

Laboratório

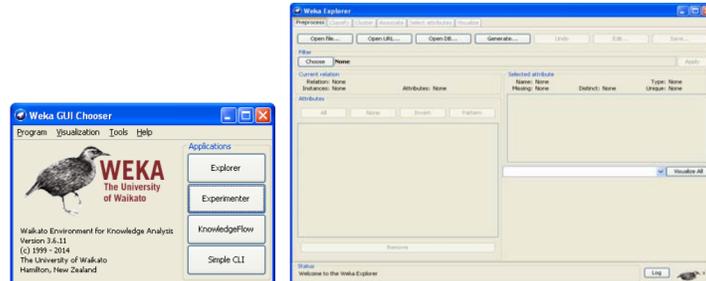
Clusterização com WEKA Explorer

Faça o download dos datasets **car-browsers.arff*** e **iris.arff***, e execute clusterização conforme as páginas abaixo, onde encontram-se **tutoriais**..

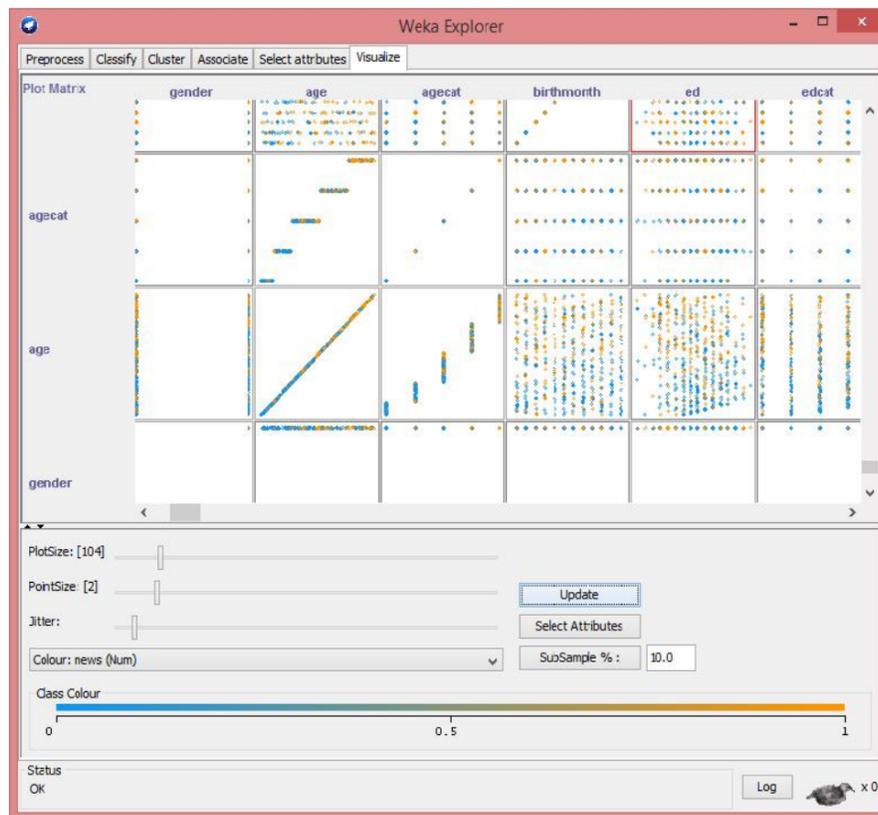
*disponível em: www.inf.ufpr.br/menotti/am-201/data.zip

1. Preparando os dados para classificação

1. Inicie uma sessão do **Weka** ou execute em linha de comando: `java -jar weka.jar`.
2. Quando a **GUI Chooser** surgir, selecione o **Explorer** a partir das quatro opções do lado direito.



