Laboratório
Classificação com o WEKA Explorer

Para esse laboratório considere os seguintes classificadores:

- C4.5 (J4.8)
- KNN
- Naïve Bayes

Você pode realizar o tutorial básico a partir da página abaixo, ou as atividades descritas no restante desta página são avançadas.

Considere as bases de treinamento e teste de **dígitos manuscrítos***

- digTrain1k.arff, digTrain2k.arff digTrain3k.arff digTrain4k.arff e digTrain5k.arff
- digTest5k.arff

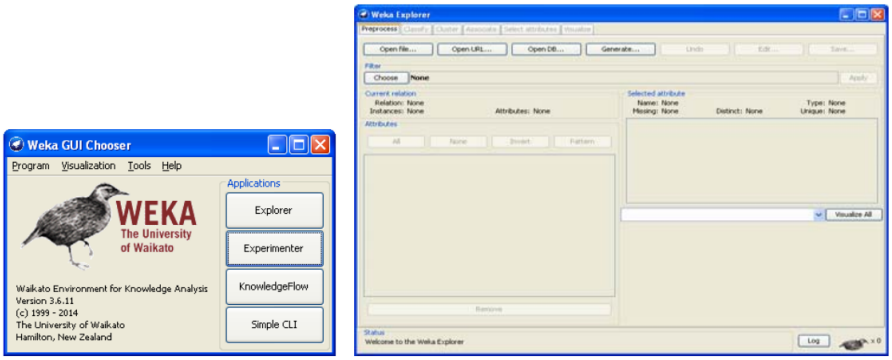
1. Compare o desempenho desses classificadores em função da disponibilidade de base de treinamento. Alimente os classificadores com blocos de 1000 exemplos e plote num gráfico o desempenho na base de testes e analise em qual ponto o tamanho da base de treinamento deixa de ser relevante.
2. Qual é o classificador que tem o melhor desempenho com poucos dados (1000 exemplos)?
3. Qual é o classificador que tem melhor desempenho com todos os dados?
4. Qual é o classificador mais rápido para classificar os 5k exemplos de teste.
5. O que você pode dizer a respeito das matrizes de confusão. Os erros são os mesmos para todos os classificadores quando todos eles utilizam toda a base de teste?

*disponível em: www.inf.ufpr.br/menotti/am-211/data.zip

Nas páginas abaixo, encontram-se tutoriais explicando como usar classificadores no Weka.

Preparando os dados para classificação

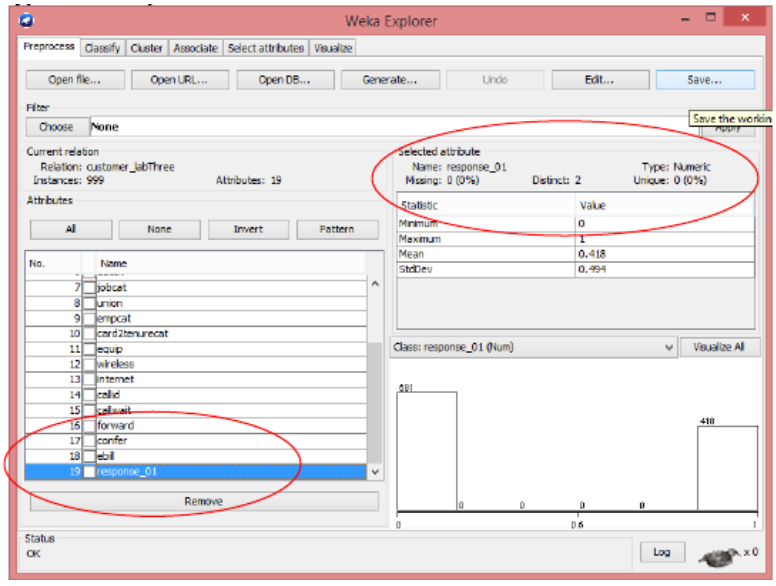
1. Inicie uma sessão do **Weka** ou execute em linha de comando: ***java –jar weka.jar***.
   Quando a **GUI Chooser** surgir, selecione o **Explorer** a partir das quatro opções do lado direito.



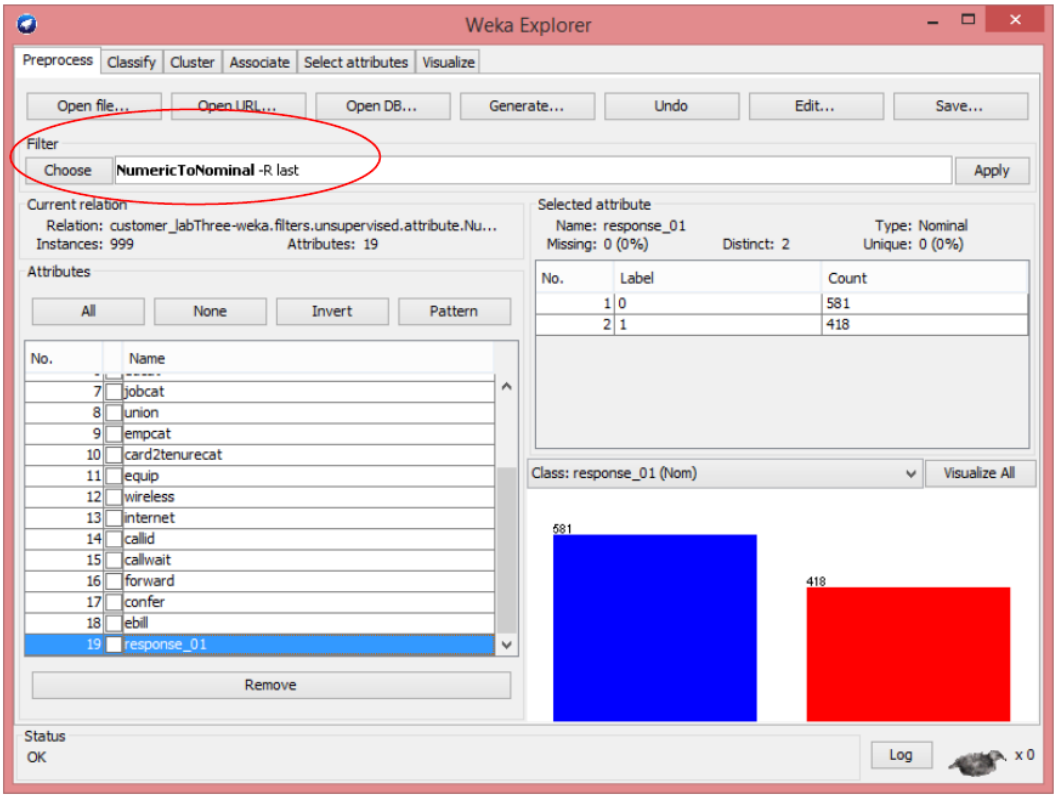
2. Estamos no **Preprocess** agora. Clique no botão **Open** para abrir a caixa de diálogo padrão através da qual você pode selecionar um arquivo. Escolha o arquivo **customer_lab2.csv**.

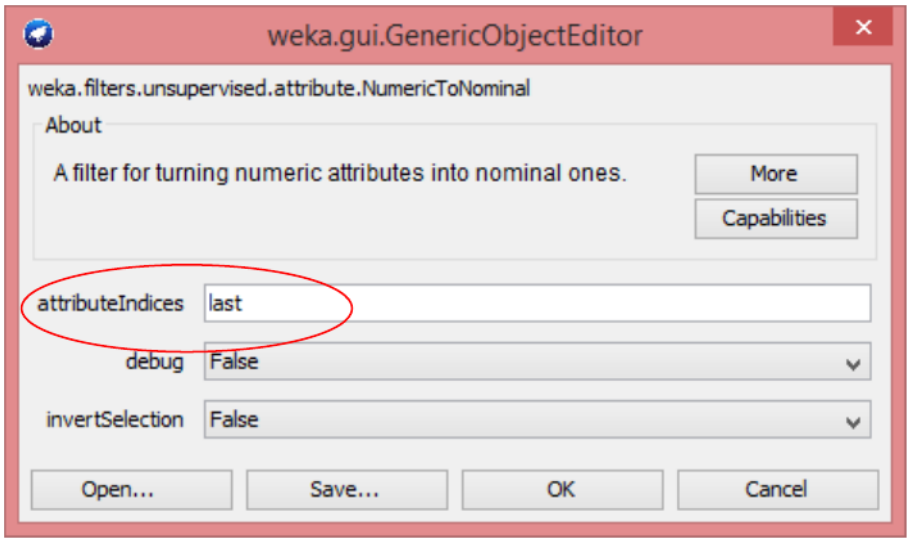
1. Elegendo o atributo meta ou classe

1. Para realizar a classificação com Weka, o último atributo no conjunto de dados é considerado como **classe** / **meta** e deve ser **nominal**. Como o último atributo do dataset **customer_lab2.csv** é do tipo numérico (1/0), devemos convertê-lo para o tipo nominal - próximo passo.

