Por: Vitor F de A Duarte

Atividade prática desenvolvida para a disciplina de Data Science Algorithms - Turma_001

1 - Você terá que analisar as características dos clusters gerados e relacioná-los com as regras geradas pelo apriori, descreva isso em um relatório e com as regras e clusters gerados.

Pelo fato dos atributos serem numéricos, eu tive dificuldade em usar o algoritmo apriori para a base de dados A4. Desta forma, **conforme concedido liberdade para tal nas introduções da atividade**, usei uma base de dados externa para gerar clusters e então relacioná-los com as regras geradas pelo apriori.

Base de dados utilizada: mercado2.arff - esta base de dados foi obtida por meio de um curso de WEKA via Udemy que pode ser acessado em Mineração de Regras de Associação com Weka, Apriori e Java | Udemy

Primeiramente, **clustering** agrupa instâncias semelhantes (descobre “perfis”) ao passo que **apriori** descobre regras de co-ocorrência de atributos (descobre “padrões”).

Apriori com a base de dados mercado2 (nessa base de dados os “não” foram retirados para que as regras de associações fossem mais pertinentes. Na geração de clusters a base de dados com os “não” foi utilizada.

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Relation:     mercado
Instances:    10
Attributes:   7
              leite
              cafe
              cerveja
              pao
              manteiga
              arroz
              feijao
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.25 (2 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 15

Generated sets of large itemsets:

Size of set of large itemsets L(1): 7

Size of set of large itemsets L(2): 5

Size of set of large itemsets L(3): 2
```

```
Best rules found:

 1. cafe=sim 3 ==> pao=sim 3    <conf:(1)> lift:(2) lev:(0.15) [1] conv:(1.5)
 2. cafe=sim 3 ==> manteiga=sim 3    <conf:(1)> lift:(2) lev:(0.15) [1] conv:(1.5)
 3. cafe=sim manteiga=sim 3 ==> pao=sim 3    <conf:(1)> lift:(2) lev:(0.15) [1] conv:(1.5)
 4. cafe=sim pao=sim 3 ==> manteiga=sim 3    <conf:(1)> lift:(2) lev:(0.15) [1] conv:(1.5)
 5. cafe=sim 3 ==> pao=sim manteiga=sim 3    <conf:(1)> lift:(2.5) lev:(0.18) [1] conv:(1.8)
 6. leite=sim 2 ==> pao=sim 2    <conf:(1)> lift:(2) lev:(0.1) [1] conv:(1)
 7. leite=sim 2 ==> manteiga=sim 2    <conf:(1)> lift:(2) lev:(0.1) [1] conv:(1)
 8. leite=sim manteiga=sim 2 ==> pao=sim 2    <conf:(1)> lift:(2) lev:(0.1) [1] conv:(1)
 9. leite=sim pao=sim 2 ==> manteiga=sim 2    <conf:(1)> lift:(2) lev:(0.1) [1] conv:(1)
10. leite=sim 2 ==> pao=sim manteiga=sim 2    <conf:(1)> lift:(2.5) lev:(0.12) [1] conv:(1.2)
```

Para estas regras encontradas, lemos: SE café ENTÃO pão com confiança de 1 (100%), ou seja, toda vez que uma pessoa compra café, ela compra pão também.

Clusters gerados - 2 5 e 6

Para gerar os clusters, eu usei a base de dados contendo os “não” (disponível em: mercado.arff) para que um resultado fosse gerado com uma certa variabilidade.

2 Clusters

```
Number of iterations: 2
Within cluster sum of squared errors: 11.0

Initial starting points (random):

Cluster 0: sim,sim,nao,sim,sim,nao,nao
Cluster 1: nao,nao,nao,sim,nao,nao,nao

Missing values globally replaced with mean/mode

Final cluster centroids:
                        Cluster#
Attribute    Full Data          0          1
                (10.0)      (4.0)      (6.0)
=============================================
leite               nao        sim        nao
cafe                nao        sim        nao
cerveja             nao        nao        nao
pao                 sim        sim        nao
manteiga            sim        sim        nao
arroz               nao        nao        nao
feijao              nao        nao        nao
```

