LUIZ RICARDO MORIGI

Aplicando experimentos de data mining utilizando o software WEKA.

São Sebastião do Paraíso
2023
Universidade de Franca - UNIFRAN

Resumo

Primeiramente gostaria de agradecer toda a equipe da cruzeiro do sul educacional, e do polo UNIFRAN pelo suporte. E em especial, os professores: prof. Jose Almir Ferreira, Prof. Dr. Alberto Messias e prof. Alexandre Miccheleti Lucena pelo conteúdo ministrado e pelo suporte! (peço desculpas se esqueci de mencionar alguém).

Nesta atividade prática irei reproduzir os experimentos de data mining propostos na disciplina de Algoritimos para Ciência de dados. Para isso, irei utilizar os conceitos de regras de associação e algoritimos de clustering para analisar os dois conjuntos de dados disponibilizados pela instituição.

Para essa atividade, vou analisar dois conjuntos de dados utilizando o software WEKA, e aplicando o algoritimo Kmeans.

- No primeiro, vamos analisar o cliente 'A'. Será analisado as características dos clusters gerados e, posteriormente, relacionados com as regras geradas pelo **algoritmo Apriori Classico**.

- No Segundo, "IrisDataSet", vamos verificar qual é o melhor número de clusters para o modelo gerado, utilizando o erro RMS com um gráfico.

Capitulo I

Análise do cliente “A”

Vamos usar o software WEKA para que possamos, através do algoritmo Kmeans, explorar os dados a fim de encontrar informações que ajude a melhorar o desempenho e rentabilidade da empresa.

Para obter um resultado correto, vamos ultilizar o gráfico de erro RMS para descobrir qual o número ideal de clusters a ser ultilizado no Kmeans. Para isso, é necessário rodar o algoritmo variando o número de clusters de 1 á 15, e observar o “Within cluster sum of squared errors:“, ou “erro RMS”.

Abaixo podemos vizualizar a tabela com os valores:

Clusters	Erro RMS
1	2590,55
2	1861,14
3	1293,98
4	1010,78
5	909,78
6	760,35
7	712,12
8	594,64
9	674,37
10	628,55
11	515,20
12	437,80
13	427,43
14	422,22
15	410,06

A validação do modelo proposto é feito através de uma análise do gráfico de erro RMS, com o objetivo de encontrar o “Joelho da curva”:

“Com k tendendo ao número de instâncias, o erro quadrado tende a 0”

Observando c0 “(k=1)”, notamos que o modelo possui um erro muito grande, e para o c14 “(k=15)” o modelo fica extremamente especializado. Sendo assim, o número ideal de clusters para o modelo é o meio termo entre eles.

Analisando o gráfico, observamos que o cluster c6 “(K=7)” está localizado onde chamamos de “Joelho da curva“, sendo assim o número ideal de clusters a ser usado para o experimento.

Sabendo qual é o numero ideal de clusters, podemos rodar o algoritmo kmeans no software WEKA:

