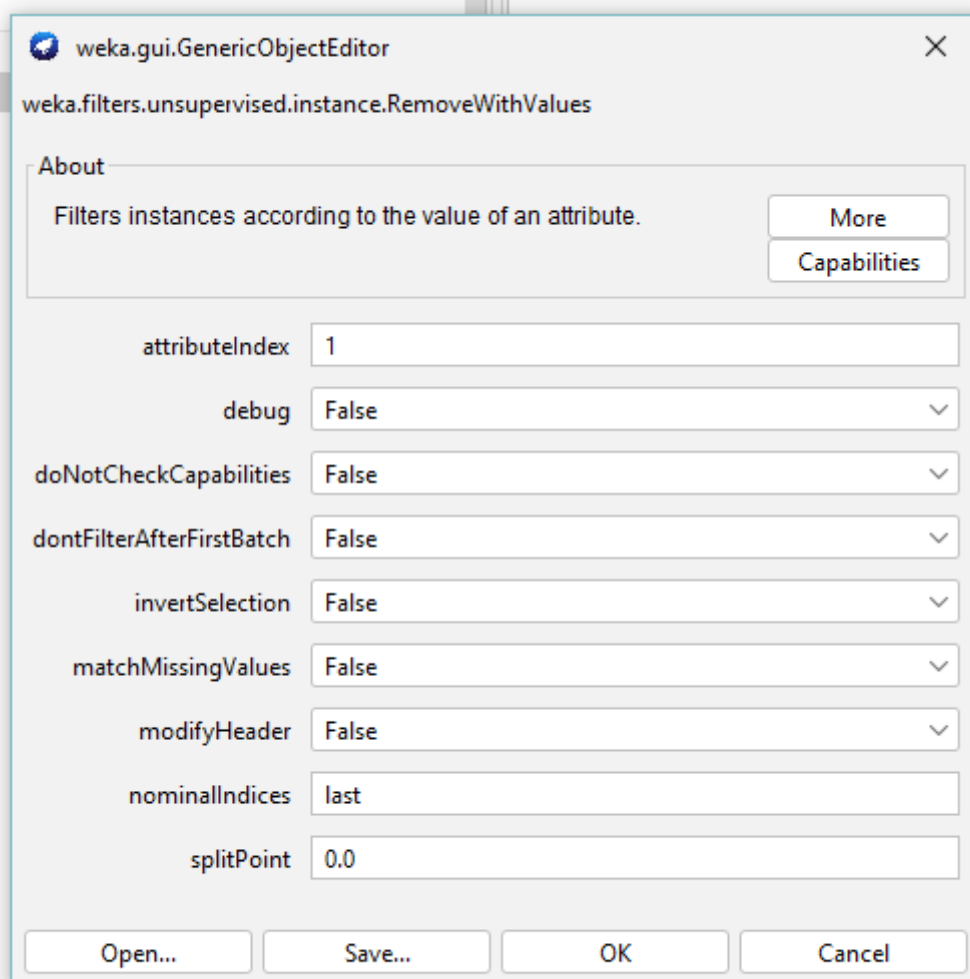


Imagem 2: Exemplo de configuração para eliminação.

As imagens 3.1, 3.2 e 3.3 representam o resultado das exclusões.

Selected attribute				
Name: City_Belo_Horizonte		Distinct: 2		Type: Nominal
Missing: 0 (0%)				Unique: 0 (0%)
No.	Label	Count	Weight	
1	0	381	381	
2	1	206	206	

Imagem 3.1: Atributo City_Belo_Horizonte após a exclusão das instâncias.

Selected attribute			
Name: City_Goiania		Type: Nominal	
Missing: 0 (0%)		Unique: 0 (0%)	
		Distinct: 2	
No.	Label	Count	Weight
1	0	406	406
2	1	181	181

Imagem 3.2: Atributo City_Goiania após a exclusão das instâncias.

Selected attribute			
Name: City_Recife		Type: Nominal	
Missing: 0 (0%)		Unique: 0 (0%)	
		Distinct: 2	
No.	Label	Count	Weight
1	0	387	387
2	1	200	200

Imagem 3.3: Atributo City_Recife após a exclusão das instâncias.

- Eliminar os atributos de 1 a 23;

Dessa vez, utilizando o filtro “Remove” do Weka, selecionamos todas os atributos (1 a 23) como está representado na imagem 4.1 e o seu resultado na 4.2.