```
Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       4 ( 40%)
1       6 ( 60%)
```

Primeiramente, podemos observar que o cluster 0 teve 4 instâncias (40%) e o cluster 1 teve 6 instâncias (60%)

Como **interpretação geral**, temos:

Cluster 0 (40% dos clientes): Compram leite, café, pão e manteiga.
Perfil: consumidores de café da manhã clássico.
Cluster 1 (60% dos clientes): Tendem a não comprar leite, café, pão nem manteiga.
Perfil: clientes que compram outras coisas.

Regra de associação:

{leite=sim, cafe=sim} → cluster=0
{leite=nao, cafe=nao, manteiga=nao} → cluster=1

5 Clusters

```
kMeans
======

Number of iterations: 3
Within cluster sum of squared errors: 6.0

Initial starting points (random):

Cluster 0: sim,sim,nao,sim,sim,nao,nao
Cluster 1: nao,nao,nao,sim,nao,nao,nao
Cluster 2: nao,sim,nao,sim,sim,nao,nao
Cluster 3: nao,nao,nao,nao,sim,nao,nao
Cluster 4: nao,nao,sim,nao,nao,nao,nao

Missing values globally replaced with mean/mode

Final cluster centroids:
                        Cluster#
Attribute    Full Data          0          1          2          3          4
                (10.0)      (2.0)      (3.0)      (3.0)      (1.0)      (1.0)
================================================================================
leite              nao        sim        nao        nao        nao        nao
cafe               nao        sim        nao        sim        nao        nao
cerveja            nao        sim        nao        nao        nao        sim
pao                sim        sim        nao        sim        nao        nao
manteiga           sim        sim        nao        sim        sim        nao
arroz              nao        nao        sim        nao        nao        nao
feijao             nao        nao        sim        nao        nao        nao
```

```
Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       2 ( 20%)
1       3 ( 30%)
2       3 ( 30%)
3       1 ( 10%)
4       1 ( 10%)
```

Primeiramente, podemos observar:

Distribuição: Cluster 0 → 2 instâncias (20%) Cluster 1 → 3 instâncias (30%) Cluster 2 → 3 instâncias (30%) Cluster 3 → 1 instância (10%) Cluster 4 → 1 instância (10%)

Cluster 0 (20%) leite = sim, café = sim, cerveja = sim, pão = sim, manteiga = sim
Perfil: clientes de “café da manhã completo”, mas que também compram cerveja. Grupo pequeno, mas com alto consumo variado.

Cluster 1 (30%) pão = sim, mas leite/café/manteiga = não
Perfil: compradores básicos de pão, sem outros itens de destaque.

Cluster 2 (30%) arroz = sim, feijão = sim, café = sim, pão = sim, manteiga = não
Perfil: compradores de refeição principal (arroz+feijão) e café/pão. Mais voltado para alimentação do dia a dia, não só café da manhã.

Cluster 3 (10%) manteiga = sim, mas sem outros produtos
Perfil: comprador específico de manteiga (cluster pequeno, isolado).

Cluster 4 (10%) cerveja = sim, sem outros itens relevantes
Perfil: comprador exclusivo de cerveja.

Como **interpretação geral**, temos:

10 clientes em 5 perfis distintos.

Clusters maiores (1 e 2, 60% dos clientes):
Cluster 1 → focados em pão simples.
Cluster 2 → perfil de refeição completa (arroz, feijão, café e pão).

Clusters menores (0, 3 e 4, 40% dos clientes):

Cluster 0 → consumidores diversificados (café da manhã + cerveja).
Cluster 3 → nicho de manteiga. Cluster 4 → nicho de cerveja.

Regras de associação:

{leite=sim, cafe=sim, manteiga=sim, cerveja=sim} → cluster=0
{arroz=sim, feijao=sim} → cluster=2
{pao=sim, leite=nao, cafe=nao} → cluster=1.
{manteiga=sim, outros=nao} → cluster=3
{cerveja=sim} → cluster=4

6 Clusters

```
kMeans
======

Number of iterations: 3
Within cluster sum of squared errors: 4.0

Initial starting points (random):