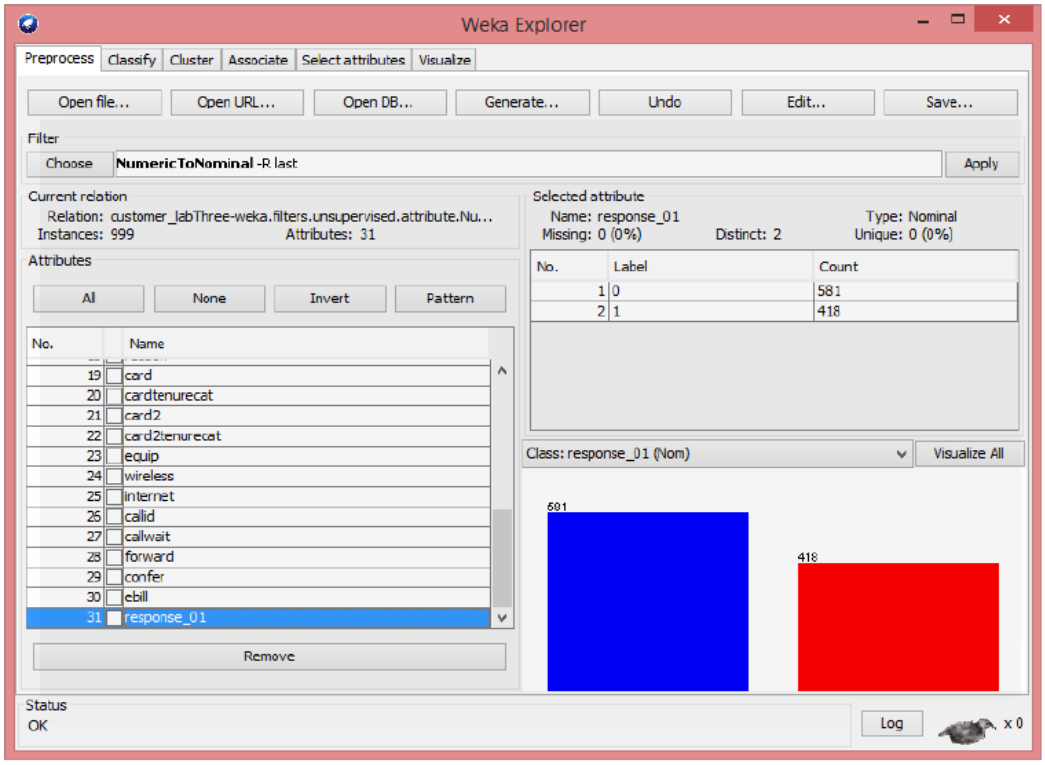


2. O filtro de atributo não supervisionado **NumericToNominal** é escolhido para executar esta conversão. Como gostaríamos de converter apenas o último atributo, altere o **attributeIndices** para **last**.

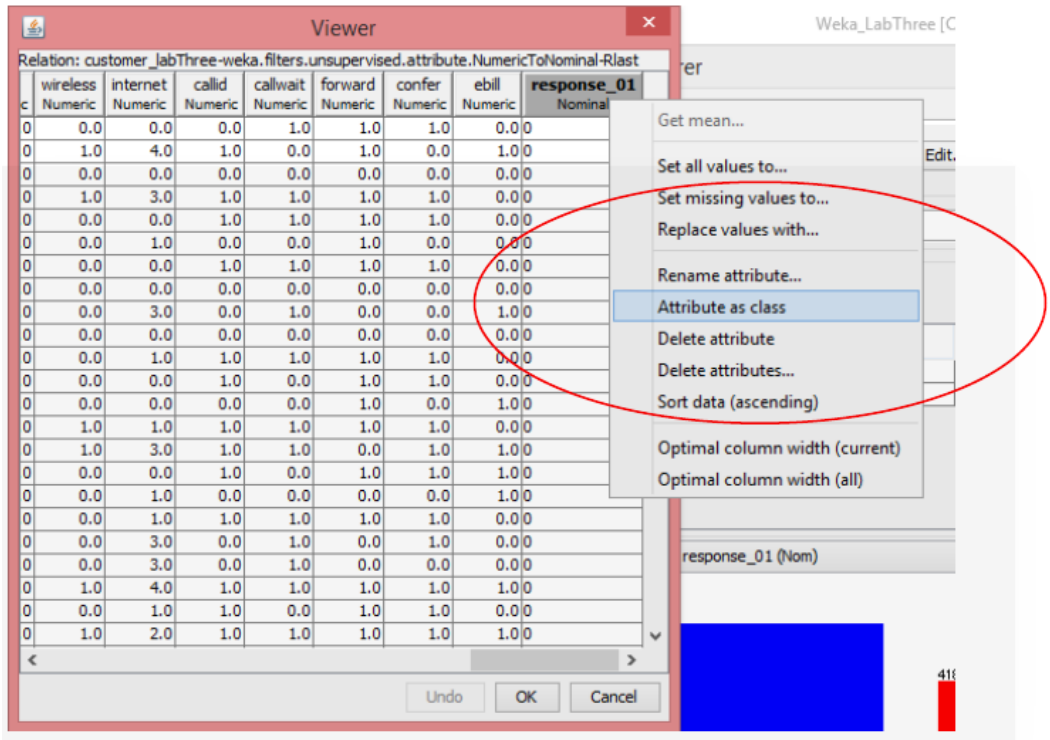

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
NumericToNominal -R last
Apply
Current relation
Relation: customer_labThree-weka.filters.unsupervised.attribute.Nu...
Instances: 999
Attributes: 19
Selected attribute
Name: response_01
Type: Nominal
Missing: 0 (0%)
Distinct: 2
Unique: 0 (0%)
Attributes
All
None
Invert
Pattern
No.
Name
7 jobcat
8 union
9 empcat
10 card2tenurecat
11 equip
12 wireless
13 internet
14 callid
15 callwait
16 forward
17 confer
18 ebill
19 response_01
Remove
No.
Label
Count
1 0 581
2 1 418
Class: response_01 (Nom)
Visualize All
581
418
Status
OK
Log
x 0


weka.gui.GenericObjectEditor
weka.filters.unsupervised.attribute.NumericToNominal
About
A filter for turning numeric attributes into nominal ones.
More
Capabilities
attributeIndices
last
debug
False
invertSelection
False
Open...
Save...
OK
Cancel

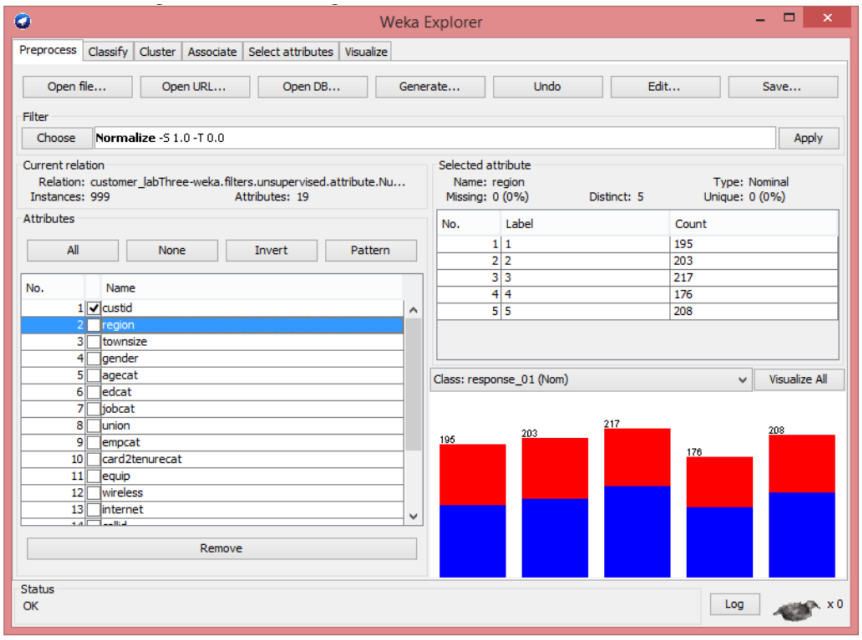
3. Depois de aplicar o filtro, o último atributo torna-se **nominal** e é considerado como o rótulo de classe para o dataset - agora o dataset é visualizado em duas cores.