Attributes

No.	Name
1	<input checked="" type="checkbox"/> City_Belem
2	<input checked="" type="checkbox"/> City_Belo_Horizonte
3	<input checked="" type="checkbox"/> City_Curitiba
4	<input checked="" type="checkbox"/> City_Florianopolis
5	<input checked="" type="checkbox"/> City_Fortaleza
6	<input checked="" type="checkbox"/> City_Goiania
7	<input checked="" type="checkbox"/> City_Porto_Alegre
8	<input checked="" type="checkbox"/> City_Recife
9	<input checked="" type="checkbox"/> Members_1
10	<input checked="" type="checkbox"/> Members_2
11	<input checked="" type="checkbox"/> Members_3
12	<input checked="" type="checkbox"/> Members_4
13	<input checked="" type="checkbox"/> Members_5
14	<input checked="" type="checkbox"/> Members_6
15	<input checked="" type="checkbox"/> Members_above_6
16	<input checked="" type="checkbox"/> Income_below_2.5
17	<input checked="" type="checkbox"/> Income_2.5_to_5
18	<input checked="" type="checkbox"/> Income_5_to_8
19	<input checked="" type="checkbox"/> Income_8_to_12
20	<input checked="" type="checkbox"/> Income_12_to_18
21	<input checked="" type="checkbox"/> Income_18_to_25
22	<input checked="" type="checkbox"/> Income_25_to_43
23	<input checked="" type="checkbox"/> Income_above_43
24	<input type="checkbox"/> acai_berry
25	<input type="checkbox"/> acai_berry_pulp
26	<input type="checkbox"/> ajinomoto_sauce
27	<input type="checkbox"/> alphabet_pasta

Imagem 4.1: Seleção dos atributos a serem retirados.

1	<input type="checkbox"/> acai_berry
2	<input type="checkbox"/> acai_berry_pulp
3	<input type="checkbox"/> ajinomoto_sauce
4	<input type="checkbox"/> alphabet_pasta
5	<input type="checkbox"/> amazon_papaya
6	<input type="checkbox"/> anchovy
7	<input type="checkbox"/> anjou_pear
8	<input type="checkbox"/> annatto
9	<input type="checkbox"/> argentinian_apple
10	<input type="checkbox"/> arrowroot_powder
11	<input type="checkbox"/> asparagus
12	<input type="checkbox"/> assorted_candies
13	<input type="checkbox"/> atlantic_seabob_shrimp
14	<input type="checkbox"/> aubergine
15	<input type="checkbox"/> avocado
16	<input type="checkbox"/> baby_potato
17	<input type="checkbox"/> bacon
18	<input type="checkbox"/> bahia_beans
19	<input type="checkbox"/> banana
20	<input type="checkbox"/> banana_other_kind
21	<input type="checkbox"/> basella
22	<input type="checkbox"/> basil
23	<input type="checkbox"/> battered_beef
24	<input type="checkbox"/> battered_chicken_breast
25	<input type="checkbox"/> battered_chicken_drumstick

Imagem 4.2: Representação parcial dos atributos após a exclusão.

- Eliminação dos atributos que não possuam suporte superior ou igual a 10%;

Após as exclusões posteriormente citadas, reduzimos uma pequena parcela do dataset original, agora, visando uma eliminação maior das instancias, é feito, orientado ao suporte, no nosso caso, maior ou igual a 10%. Como nosso dataset possuía algo em torno de 580 instancias, quaisquer vendas com valor inferior à 58 seriam eliminados.

Já a metodologia para eliminação, foi feita manualmente, olhando cada item para decidir seu destino.

Após isso, saímos de 735 instancias para 97. Agora, já podemos prosseguir para as associações.

- Execução do Método à Priori;

A primeira das considerações do método à priori é definir valores igual a 0 como ausentes, este, que foi feito no campo “treatZeroAsMissing” setando seu valor como Verdadeiro para realização da tarefa.

A próxima consideração é o teste com valores de “loweBoundMinSupport” (Suporte mínimo) variando de 0.1 à 0.5 (10% a 50%), para análise da confiança foi necessário selecionar a opção “Confidence” no campo “metricType” que corresponde ao tipo de métrica a ser utilizada, dessa forma, o “minMetric” (Confiança mínima) seta os valores, no nosso caso, variando de 0.6 à 0.9 (60% a 90%).

A ultima das considerações é a geração e seleção de 20 regras, modificação essa também constada no campo “numRules” que define a quantidade de regras que exigimos, todas arquitetadas automaticamente pela aplicação.

- Ordenação por Lift

Já a ordenação por lift, é feito de maneira semelhante à confiança, alternando para opção “lift” dentro do “metricType” citado anteriormente.

Todos os campos modificados e alterados foram contemplados na imagem abaixo.