3. Estamos no **Preprocess** agora. Clique no botão **Open** para abrir a caixa de diálogo padrão através da qual você pode selecionar um arquivo. Escolha o arquivo **telco_lab3.csv**.
4. Você pode ignorar atributos irrelevantes durante o processo de **Clustering**, como **custIds**. Para identificar atributos redundantes, poderíamos verificar a correlação a partir da Visualização do dataset na aba **Visualize**. Pode-se ver que os atributos **age** and **agecat** estão correlacionados. Um deles deve ser ignorado. Nós mantemos o atributo **age** para fins de agrupamento; também mantemos **ed** (removendo **edcat**), então temos 8 atributos restantes para **Clustering** (vamos ignorar **custIds**, **agecat** e **edcat** quando realizamos o **Clustering**).

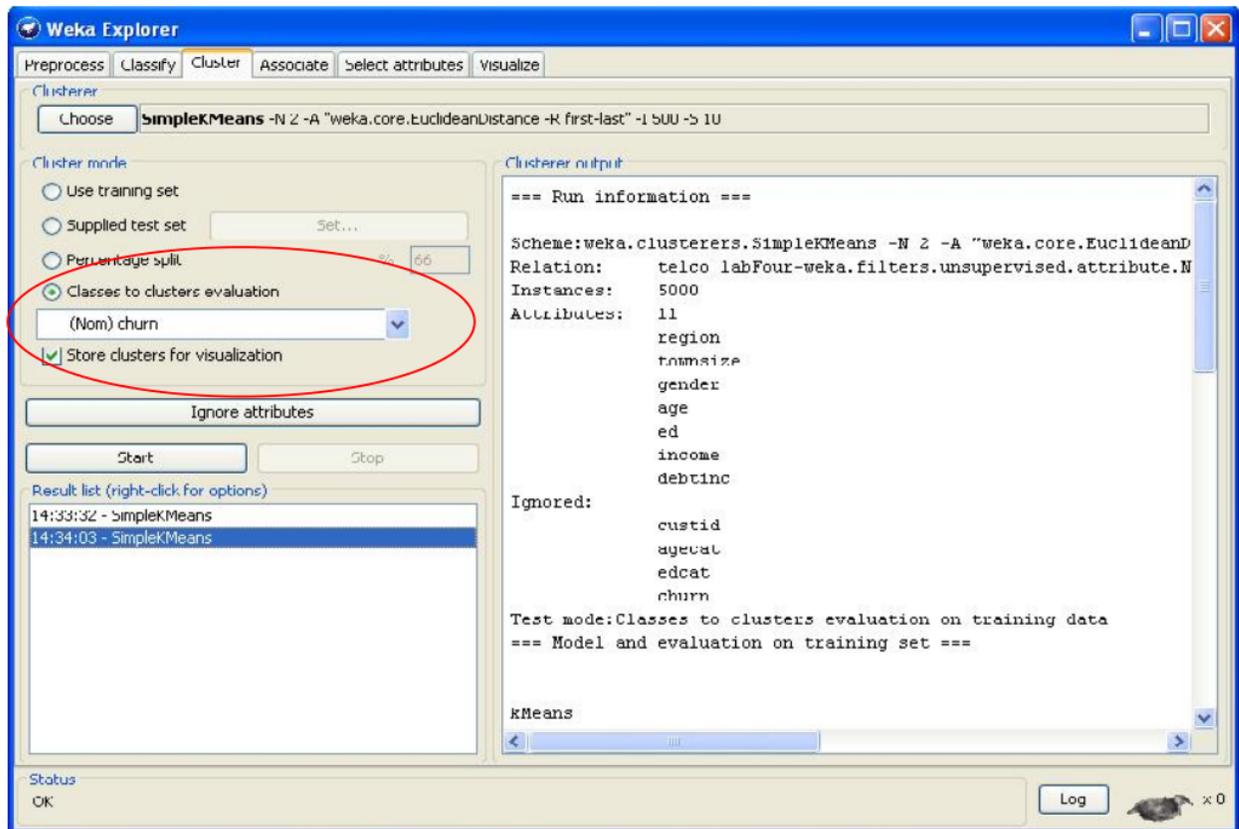


5. Antes de fazer o **Clustering** com Weka, precisamos **normalizar** seus valores de dados

numéricos (use o filtro **Normalize**). Como temos o rótulo de classe, gostaríamos de defini-lo como **nominal** antes da normalização. Esta informação será usada para avaliar o desempenho do clustering.