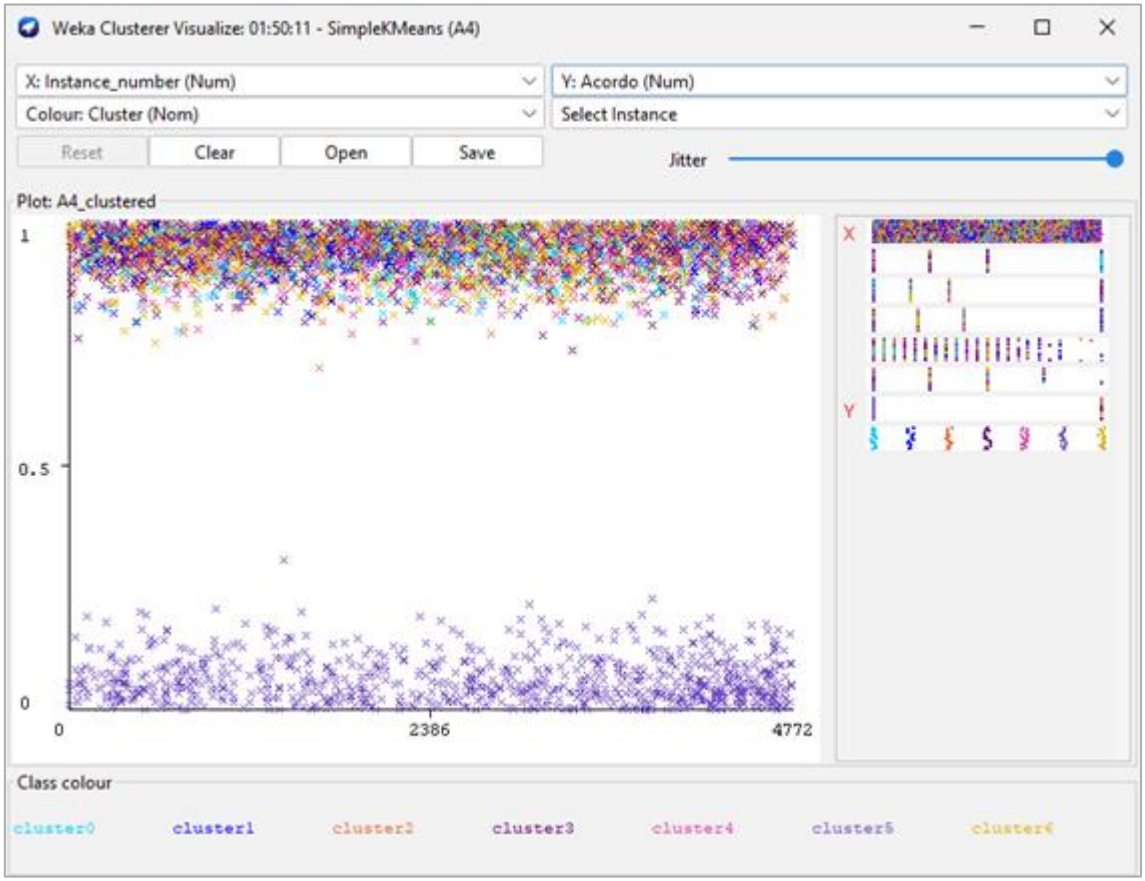
```
=== Clustering model (full training set) ===


kMeans
======

Number of iterations: 14
Within cluster sum of squared errors: 712.1244016805499

Initial starting points (random):

Cluster 0: 4,1,1,4,1,1
Cluster 1: 4,2,5,10,2,1
Cluster 2: 4,2,2,2,2,1
Cluster 3: 1,2,1,2,1,1
Cluster 4: 1,2,2,4,1,1
Cluster 5: 1,2,1,8,1,1
Cluster 6: 2,2,1,10,2,1

Missing values globally replaced with mean/mode
```

Podemos observar que o erro RMS é um meio termo entre c0 e c14.

```
Final cluster centroids:
                        Cluster#
Attribute    Full Data          0          1          2          3          4          5          6
              (4773.0)    (534.0)    (384.0)    (236.0)    (869.0)    (759.0)    (867.0)   (1124.0)
=============================================================================================================
Idade           1.7461          4     2.1432          4     1.0311     0.2688     1.9146     1.4867
Atraso          2.4588     1.2772     0.7969          6          6     0.8814      2.045     1.4911
Valor           1.4718     1.2247          5     0.4576     0.3751     1.2648     2.0092     1.1699
CONTATO         4.7591      4.161     5.8203     3.9958     4.7135     4.8577     5.2549      4.427
EFETIVO         1.1672     1.1685     1.2422     1.2246     1.1749     1.1291      1.113     1.1904
Acordo          0.8184          1          1          1          1          1          0          1



Time taken to build model (full training data) : 0.03 seconds

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

Clustered Instances

0       534 ( 11%)
1       384 (  8%)
2       236 (  5%)
3       869 ( 18%)
4       759 ( 16%)
5       867 ( 18%)
6      1124 ( 24%)
```

Com base nessa análise, podemos observar que o número ideal de clusters é 7 (c6). Rodando o algoritmo Kmeans com K=7 obtemos o seguinte resultado:

Número de Instâncias por Acordo



Observe que o algoritmo agrupou todas as instancias que não fecham acordo e posicionou na parte inferior, enquanto as instancias que fecham acordo ficaram posicionadas na parte superior do gráfico.

Podemos agora, explorar todos os clusters gerados, e comparar os resultados com as regras geradas.

A base de dados foi resumida em faixas para que o algoritmo pudesse processá-la. A codificação ficou da seguinte maneira:

Faixa Etária:

- Código 0: De 0 a 25 Anos;
- Código 1: De 26 a 35 Anos;
- Código 2: De 36 a 45 Anos;
- Código 4: Maior que 45 Anos;

Faixa de Valores:

- Código 0: De 0 a 200 Reais;
- Código 1: De 200 a 500 Reais;
- Código 2: De 500 a 1000 Reais;
- Código 5: Maior que 1000 Reais;

Faixa de Atraso:

- Código 0: De 0 a 15 dias;
- Código 1: De 15 a 30 dias;
- Código 2: De 31 a 120 dias;
- Código 6: acima de 120 dias

Tendo as regras definidas, podemos partir para classificação e comparação dos clusters

Classificação e comparação dos clusters:

Cluster 0:

Atributos:

- Número de Instâncias = 534
- Idade: 4 = Maior que 45;
- Atraso: 1.2772 = De 15 a 30 dias;
- Valor: 1.2247 = De 200 a 500 reais;
- Contato: 4.161 =
- Efetivo: 1.1685 =
- Acordo:

Cluster#	0
	(534.0)
Idade	4
Atraso	1.2772
Valor	1.2247
CONTATO	4.161
EFETIVO	1.1685
Acordo	1

Cluster 1:

Atributos:

- Número de Instâncias = 384
- Idade: 2.1432 = De 36 a 45 anos;
- Atraso: 0.7969 = De 15 a 30 dias;
- Valor: 5 = Maior que 1000 reais.
- Contato: 5.8203 =
- Efetivo: 1.2422 =
- Acordo: 1 =

	1
	(384.0)
Idade	2.1432
Atraso	0.7969
Valor	5
CONTATO	5.8203
EFETIVO	1.2422
Acordo	1

Cluster 2:

Atributos:

- Número de Instâncias = 236
- Idade: 4 = Maior que 45 anos.
- Atraso: 6 = Acima de 120 dias.
- Valor: 0.4576 = De 0 a 200 reais;
- Contato: 3.9958 =
- Efetivo: 1.2246 =
- Acordo: 1 =

	2
	(236.0)
Idade	4
Atraso	6
Valor	0.4576
CONTATO	3.9958
EFETIVO	1.2246
Acordo	1

Cluster 3:

Atributos:

- Número de Instâncias = 869
- Idade: 1.0311 = De 26 a 35 anos;
- Atraso: 6 = acima de 120 dias.
- Valor: 0.3751 = De 0 a 200 reais;
- Contato: 4.7135 =
- Efetivo: 1.1749 =
- Acordo: 1 =

	3
	(869.0)
Idade	1.0311
Atraso	6
Valor	0.3751
CONTATO	4.7135
EFETIVO	1.1749
Acordo	1

Cluster 4:

Atributos:

- Número de Instâncias = 759
- Idade: 0.2688 = De 0 a 25 anos;
- Atraso: 0.8814 = De 15 a 30 dias;
- Valor: 1.2648 = De 200 a 500 reais;
- Contato: 4.8577 =
- Efetivo: 1.1291 =
- Acordo: 1 =

	4
	(759.0)
Idade	0.2688
Atraso	0.8814
Valor	1.2648
CONTATO	4.8577
EFETIVO	1.1291
Acordo	1

Cluster 5:

Atributos:

- Idade: 1.9146 = De 26 a 35 anos;
- Atraso: 2.045 = 31 a 120 dias;
- Valor: 2.0092 = 500 a 1000 reais;
- Contato: 5.2549 =
- Efetivo: 1.1130 =
- Acordo: 0 =

	5
	(867.0)
Idade	1.9146
Atraso	2.045
Valor	2.0092
CONTATO	5.2549
EFETIVO	1.113
Acordo	0

Cluster 6:

Atributos:

- Idade: 1.4867 = De 26 a 35 anos;
- Atraso: 1.4911 = De 15 a 30 dias;
- Valor: 1.1699 = De 200 a 500 reais;
- Contato: 4.427. =
- Efetivo: 1.1904 =
- Acordo: 1 =

	6
	(1124.0)
Idade	1.4867
Atraso	1.4911
Valor	1.1699
CONTATO	4.427
EFETIVO	1.1904
Acordo	1

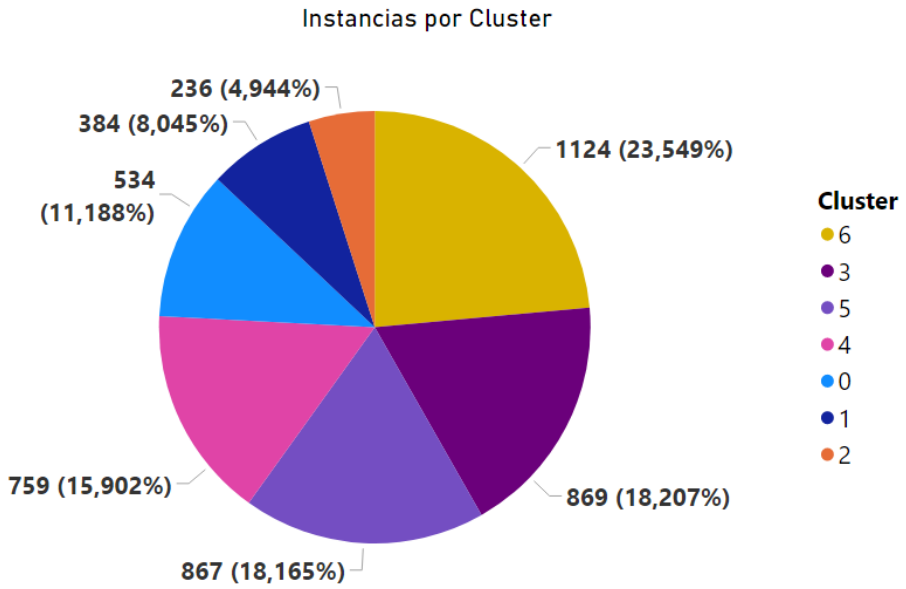
Podemos agora comparar os dados para ver se encontramos padrões:

Se classificarmos por instancias temos:

1. Cluster 6 com 1124 Instancias (23,549%)
2. Cluster 3 com 869 Instancias (18,207%)
3. Cluster 5 com 867 Instancias (18,165%)
4. Cluster 4 com 759 Instancias (15,902%)
5. Cluster 0 com 534 Instancias (11,188%)
6. Cluster 1 com 384 Instancias (8,045%)
7. Cluster 2 com 236 Instancias (4,944%)

Instancias	Cluster
1124	6
869	3
867	5
759	4
534	0
384	1
236	2
4773	

Podemos vizualizar o número de instancias por clusters em um gráfico de pizza, para termos noção da relevancia de cada cluster no todo.



Clusters Filtrados:

Por Efetividade:

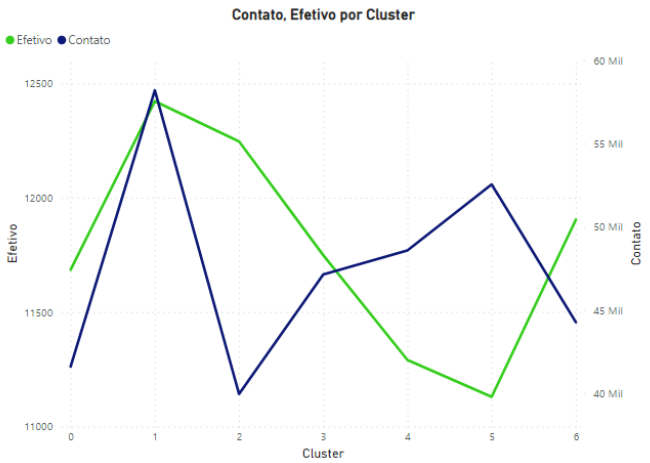
Cluster	Idade	Atraso	Valor	Contato	Efetivo	Acordo
1	36 a 45	0 a 15	1000+	6	1,24	1.00
2	45+	120+	0 a 200	3,99	1,22	1.00
6	26 a 35	15 a 30	200 a 500	4,43	1,19	1.00
3	26 a 35	120+	0 a 200	4,71	1,17	1.00
0	45+	15 a 30	200 a 500	4,16	1,17	1.00
4	0 a 25	0 a 15	200 a 500	4,86	1,13	1.00
5	26 a 35	31 a 120	500 a 1000	5,25	1,11	0

Podemos observar que o cluster 1 é o cluster com melhor efetividade em seus contatos, enquanto o cluster 5 é o pior.

Por Contato:

Cluster	Idade	Atraso	Valor	Contato	Efetivo	Acordo
1	36 a 45	0 a 15	1000+	6	1,24	1.00
5	26 a 35	31 a 120	500 a 1000	5,25	1,11	0
4	0 a 25	0 a 15	200 a 500	4,86	1,13	1.00
3	26 a 35	120+	0 a 200	4,71	1,17	1.00
6	26 a 35	15 a 30	200 a 500	4,43	1,19	1.00
0	45+	15 a 30	200 a 500	4,16	1,17	1.00
2	45+	120+	0 a 200	3,99	1,22	1.00

Podemos observar que o cluster 1 também conta com o maior número de contatos realizados.



Explorando o cluster n°5

É interessante analisar esse cluster, pois dentre todos os clusters, ele é o unico agrupamento que não gera acordo.

Contando com um total 867 instancias, podemos observar que 18% das instancias não fecham acordo.

Cluster 5:

Atributos:

- Idade: 1.9146 = De 26 a 35 anos;
- Atraso: 2.045 = 31 a 120 dias;
- Valor: 2.0092 = 500 a 1000 reais;
- Contato: 5.2549 =
- Efetivo: 1.1130 =
- Acordo: 0 =

	5 (867.0)
Idade	1.9146
Atraso	2.045
Valor	2.0092
CONTATO	5.2549
EFETIVO	1.113
Acordo	0