About

Class implementing an Apriori-type algorithm. More Capabilities

car False

classIndex -1

delta 0.05

doNotCheckCapabilities False

lowerBoundMinSupport 0.5

metricType Confidence

minMetric 0.9

numRules 20

outputItemSets False

removeAllMissingCols False

significanceLevel -1.0

treatZeroAsMissing True

upperBoundMinSupport 1.0

verbose False

Imagem 5: Campos utilizados para as operações de associação

3 Análise dos Resultados

- Apresentação das regras encontradas

Visando uma maior pureza e confiabilidade nos resultados, optamos por aqueles que resultaram em 20 regras e que obedecem aos maiores valores citados anteriormente, dessa forma, com suporte mínimo de 0.5 e confiança de 0.6. Houveram outras formações de regras com valores de suporte ou confiança maiores, mas optamos por aqueles que resultaram o número de regras pedido. As regras geradas conferem na imagem abaixo.


```

Apriori
=====

Minimum support: 0.5 (294 instances)
Minimum metric <confidence>: 0.6
Number of cycles performed: 1

Generated sets of large itemsets:

Size of set of large itemsets L(1): 9

Size of set of large itemsets L(2): 11

Best rules found:

1. granulated_sugar=1 349 ==> french_bread=1 309    <conf:(0.89)> lift:(1.04) lev:(0.02) [12] conv:(1.29)
2. mayonnaise=1 339 ==> french_bread=1 296    <conf:(0.87)> lift:(1.03) lev:(0.01) [8] conv:(1.17)
3. soy_oil=1 438 ==> french_bread=1 381    <conf:(0.87)> lift:(1.03) lev:(0.02) [9] conv:(1.14)
4. garlic=1 401 ==> french_bread=1 348    <conf:(0.87)> lift:(1.02) lev:(0.01) [7] conv:(1.13)
5. vinegar=1 414 ==> french_bread=1 358    <conf:(0.86)> lift:(1.02) lev:(0.01) [6] conv:(1.1)
6. egg=1 429 ==> french_bread=1 370    <conf:(0.86)> lift:(1.02) lev:(0.01) [6] conv:(1.08)
7. garlic=1 401 ==> soy_oil=1 314    <conf:(0.78)> lift:(1.05) lev:(0.03) [14] conv:(1.16)
8. egg=1 429 ==> soy_oil=1 330    <conf:(0.77)> lift:(1.03) lev:(0.02) [9] conv:(1.09)
9. french_bread=1 498 ==> soy_oil=1 381    <conf:(0.77)> lift:(1.03) lev:(0.02) [9] conv:(1.07)
10. soy_oil=1 438 ==> egg=1 330    <conf:(0.75)> lift:(1.03) lev:(0.02) [9] conv:(1.08)
11. vinegar=1 414 ==> soy_oil=1 311    <conf:(0.75)> lift:(1.01) lev:(0) [2] conv:(1.01)
12. garlic=1 401 ==> egg=1 300    <conf:(0.75)> lift:(1.02) lev:(0.01) [6] conv:(1.06)
13. french_bread=1 498 ==> egg=1 370    <conf:(0.74)> lift:(1.02) lev:(0.01) [6] conv:(1.04)
14. vinegar=1 414 ==> egg=1 305    <conf:(0.74)> lift:(1.01) lev:(0) [2] conv:(1.01)
15. french_bread=1 498 ==> vinegar=1 358    <conf:(0.72)> lift:(1.02) lev:(0.01) [6] conv:(1.04)
16. soy_oil=1 438 ==> garlic=1 314    <conf:(0.72)> lift:(1.05) lev:(0.03) [14] conv:(1.11)
17. egg=1 429 ==> vinegar=1 305    <conf:(0.71)> lift:(1.01) lev:(0) [2] conv:(1.01)
18. soy_oil=1 438 ==> vinegar=1 311    <conf:(0.71)> lift:(1.01) lev:(0) [2] conv:(1.01)
19. egg=1 429 ==> garlic=1 300    <conf:(0.7)> lift:(1.02) lev:(0.01) [6] conv:(1.05)
20. french_bread=1 498 ==> garlic=1 348    <conf:(0.7)> lift:(1.02) lev:(0.01) [7] conv:(1.05)

```

Imagem 6: Resultados do dataset com valores específicos.

- Análise dos resultados

Analisando as regras obtidas, percebe-se que o pão francês se mostra relacionado no tocante do: açúcar granulado, maionese, óleo de soja, vinagre e ovo; dessa forma, compreende-se que quando o cliente compra um dos itens anteriores citados, possui uma certa tendência a realizar a compra do pão francês também.

Já o óleo de soja, aparece relacionado nas compras quando se fala de: alho, ovo ou pão francês, indicando que na compra dos produtos, uma possível fritura destes pode ser feita utilizando o óleo de soja.

O ovo aparece relacionado com: óleo de soja, alho, pão francês e vinagre, nos indicando que na compra destes itens a tendência da compra do ovo também é feita, mas neste caso, em diferentes contextos de preparo do ovo.

O vinagre se mostra tendência também na compra de: pão francês, ovo ou óleo de soja.

O último item analisado é o alho que se mostrou tendência de compra em: óleo de soja, ovo ou pão francês.

Dessa forma, é possível compreender que alguns itens são praticamente unânimes na compra de um ou outro, como é o caso do ovo e o pão francês, e, que poderiam possivelmente entrar em de promoção juntos ou quaisquer outras medidas que a direção do mercado achar cabível para o desenvolvimento do seu negócio