4. Se o atributo **classe** não for o último atributo, você poderá definí-lo na janela de edição (**Edit**).

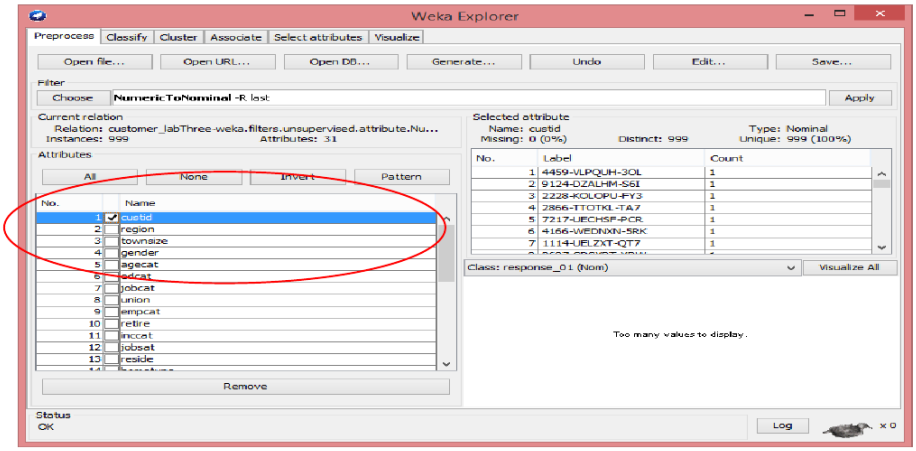


5. Você também deve converter os tipos de outros atributos. Os atributos **region**, **townsize**, **agecat**, **edcat**, **jobcat**, **empcat**, **card2tenurecat** e **internet** são todos valores nominais, no entanto, eles são tratados como tipo numérico pelo Weka. E os atributos **gender**, **union**, **equip**, **wireless**, **callid**, **callwait**, **forward**, **confer**, e **ebill** são todos de valores binários, e também são tratados como tipos numéricos pelo Weka. O filtro **NumericToNominal** deve ser aplicado para convertê-los. Você também pode **Normalize** o atributo **educat** para [0, 1], já que as categorias de educação são rankings.

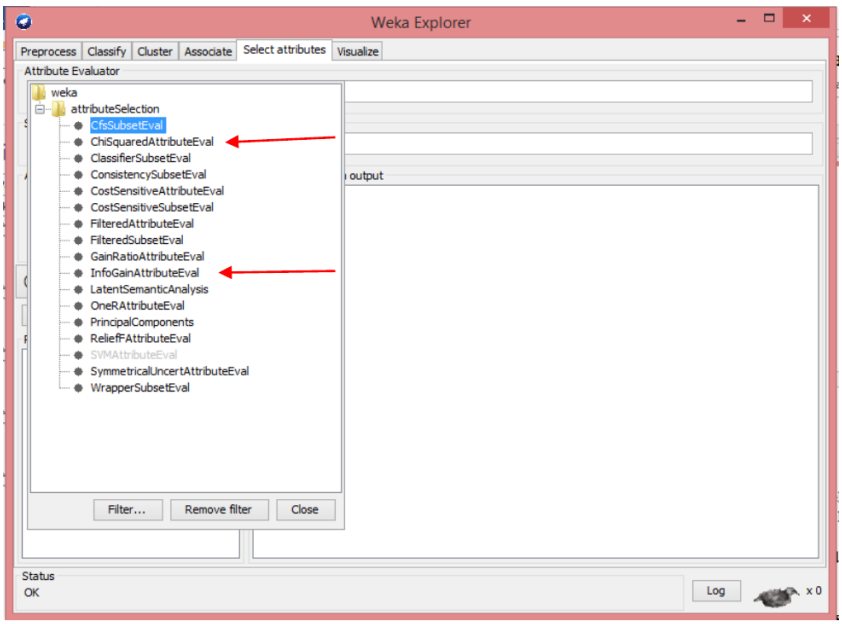


2. Seleção de Atributos (Select attributes) - Como nem todos os atributos são relevantes para o trabalho de classificação, você deve executar a seleção de atributos antes de treinar o classificador.

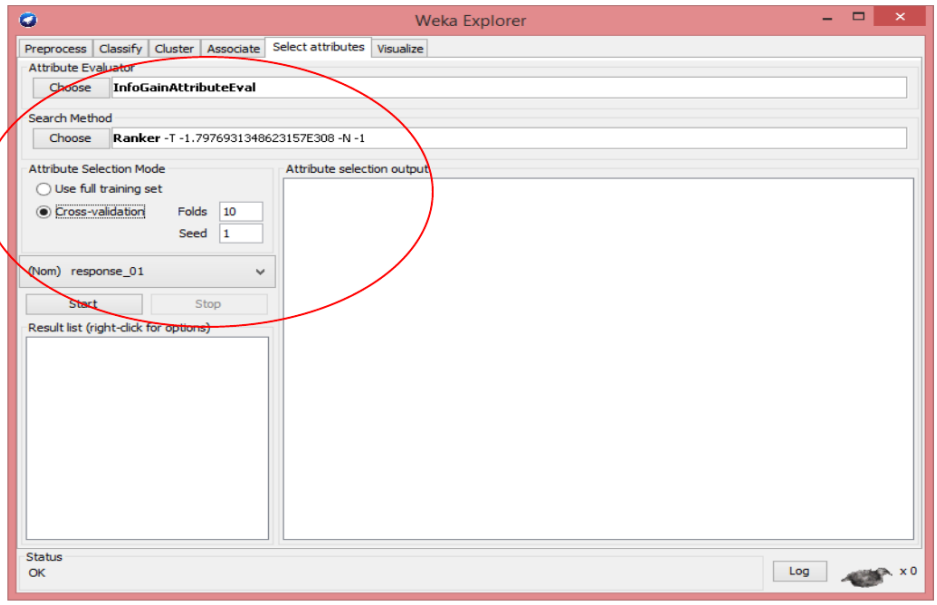
1. Você pode remover atributos irrelevantes à mão. Por exemplo, o primeiro atributo **custId** deve ser removido. Selecione-o e clique no botão **Remove** para removê-lo.

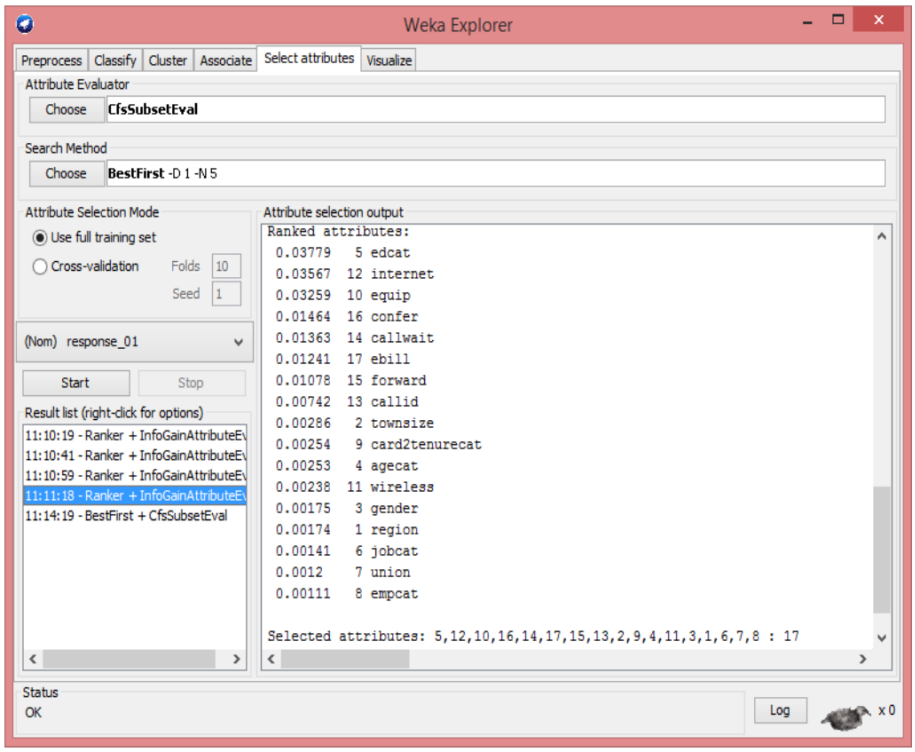


2. Você também pode executar a seleção automática de atributos. Introduzimos dois métodos de avaliação de atributos de forma individual - **InfoGainAttributeEval** e **ChiSquaredAttributeEval**. O método de seleção de atributo padrão de Weka é **CfsSubsetEval**, que avalia subconjuntos de atributos.