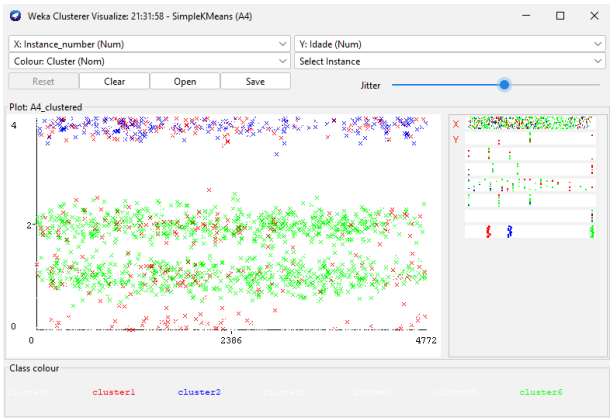
Clusters com maior efetividade:

Os clusters com o melhor número de efetividade são os clusters 1, 2 e 6, como podemos observar na tabela abaixo:

Cluster	Idade	Atraso	Valor	Contato	Efetivo	Acordo
1	36 a 45	0 a 15	1000+	6	1,24	1.00
2	45+	120+	0 a 200	3,99	1,22	1.00
6	26 a 35	15 a 30	200 a 500	4,43	1,19	1.00
3	26 a 35	120+	0 a 200	4,71	1,17	1.00
0	45+	15 a 30	200 a 500	4,16	1,17	1.00
4	0 a 25	0 a 15	200 a 500	4,86	1,13	1.00
5	26 a 35	31 a 120	500 a 1000	5,25	1,11	0

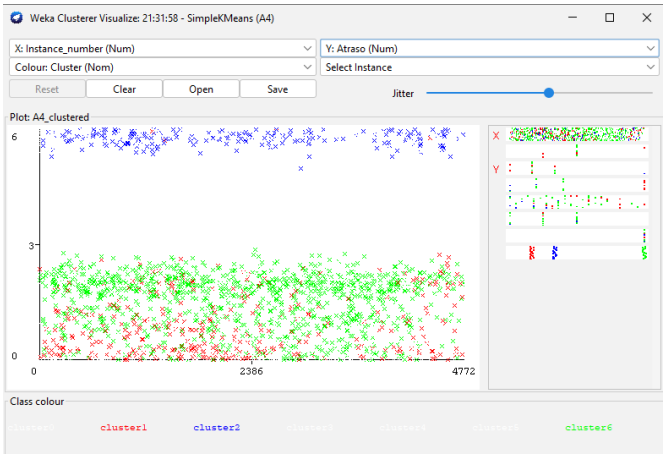
Os 3 primeiros clusters em número de efetividade representam (36,538%) no total de instancias.

Instancias por Idade:



Podemos observar que a maioria das instancias se posicionam nas faixas 1 e 2, e, apenas uma pequena faixa pertence a faixa 0. A faixa 4 aparece em segundo lugar com um número relevante de instancias.

Instancias por Atraso:



Podemos observar que grande parte das instancias ficam abaixo de 3, enquanto apenas o cluster 2 ocupa a faixa 6.

Clusters com menor efetividade:

Os clusters com o pior número em efetividade são os clusters 5, 4 e 3 respectivamente, como podemos observamos na tabela. Os clusters com menor efetividade somam (52,274%) do valor total de instancias.

Cluster	Idade	Atraso	Valor	Contato	Efetivo	Acordo
5	26 a 35	31 a 120	500 a 1000	5,25	1,11	0
4	0 a 25	0 a 15	200 a 500	4,86	1,13	1.00
3	26 a 35	120+	0 a 200	4,71	1,17	1.00
0	45+	15 a 30	200 a 500	4,16	1,17	1.00
6	26 a 35	15 a 30	200 a 500	4,43	1,19	1.00
2	45+	120+	0 a 200	3,99	1,22	1.00
1	36 a 45	0 a 15	1000+	6	1,24	1.00

A mediana sendo 0, pertence a (11,188%) do total de instancias.