SimpleKMeans

1. Para executar o **Clustering** no dataset dados, clique na guia **Cluster** e escolha o algoritmo **SimpleKMeans**. Definimos $k = 2$ para este dataset. Clique em **Classes to clusters evaluation** e selecione o último atributo (**churn**) como **classe**. Clique em **Store Clusters for visualization**. Clique em **Ignore attributes** e seleciona **custIds**, **agecat**, **edcat** e o último atributo **churn** (classe, que não será clusterizado). Em seguida, clique em **Start**.



```

14:34:03 - SimpleKMeans

kMeans
=====

Number of iterations: 3
Within cluster sum of squared errors: 2000.5871625331147
Missing values globally replaced with mean/mode

Cluster centroids:

Attribute      Full Data      Cluster#
              (5000)        0          1
              (2482)       (2518)
-----
region         0.5004         0.5049     0.4958
townsize      0.4218         0.4184     0.4252
gender        0.5036         0          1
age           0.4758         0.4788     0.4729
ed            0.5025         0.5027     0.5024
income        0.043          0.0435     0.0425
debtinc       0.231          0.2302     0.2317

Time taken to build model (full training data) : 0.09 seconds

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

Clustered Instances

0      2482 ( 50%)
1      2518 ( 50%)

Class attribute: churn
Classes to Clusters:

    0    1 <-- assigned to cluster
1839 1895 | 0
 643  623 | 1

Cluster 0 <-- 1
Cluster 1 <-- 0

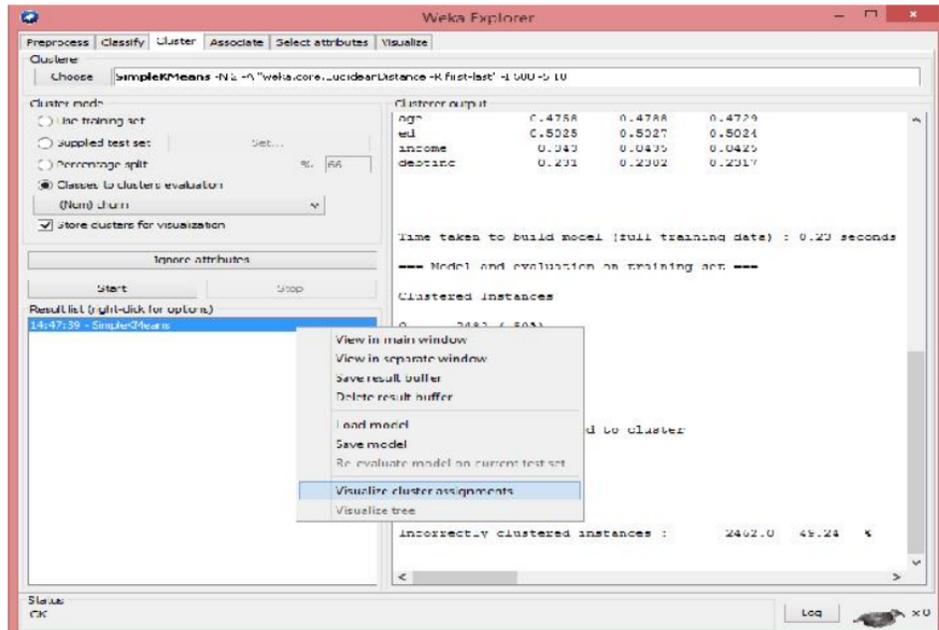
Incorrectly clustered instances :      2462.0    49.24    %