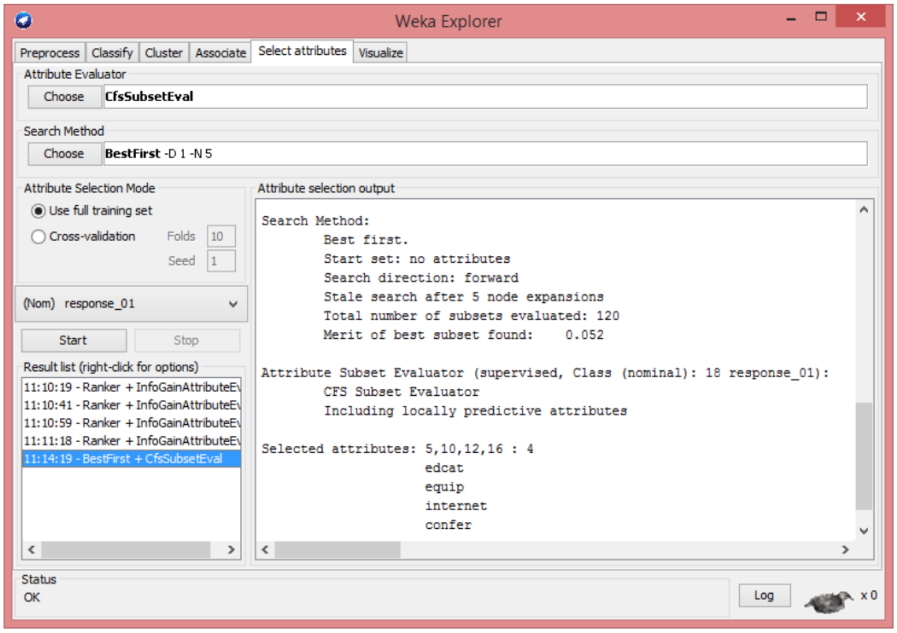


3. Para usar o avaliador **InfoGainAttributeEval**, um método de busca **Ranker** é selecionado para ordenar todos os atributos usando o resultado da avaliação. Usamos todo dataset como conjunto de treinamento. Os resultados mostram que os primeiros 8 atributos são bons.

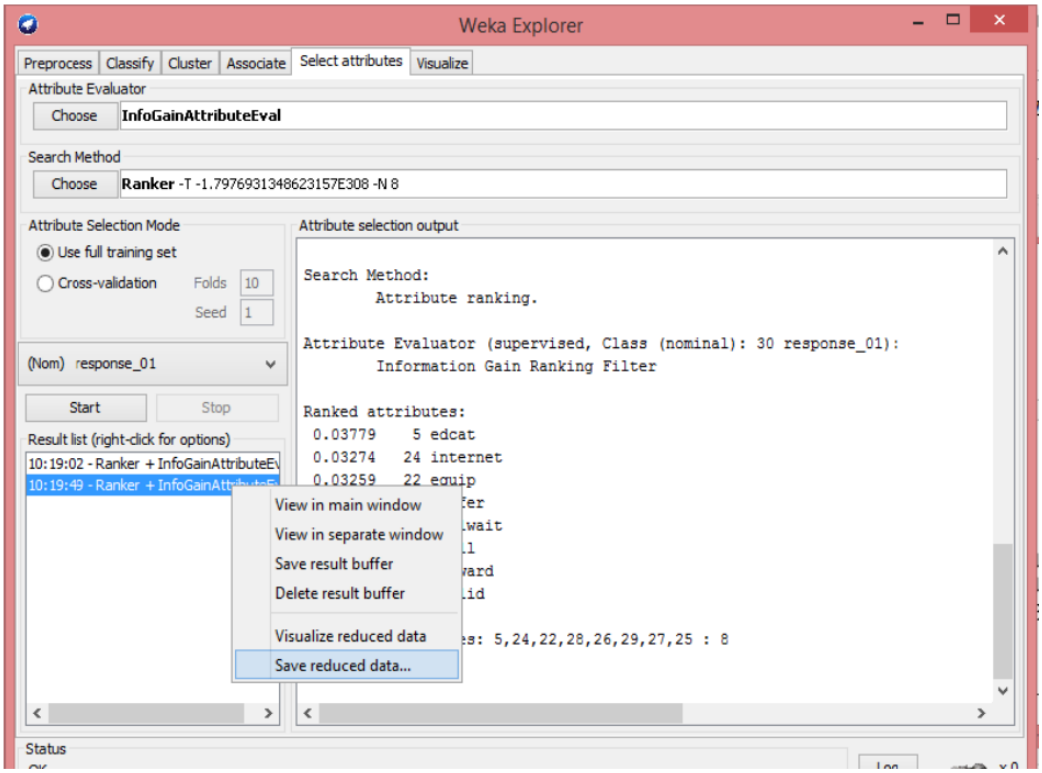




4. Execute a seleção de atributos uma segunda vez mas agora usando o avaliador **CfsSubsetEval** com o método de busca **BestFirst**. Compare os resultados dos dois métodos de seleção de atributos.

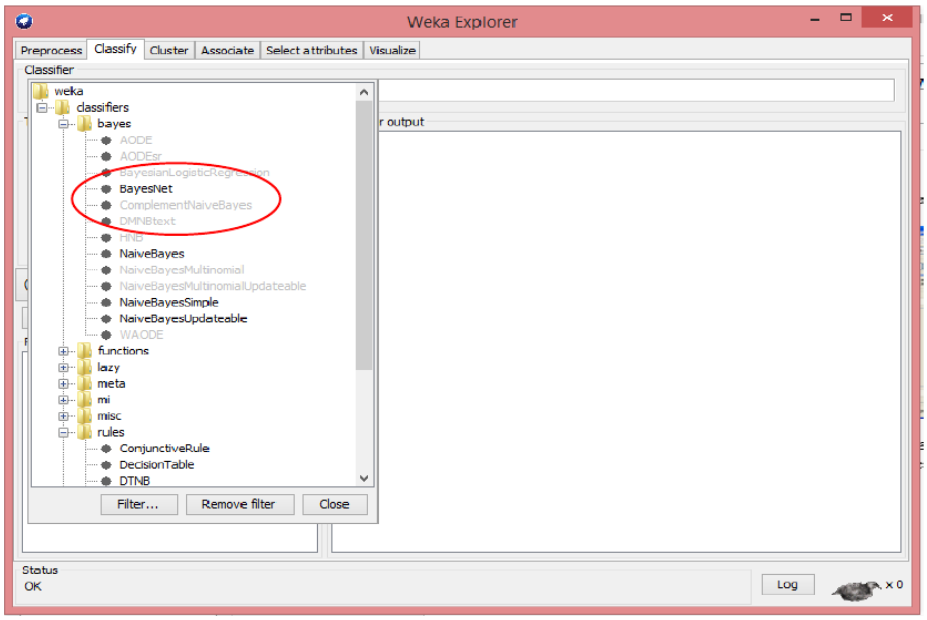


5. Se você decidir reduzir o conjunto de dados removendo atributos sem importância, escolha **Save reduced data...** clicando com o botão Direito em **Result list**. Salve o arquivo com o nome **customer.arff**.