Capitulo II

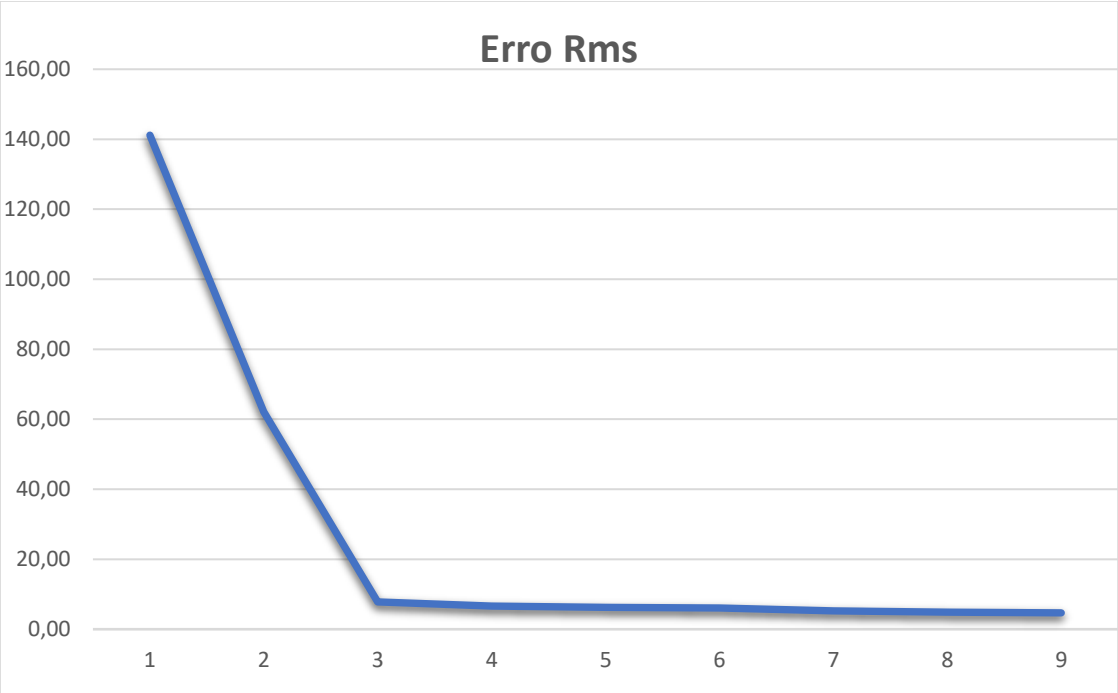
Análise de “IrisDataSet”

Faremos agora a análise desse dataset, que é bem famoso para esse tipo de exercicio. Assim como na análise anterior, vamos começar identificando através do gráfico de erro RMS, qual o número ideal de clusters para esse experimento.

Após rodar o algortitmo Kmeans, variando o valor de k até 15, obtemos a seguinte tabela:

N° de Clusters	Erro RMS
1	141,16
2	62,13
3	7,80
4	6,60
5	6,28
6	6,11
7	5,22
8	4,86
9	4,68

E a partir da tabela acima, obtemos o seguinte gráfico:



Claramente, o número ideal de clusters é 3, sabendo disso, podemos rodar o algoritmo com esse número de clusters:

```
=== Clustering model (full training set) ===


kMeans
======

Number of iterations: 3
Within cluster sum of squared errors: 7.801559361268048

Initial starting points (random):

Cluster 0: 6.1,2.9,4.7,1.4,Versicolor
Cluster 1: 6.2,2.9,4.3,1.3,Versicolor
Cluster 2: 6.9,3.1,5.1,2.3,Virginica

Missing values globally replaced with mean/mode
```

Obtemos as seguintes informações:

```
Final cluster centroids:
                           Cluster#
Attribute       Full Data          0          1          2
                  (150.0)     (50.0)     (50.0)     (50.0)
============================================================
sepal.length       5.8433      5.936      5.006      6.588
sepal.width        3.0573       2.77      3.428      2.974
petal.length        3.758       4.26      1.462      5.552
petal.width        1.1993      1.326      0.246      2.026
variety            Setosa Versicolor     Setosa  Virginica



Time taken to build model (full training data) : 0.01 seconds

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

Clustered Instances

0       50 ( 33%)
1       50 ( 33%)
2       50 ( 33%)
```

Classificação dos clusters:

Cluster 0:

Atributos:

- Número de Instâncias = 50
- Comprimento Sépala = 5.936
- Largura Sépala = 2.77
- Comprimento Pétala = 4.26
- Largura Pétala = 1.326
- Variedade = Versicolor

Attribute	0 (50.0)
sepal.length	5.936
sepal.width	2.77
petal.length	4.26
petal.width	1.326
variety	Versicolor

Cluster 1:

Atributos:

- Número de Instâncias = 50
- Comprimento Sépala = 5.006
- Largura Sépala = 3.428
- Comprimento Pétala = 1.462
- Largura Pétala = 0.246
- Variedade = Setosa

Attribute	1 (50.0)
sepal.length	5.006
sepal.width	3.428
petal.length	1.462
petal.width	0.246
variety	Setosa

Cluster 2:

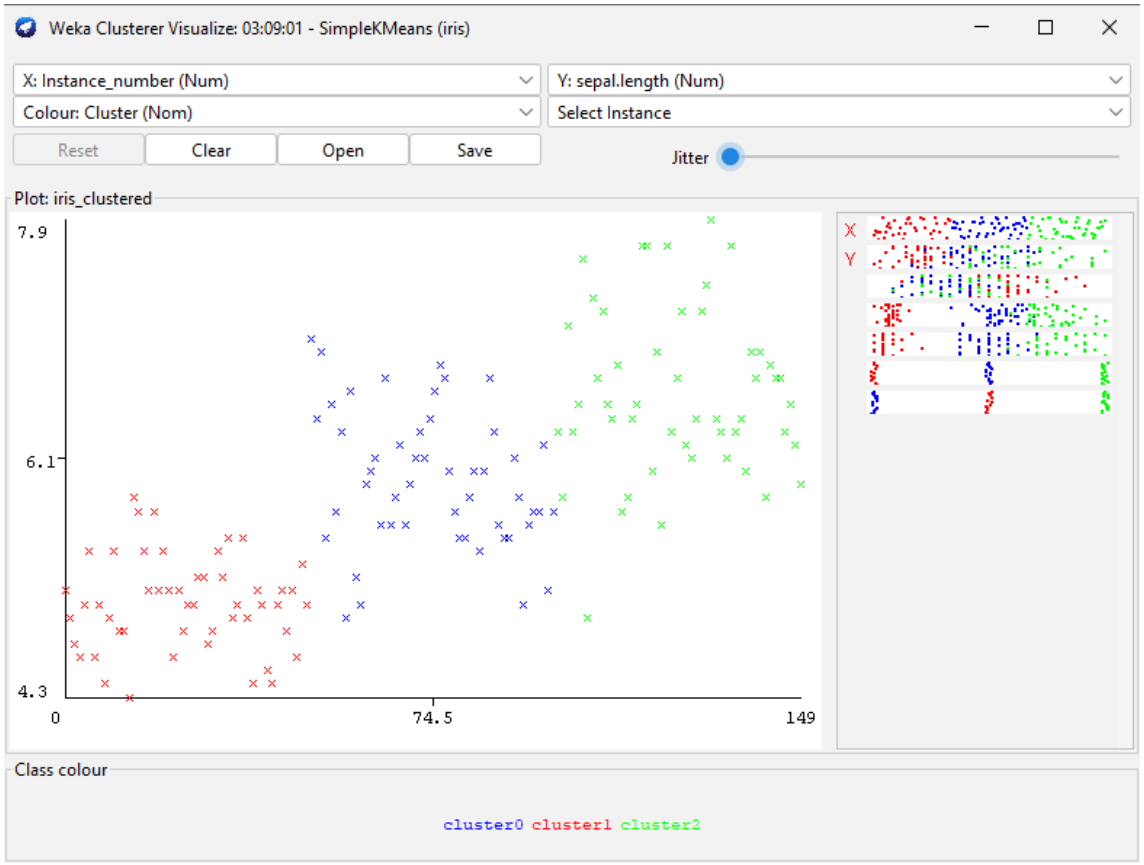
Atributos:

- Número de Instâncias = 50
- Comprimento Sépala = 6.588
- Largura Sépala = 2.974
- Comprimento Pétala = 5.552
- Largura Pétala = 2.026
- Variedade = Virginica

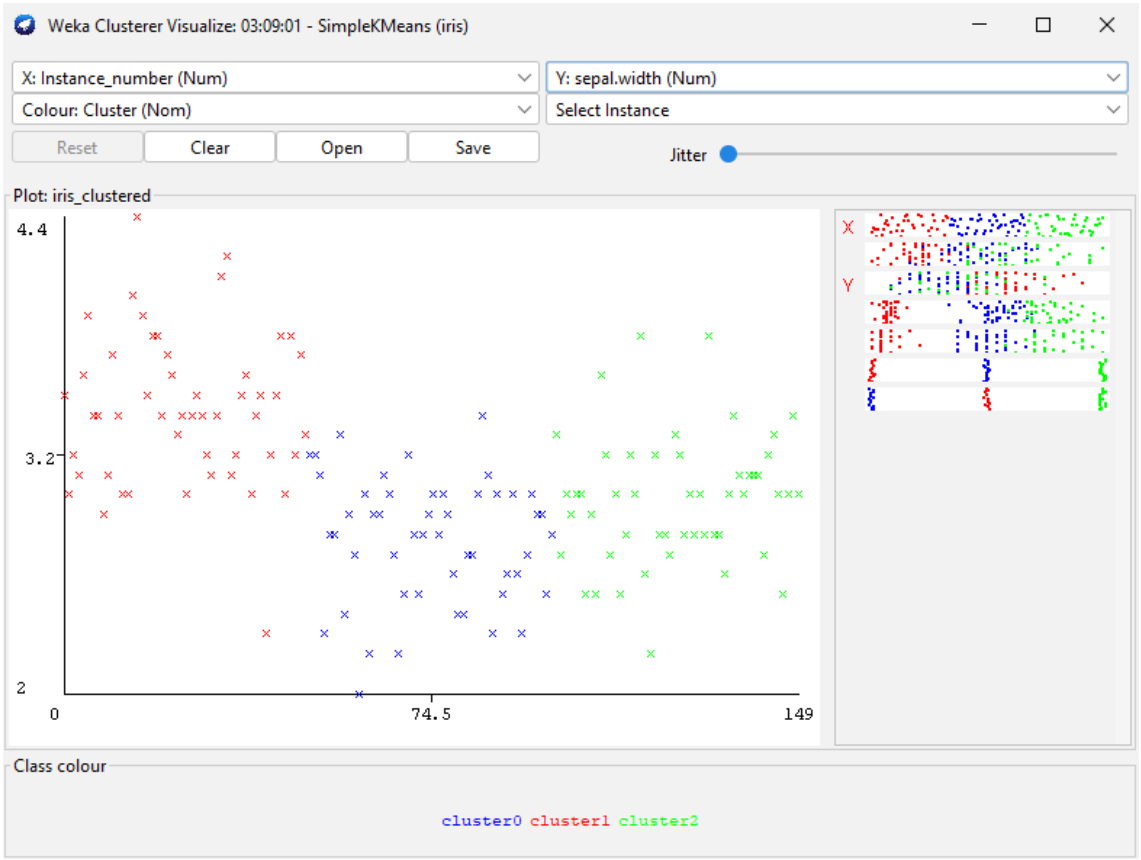
Attribute	2 (50.0)
sepal.length	6.588
sepal.width	2.974
petal.length	5.552
petal.width	2.026
variety	Virginica