```

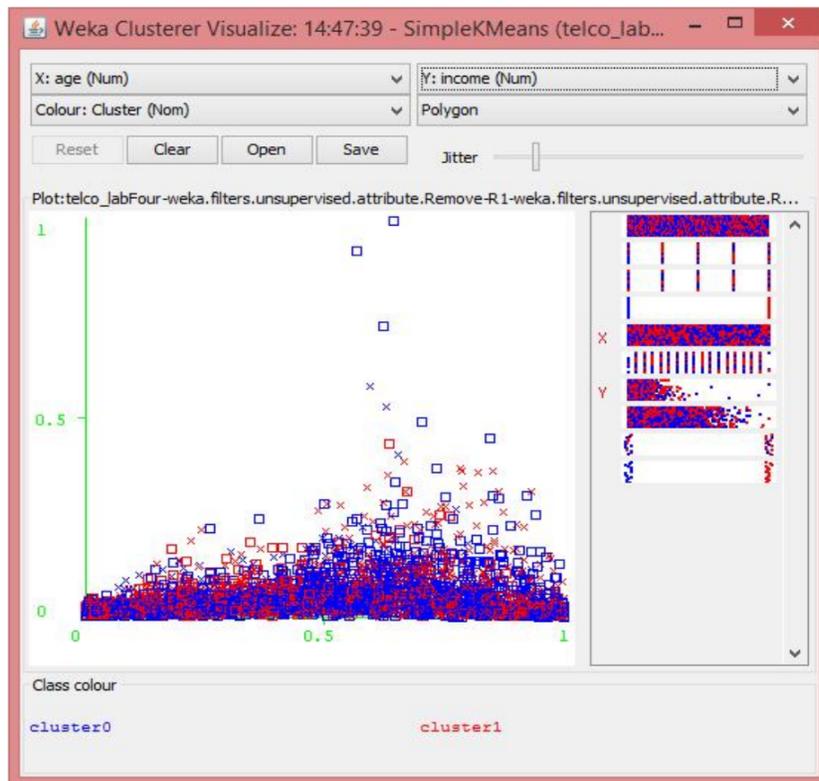
$$1839 + 623 = 2462$$

2. Você poderia visualizar os resultados da clusterização clicando com o botão **Direito** na

lista de resultados e escolher **Visualize clusters assignments**. Você pode selecionar uma combinação diferente de dois atributos como **X** e **Y**.

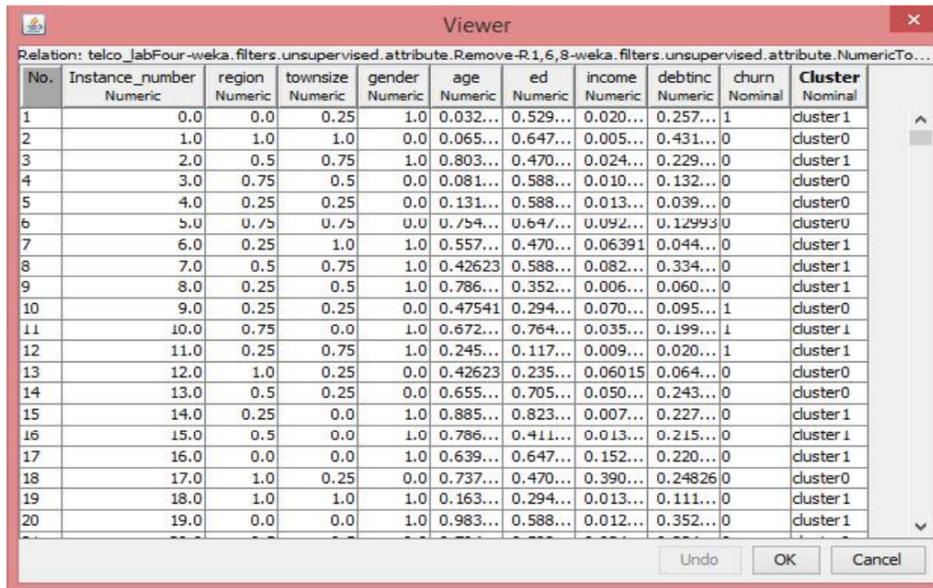


3. Você pode salvar os resultados da clusterização clicando no botão **Save** quando estiver visualizando o resultado da clusterização.