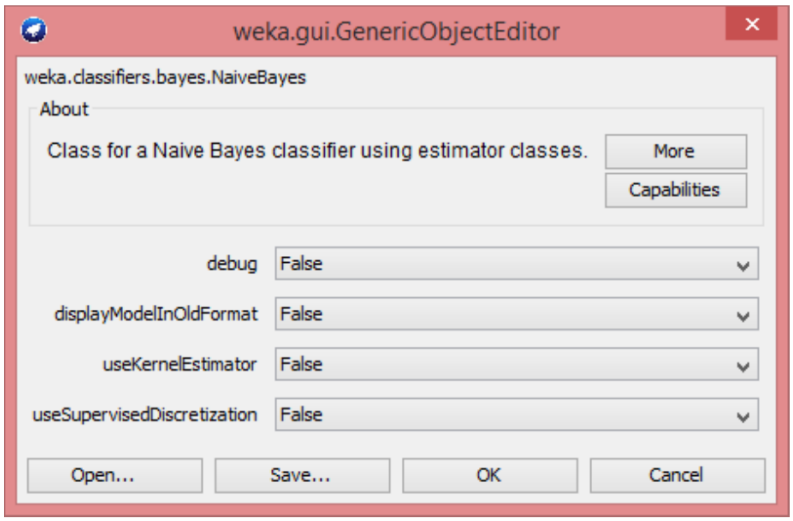


3. Classificador Naïve Bayes: bayes / NaïveBayes

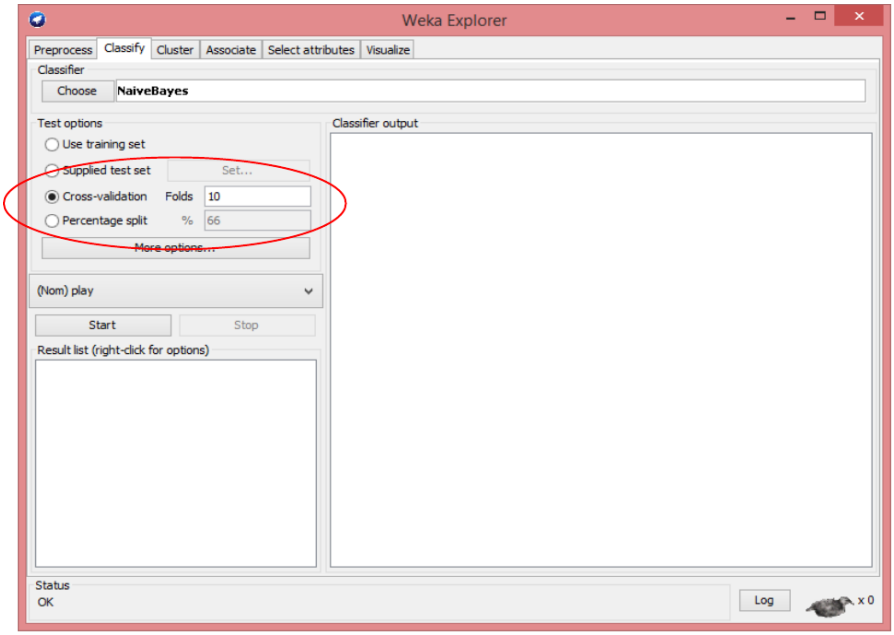
1. Abra o dataset salvo **customer.arff** e clique na guia **Classify** na parte superior da janela. Clique no botão Choose abaixo de ***Classifier***. A lista drop-down de todos os classificadores são exibidos. Escolha **NaiveBayes** da pasta **bayes**.



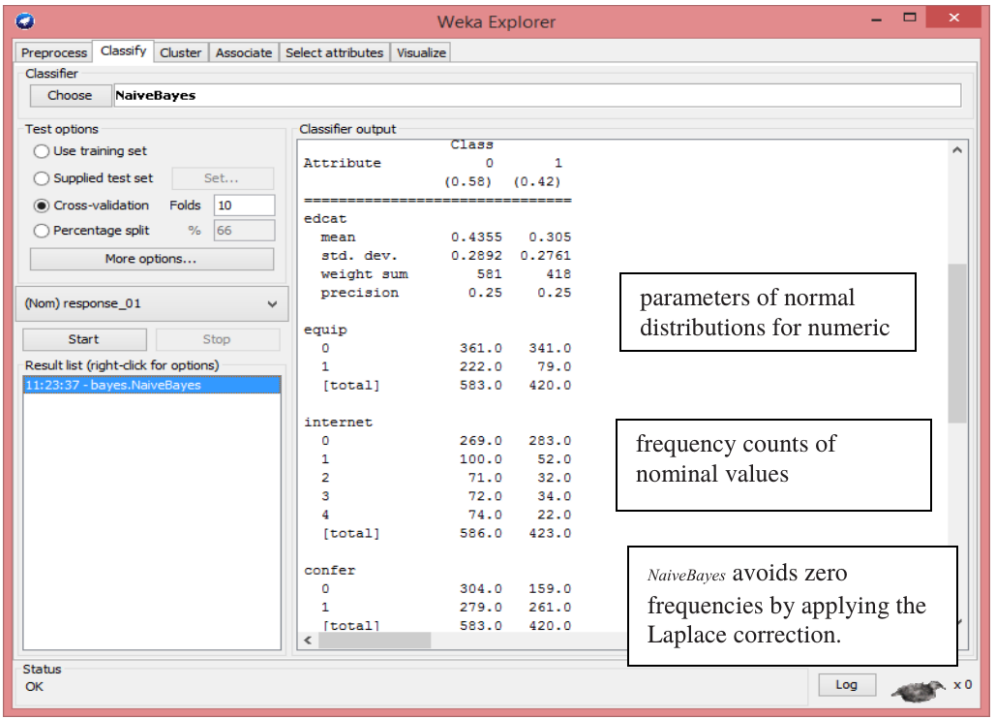
2. Clique (com o botão **Esquerdo**) dentro da caixa **Filter**, e então a janela de propriedades é apresentada. Então, a janela de propriedades do **NaiveBayes** será aberta, se você não quiser usar a Distribuição Normal para dados numéricos, defina **useKernelEstimator** para **true**; Você também pode realizar discretização supervisionada em dados numéricos definindo **useSupervisedDiscretization** como **true**. Clique no botão **OK** para salvar todas as configurações.

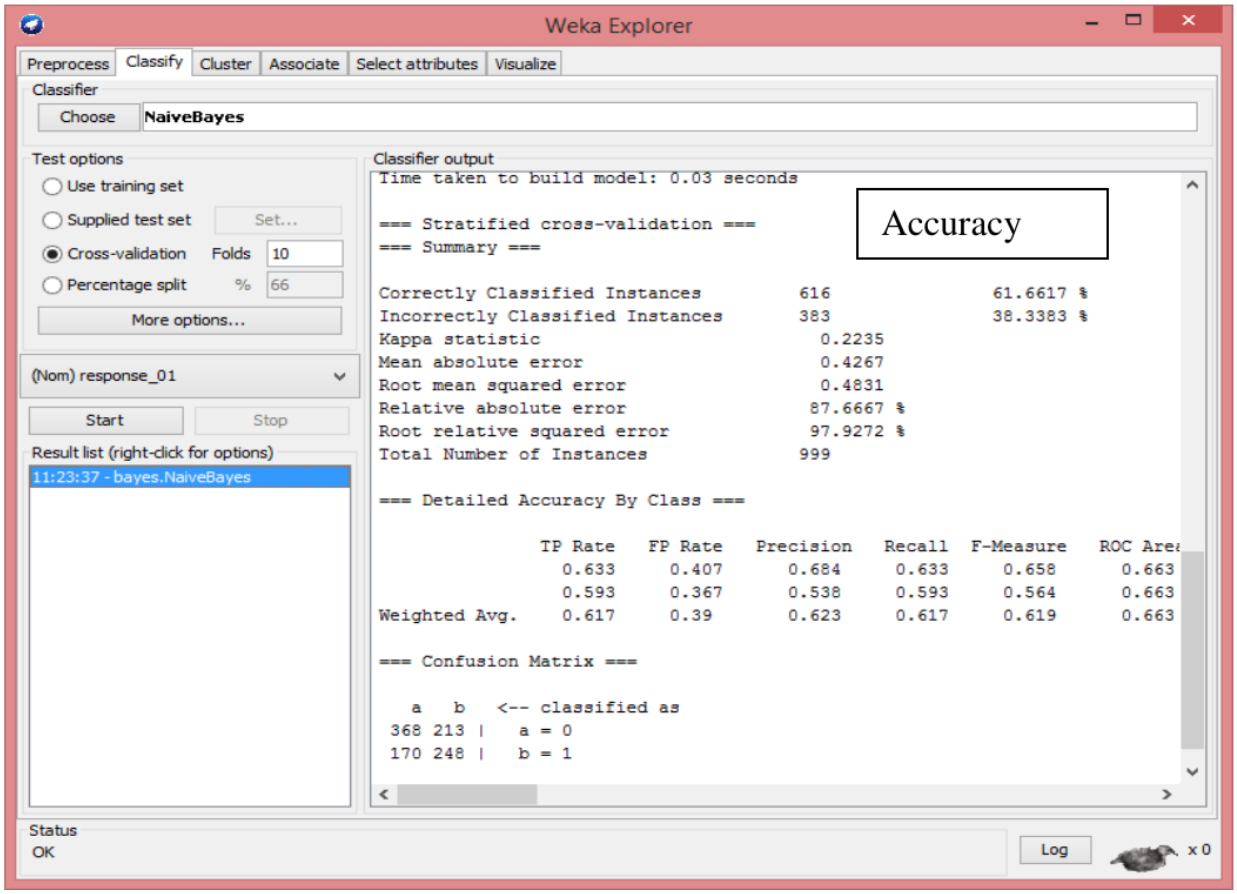


3. Para dividir o dataset em **training set** and **testing set**, escolha a **Cross-validation** de 10 vezes. Para usar conjuntos de **training**, **validation** and **testing set**, escolha **Supplied test set**, após realizar **Cross-validation**. Neste formato de **Cross-validation**+**Supplied test set**, todo dataset inicial é usado para **training** e **validation**, e o **testing set** é aquele escolhido .