Cluster 0: sim,sim,nao,sim,sim,nao,nao
Cluster 1: nao,nao,nao,sim,nao,nao,nao
Cluster 2: nao,sim,nao,sim,sim,nao,nao
Cluster 3: nao,nao,nao,nao,sim,nao,nao
Cluster 4: nao,nao,sim,nao,nao,nao,nao
Cluster 5: sim,nao,sim,sim,sim,nao,nao

Missing values globally replaced with mean/mode

Final cluster centroids:
                        Cluster#
Attribute    Full Data          0          1          2          3          4          5
                (10.0)      (1.0)      (3.0)      (3.0)      (1.0)      (1.0)      (1.0)
=============================================================================================
leite               nao        sim        nao        nao        nao        nao        sim
cafe                nao        sim        nao        sim        nao        nao        nao
cerveja             nao        nao        nao        nao        nao        sim        sim
pao                 sim        sim        nao        sim        nao        nao        sim
manteiga            sim        sim        nao        sim        sim        nao        sim
arroz               nao        nao        sim        nao        nao        nao        nao
feijao              nao        nao        sim        nao        nao        nao        nao
```

```
Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       1 ( 10%)
1       3 ( 30%)
2       3 ( 30%)
3       1 ( 10%)
4       1 ( 10%)
5       1 ( 10%)
```

Primeiramente, podemos observar:

Cluster 0 → 1 instância (10%) Cluster 1 → 3 instâncias (30%) Cluster 2 → 3 instâncias (30%) Cluster 3 → 1 instância (10%) Cluster 4 → 1 instância (10%) Cluster 5 → 1 instância (10%)

Cluster 0 (10%) leite = sim, café = sim, pão = sim, manteiga = sim Perfil: consumidor típico de café da manhã completo.

Cluster 1 (30%) pão = sim, os outros principais itens = não Perfil: comprador básico de pão. Esse é o grupo mais frequente junto com o cluster 2.

Cluster 2 (30%) arroz = sim, feijão = sim, café = sim, pão = sim Perfil: consumidor de refeição completa (arroz + feijão), com café e pão.

Cluster 3 (10%) manteiga = sim, café = sim, pão = sim, mas sem arroz/feijão/leite/cerveja Perfil: comprador focado em café da manhã simples (pão, manteiga e café).

Cluster 4 (10%) cerveja = sim, manteiga = sim Perfil: comprador específico e curioso: combina cerveja + manteiga (perfil isolado).

Cluster 5 (10%) leite = sim, cerveja = sim, pão = sim, manteiga = sim Perfil: consumidor diversificado (café da manhã + cerveja).