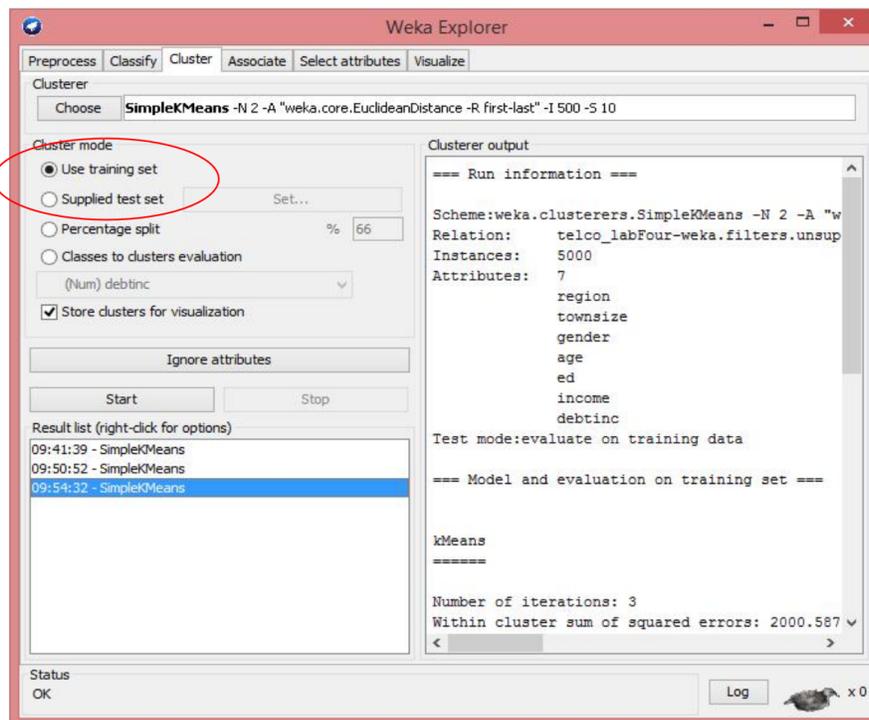
4. Os resultados são salvos em um arquivo **.arff**. Você pode usar o próprio Weka para abri-lo

e ver os resultados no novo dataset.



No.	Instance_number	region	townsize	gender	age	ed	income	debtinc	churn	Cluster
1	0.0	0.0	0.25	1.0	0.032...	0.529...	0.020...	0.257...	1	cluster1
2	1.0	1.0	1.0	0.0	0.065...	0.647...	0.005...	0.431...	0	cluster0
3	2.0	0.5	0.75	1.0	0.803...	0.470...	0.024...	0.229...	0	cluster1
4	3.0	0.75	0.5	0.0	0.081...	0.588...	0.010...	0.132...	0	cluster0
5	4.0	0.25	0.25	0.0	0.131...	0.588...	0.013...	0.039...	0	cluster0
6	5.0	0.75	0.75	0.0	0.754...	0.647...	0.092...	0.12993	0	cluster0
7	6.0	0.25	1.0	1.0	0.557...	0.470...	0.06391	0.044...	0	cluster1
8	7.0	0.5	0.75	1.0	0.42623	0.588...	0.082...	0.334...	0	cluster1
9	8.0	0.25	0.5	1.0	0.786...	0.352...	0.006...	0.060...	0	cluster1
10	9.0	0.25	0.25	0.0	0.47541	0.294...	0.070...	0.095...	1	cluster0
11	10.0	0.75	0.0	1.0	0.672...	0.764...	0.035...	0.199...	1	cluster1
12	11.0	0.25	0.75	1.0	0.245...	0.117...	0.009...	0.020...	1	cluster1
13	12.0	1.0	0.25	0.0	0.42623	0.235...	0.06015	0.064...	0	cluster0
14	13.0	0.5	0.25	0.0	0.655...	0.705...	0.050...	0.243...	0	cluster0
15	14.0	0.25	0.0	1.0	0.885...	0.823...	0.007...	0.227...	0	cluster1
16	15.0	0.5	0.0	1.0	0.786...	0.411...	0.013...	0.215...	0	cluster1
17	16.0	0.0	0.0	1.0	0.639...	0.647...	0.152...	0.220...	0	cluster1
18	17.0	1.0	0.25	0.0	0.737...	0.470...	0.390...	0.24826	0	cluster0
19	18.0	1.0	1.0	1.0	0.163...	0.294...	0.013...	0.111...	0	cluster1
20	19.0	0.0	0.0	1.0	0.983...	0.588...	0.012...	0.352...	0	cluster1