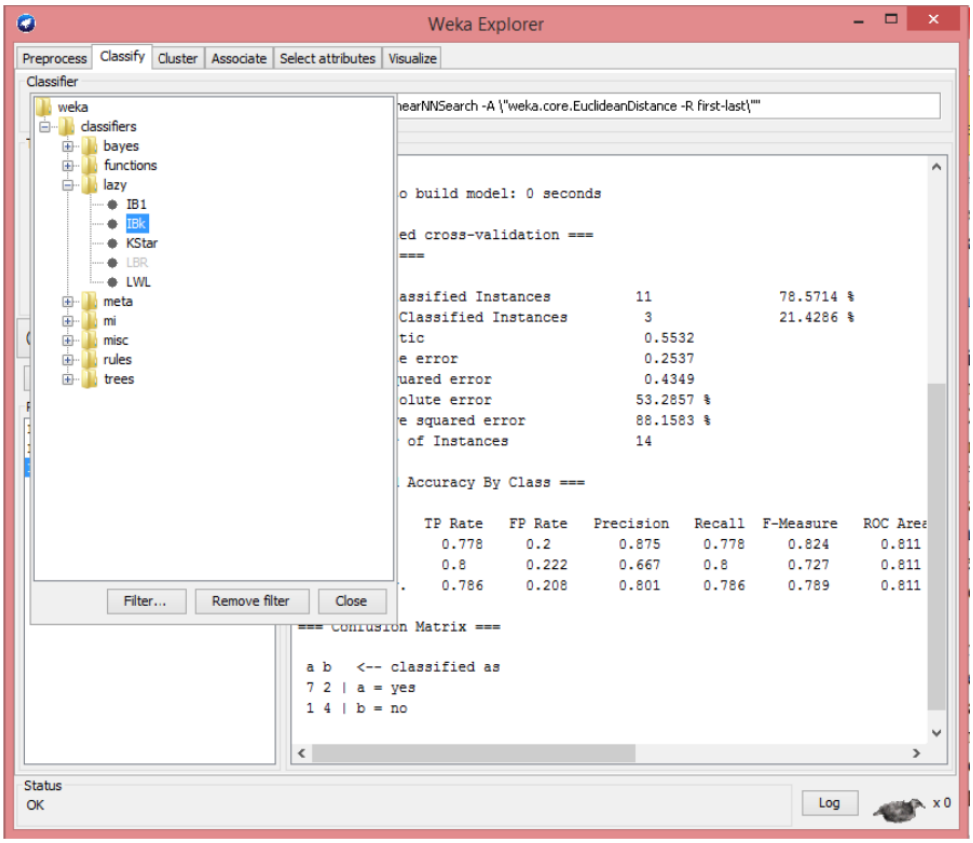
4. Clique no botão **Start**, à esquerda da janela, e então o algoritmo começa a ser executado. A saída é apresentada na janela da direita.

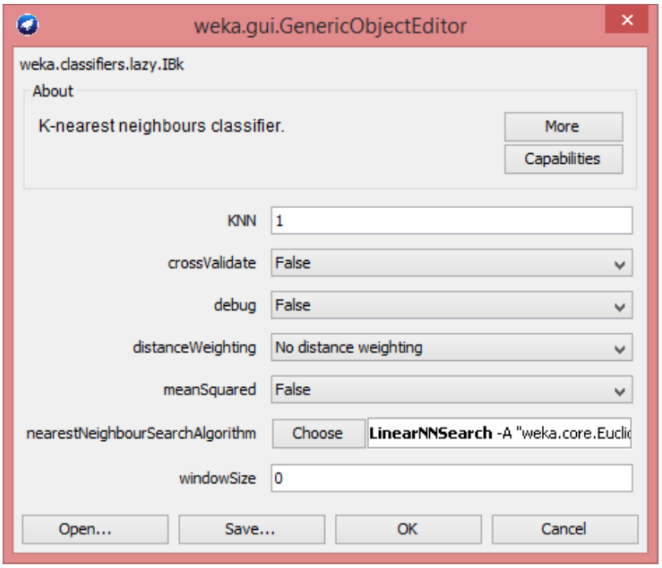


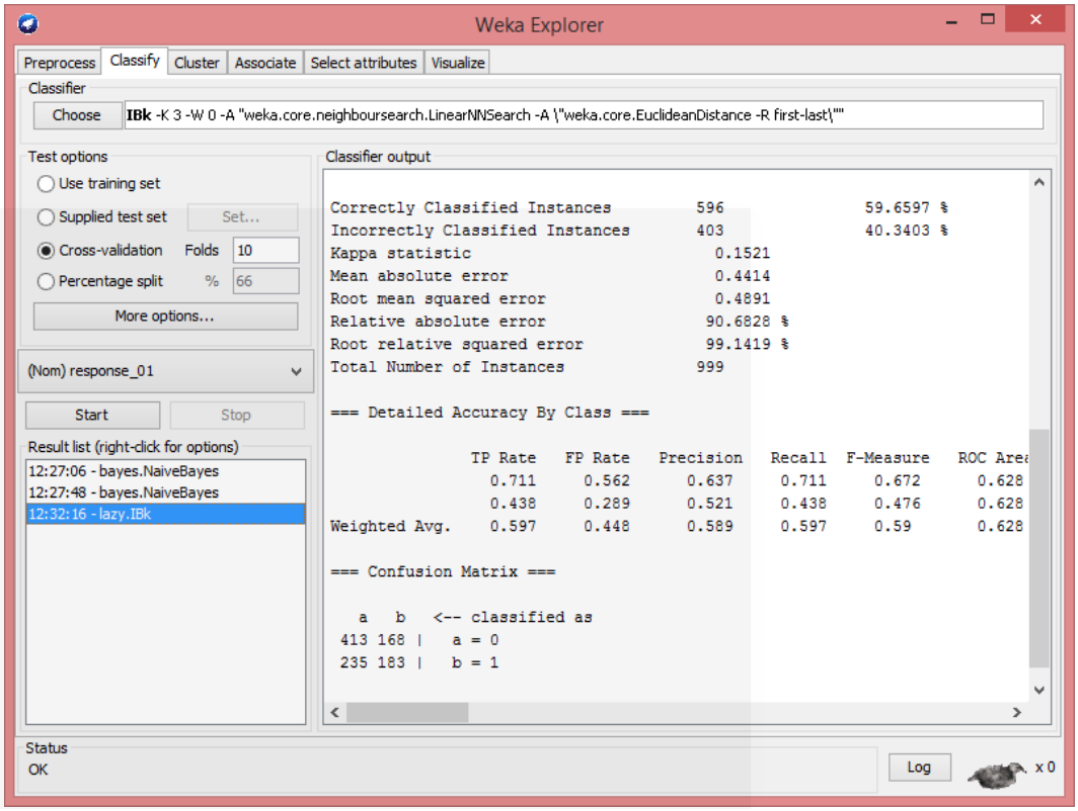

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) response_01
Start
Stop
Result list (right-click for options)
11:23:37 - bayes.NaiveBayes
Classifier output
Time taken to build model: 0.03 seconds
=== Stratified cross-validation ===
=== Summary ===
Accuracy
Correctly Classified Instances 616 61.6617 %
Incorrectly Classified Instances 383 38.3383 %
Kappa statistic 0.2235
Mean absolute error 0.4267
Root mean squared error 0.4831
Relative absolute error 87.6667 %
Root relative squared error 97.9272 %
Total Number of Instances 999
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure ROC Area
0.633 0.407 0.684 0.633 0.658 0.663
0.593 0.367 0.538 0.593 0.564 0.663
Weighted Avg. 0.617 0.39 0.623 0.617 0.619 0.663
=== Confusion Matrix ===
a b <-- classified as
368 213 | a = 0
170 248 | b = 1
Status
OK
Log
x 0