Como **interpretação geral**, temos:

10 consumidores em 6 grupos bem distintos, mas com clusters pequenos (4 deles têm apenas 1 instância cada).

Clusters dominantes (1 e 2, 60%): Cluster 1 → consumidores que compram apenas pão.

Cluster 2 → consumidores de refeição completa (arroz e feijão + café e pão).

Clusters de nicho (0, 3, 4, 5, totalizando 40%): Café da manhã robusto (cluster 0), café da manhã simples (cluster 3), mistura inusitada de cerveja+manteiga (cluster 4), e perfil diversificado com leite e cerveja juntos (cluster 5).

Regras de associação:

{leite=sim, cafe=sim, pao=sim, manteiga=sim} → cluster=0
{pao=sim, leite=nao, cafe=nao} → cluster=1
{arroz=sim, feijao=sim} → cluster=2
{arroz=sim, feijao=sim, cafe=sim} → cluster=2
{manteiga=sim, pao=sim, cafe=sim} → cluster=3
cerveja=sim} → cluster=4
{leite=sim, cerveja=sim, pao=sim} → cluster=5

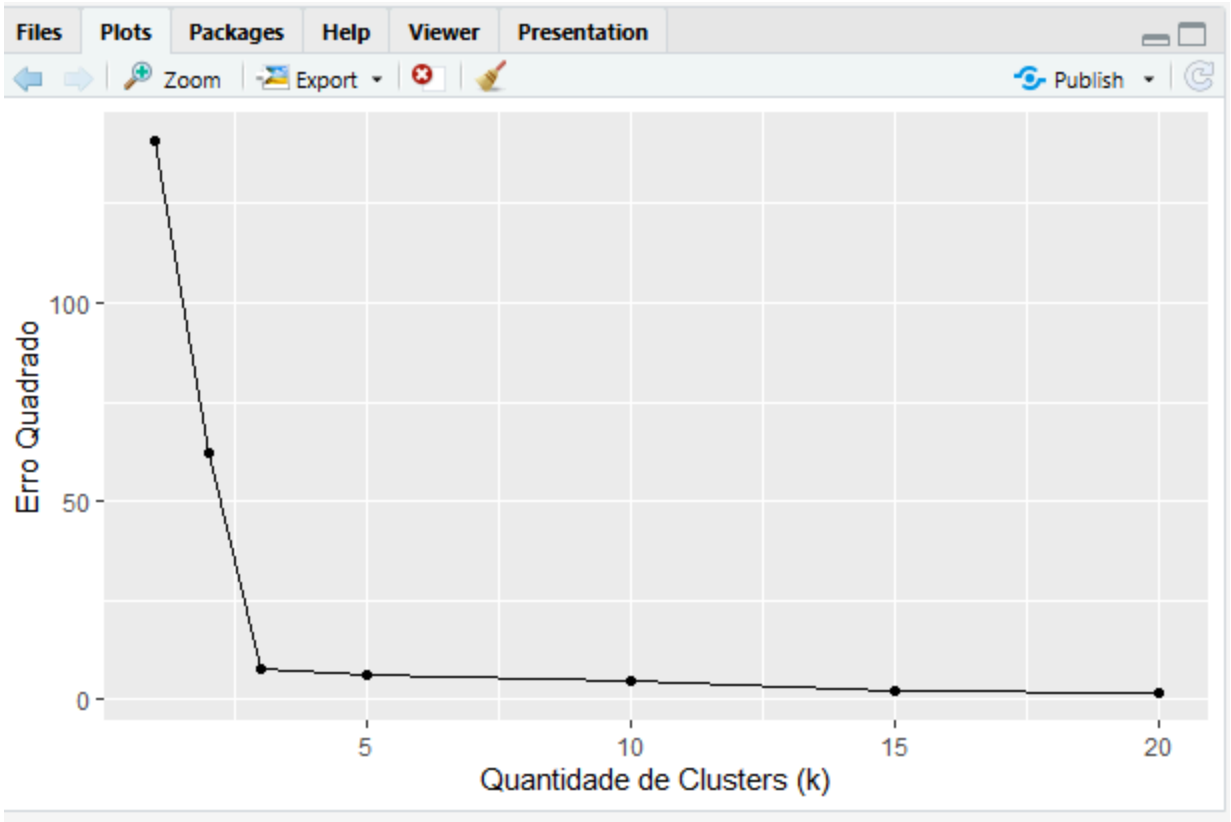
2 - No segundo experimento você deverá usar a base de dados "IrisDataSet" no arquivo "iris.csv" bastante conhecida para experimentos e clustering. Você deverá executar o experimento com o Kmeans no Weka e verificar qual é o melhor número de clusters para o modelo gerado, utilizando o erro RMS com um gráfico, como foi feito na unidade 6 com a base de dados "A".

Para tal verificação, o algoritmo Kmeans foi executado 7 vezes (1,2,3, 5,10,15,20)

Erro RMS com um gráfico para cada um desses clusters:

1 - 141.16611042137328
2 - 62.127790750538175
3 - 7.801559361268048
5 - 6.277659330769319
10 - 4.587500225526149
15 - 2.1432209241343325
20 - 1.587947152482922

Gráfico a partir destes erros RMS (Gráfico feito no R)



Interpretação do gráfico:

A quantidade ideal de clusters para um dado modelo ocorre quando há um joelho ou cotovelo no gráfico, ou seja, nesse caso 3 clusters (k= 3). O erro quadrado começa bem alto e à medida que a quantidade de clusters aumenta, ele diminui, posteriormente, ele tende a zero.

Cada ponto fica mais perto do seu centroide quando há mais centroides disponíveis. E em casos extremos se tivermos k = número de pontos, cada ponto é seu próprio cluster, e o erro quadrado será ZERO (cada ponto é exatamente no centroide).

Quanto mais clusters melhor será o ajuste matemático, mas pior interpretação prática. O desafio é encontrar o equilíbrio ideal (que seria aproximadamente onde está o joelho/cotovelo do gráfico).