Gráficos Gerados:

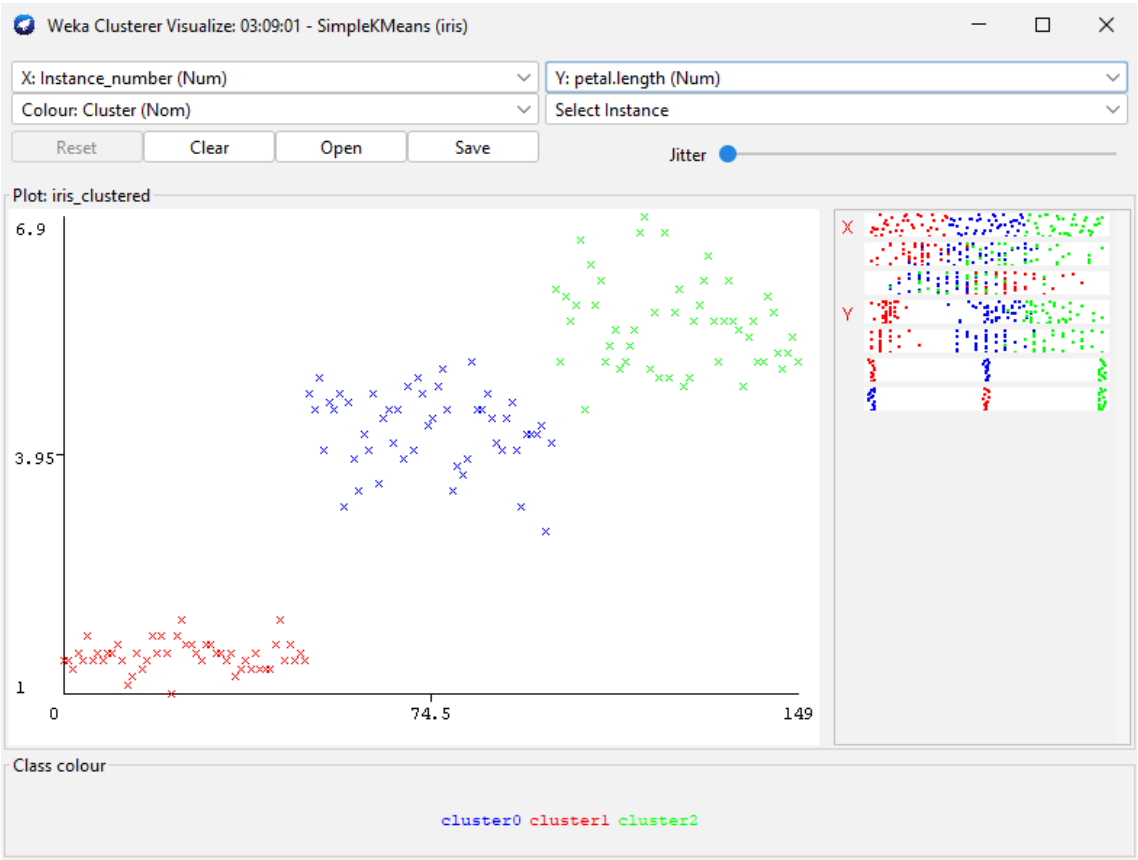
Número de Instâncias por Largura da Sépala



Número de Instâncias por Largura da Sépala



Número de Instâncias por Comprimento da Pétala



Número de Instâncias por Largura da Pétala