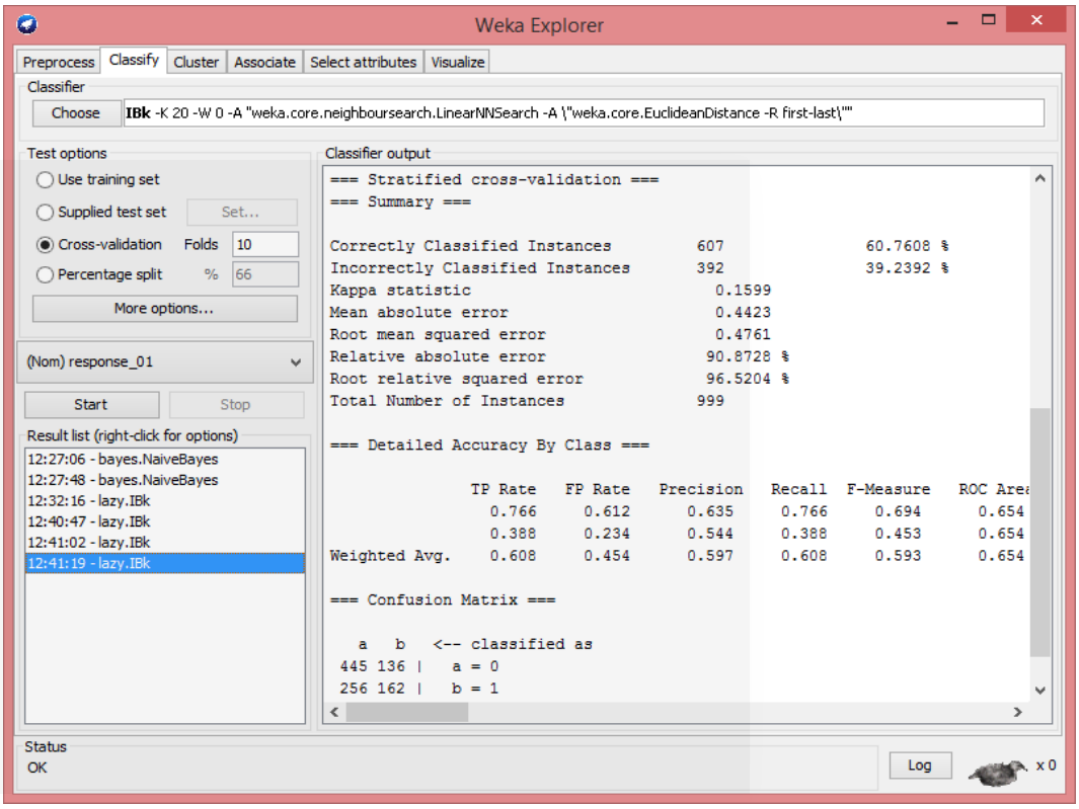
K-Nearest-Neighbor: lazy/IBK

1. Gostaríamos de realizar a classificação K-Nearest-Neighbor no mesmo dataset.
   Para isso você deverá escolher: **classifiers** => **lazy** => **IBK**.
   Você pode experimentar valores diferentes de K e ver qual valor dá um resultado melhor. Compare os resultados com o classificador Naïve Bayes.



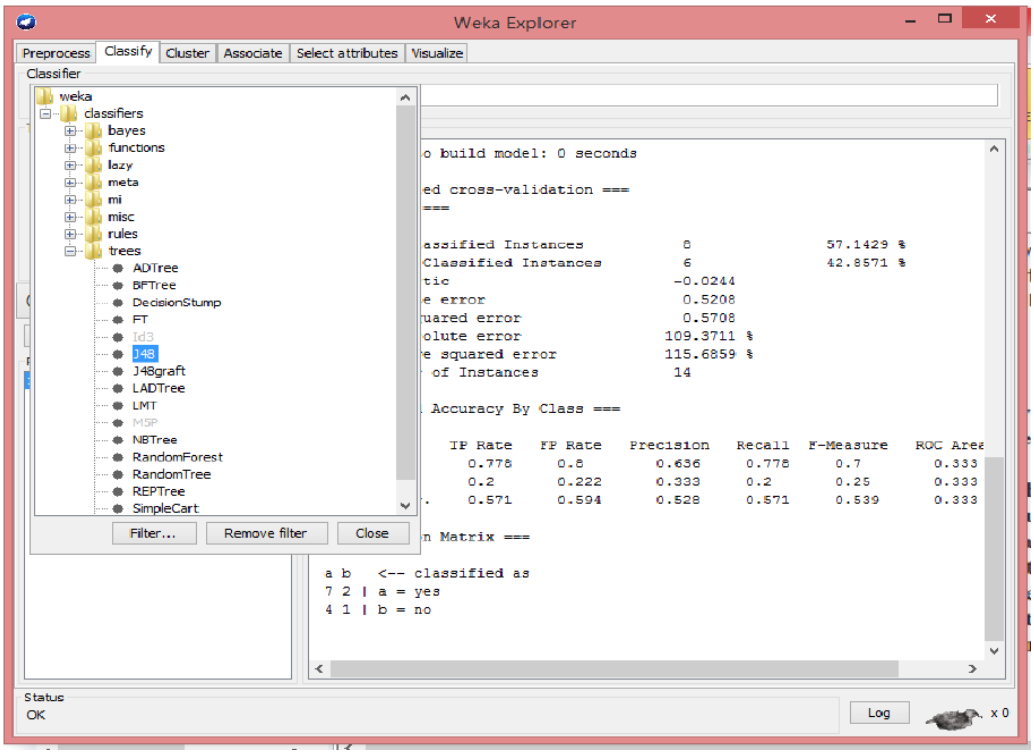


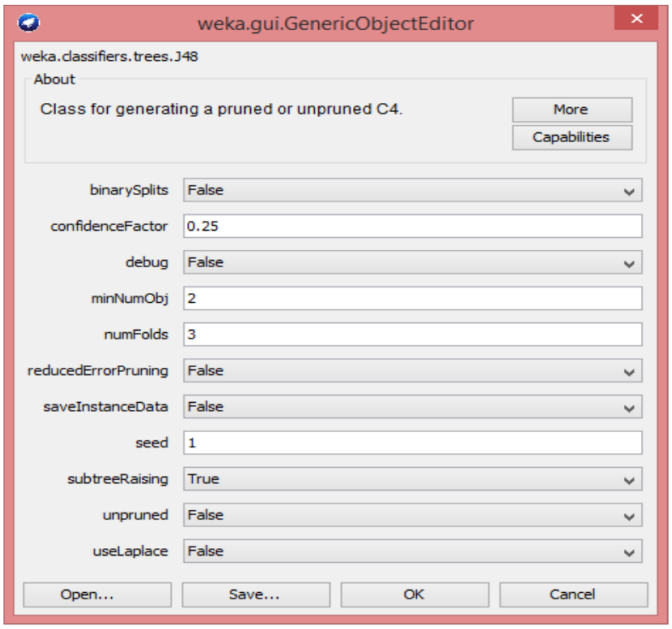

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
IBk -K 3 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) response_01
Start
Stop
Result list (right-click for options)
12:27:06 - bayes.NaiveBayes
12:27:48 - bayes.NaiveBayes
12:32:16 - lazy.IBk
Classifier output
Correctly Classified Instances 596 59.6597 %
Incorrectly Classified Instances 403 40.3403 %
Kappa statistic 0.1521
Mean absolute error 0.4414
Root mean squared error 0.4891
Relative absolute error 90.6828 %
Root relative squared error 99.1419 %
Total Number of Instances 999
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure ROC Area
0.711 0.562 0.637 0.711 0.672 0.628
0.438 0.289 0.521 0.438 0.476 0.628
Weighted Avg. 0.597 0.448 0.589 0.597 0.59 0.628
=== Confusion Matrix ===
a b <-- classified as
413 168 | a = 0
235 183 | b = 1
Status
OK
Log
x 0