5. Se o dataset não tiver classe definida, quando você realizar a clusterização no dataset, escolha **Use Training Dataset** como Cluster mode.



Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Clusterer: Choose SimpleKMeans -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10

Cluster mode

- Use training set
- Supplied test set
- Percentage split
- Classes to clusters evaluation

Ignore attributes

Start Stop

Result list (right-click for options)

- 09:41:39 - SimpleKMeans
- 09:50:52 - SimpleKMeans
- 09:54:32 - SimpleKMeans

Clusterer output

```
=== Run information ===
Scheme:weka.clusterers.SimpleKMeans -N 2 -A "w
Relation: telco_labFour-weka.filters.unsup
Instances: 5000
Attributes: 7
    region
    townsize
    gender
    age
    ed
    income
    debtinc
Test mode:evaluate on training data

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

kMeans
=====

Number of iterations: 3
Within cluster sum of squared errors: 2000.587
```

```
kMeans
=====

Number of iterations: 3
Within cluster sum of squared errors: 2000.587162533113
Missing values globally replaced with mean/mode

Cluster centroids:

Attribute      Full Data      Cluster#
                (5000)         0           1
                (2482)        (2518)
=====
region         0.5004         0.5049      0.4958
townsize       0.4218         0.4184      0.4252
gender         0.5036         0           1
age            0.4758         0.4788      0.4729
ed             0.5025         0.5027      0.5024
income         0.043          0.0435      0.0425
debtinc       9.9542         9.9199      9.9879

Time taken to build model (full training data) : 0.09 s

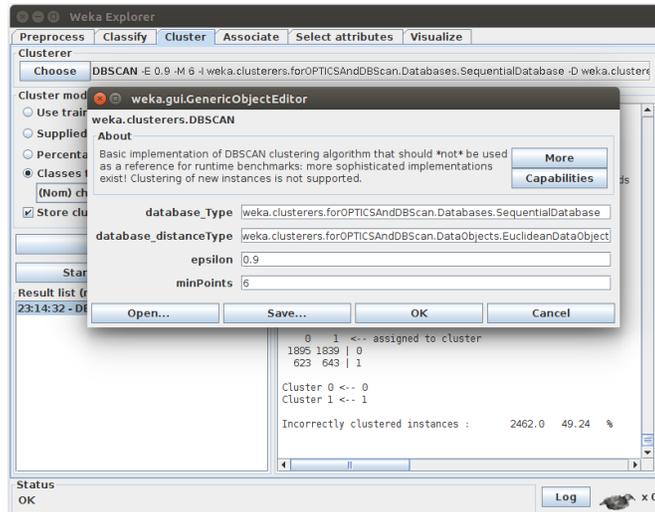
=== Model and evaluation on training set ===

Clustered Instances

0      2482 ( 50%)
1      2518 ( 50%)
```

DBScan

1. Agora vamos abordar o algoritmo **DBScan**. Da mesma forma que fizemos para **SimpleKMeans**, clique em **Classes to clusters evaluation** e selecione o último atributo (**churn**) como **classe**. Clique em **Store Clusters for visualization**. Clique em **Ignore attributes** e selecione **custIds**, **agecat**, **edcat** e o último atributo **churn** (classe, que não será clusterizado). Em seguida, clique em **Start**. Inicialmente vamos usar os valores default para **epsilon** e **minPoints**.



2. Experimente alterar o valor de **epsilon** para 0.3 e de **minPoints** para 100 e verifique o novo resultado da **Clusterização**.