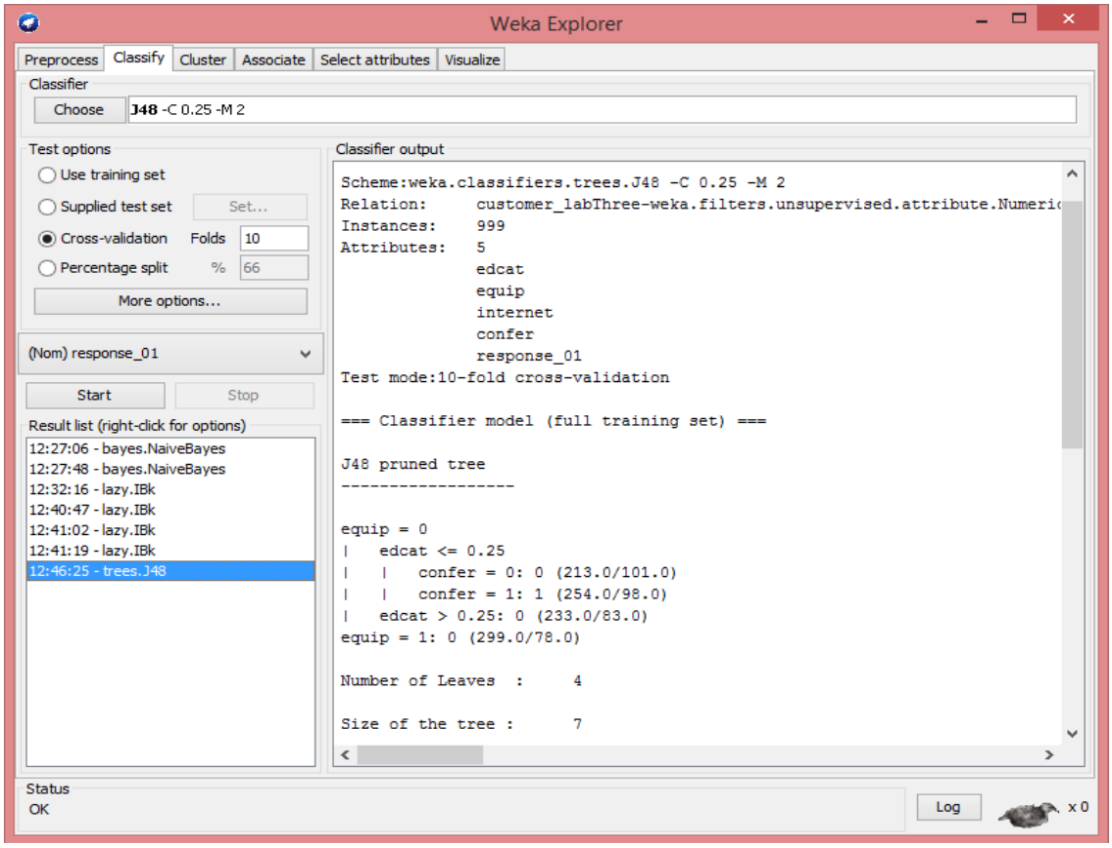

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
IBk -K 20 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) response_01
Start
Stop
Result list (right-click for options)
12:27:06 - bayes.NaiveBayes
12:27:48 - bayes.NaiveBayes
12:32:16 - lazy.IBk
12:40:47 - lazy.IBk
12:41:02 - lazy.IBk
12:41:19 - lazy.IBk
Classifier output
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 607 60.7608 %
Incorrectly Classified Instances 392 39.2392 %
Kappa statistic 0.1599
Mean absolute error 0.4423
Root mean squared error 0.4761
Relative absolute error 90.8728 %
Root relative squared error 96.5204 %
Total Number of Instances 999
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure ROC Area
0.766 0.612 0.635 0.766 0.694 0.654
0.388 0.234 0.544 0.388 0.453 0.654
Weighted Avg. 0.608 0.454 0.597 0.608 0.593 0.654
=== Confusion Matrix ===
a b <-- classified as
445 136 | a = 0
256 162 | b = 1
Status
OK
Log
x 0

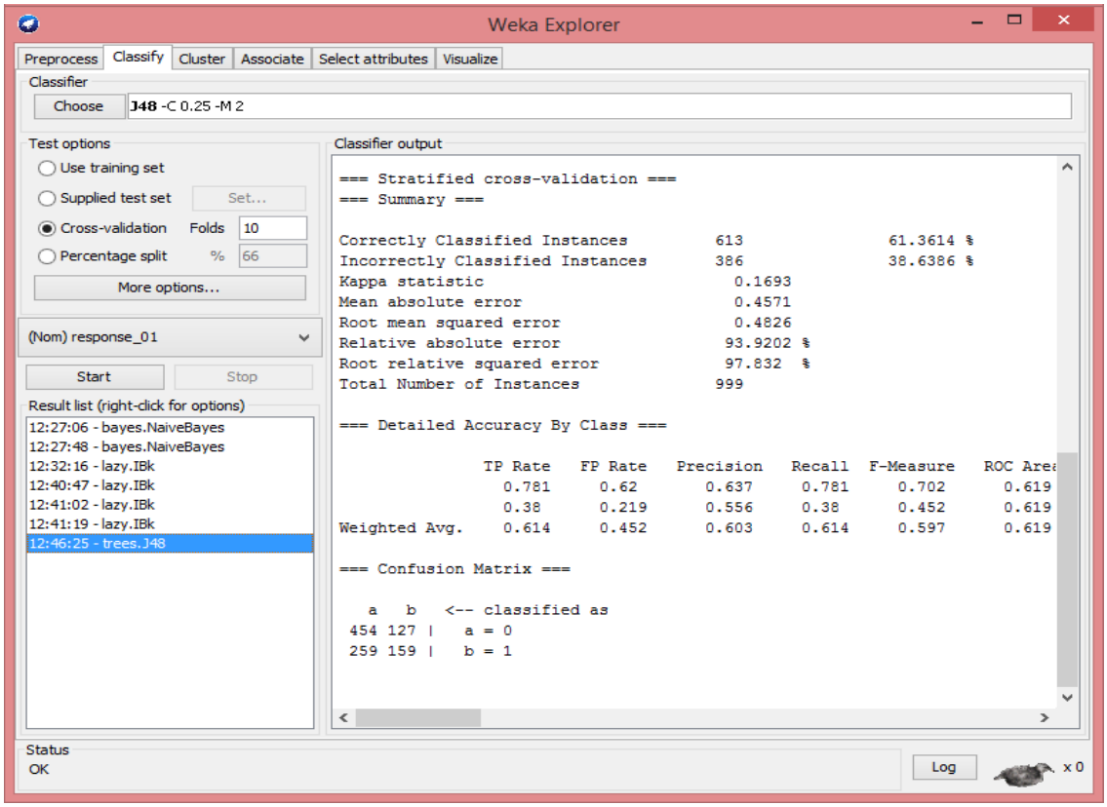
Árvores de Decisão: trees/J48 (Implementing C4.5)

1. Gostaríamos de construir um modelo de árvore de decisão no mesmo dataset de treinamento.
   Para isso você deverá escolher: **classifiers** => **lazy** => **IBK**.
   Utilize todos os valores padrão dos parâmetros e depois gere diferentes árvores de decisão mudando estes parâmetros (**confidenceFactor**, **minNumObj** e **numFolds**).

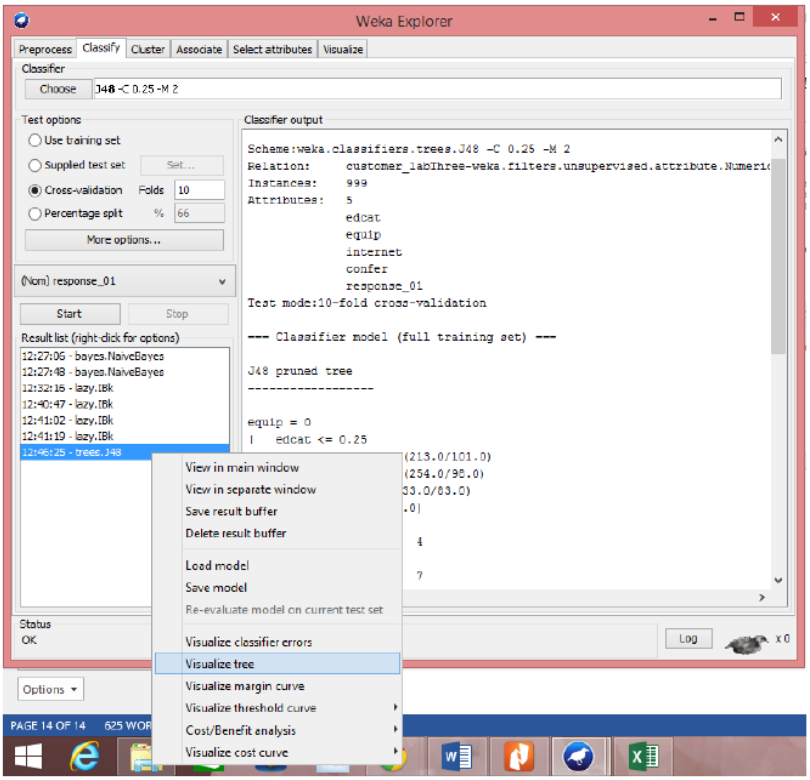


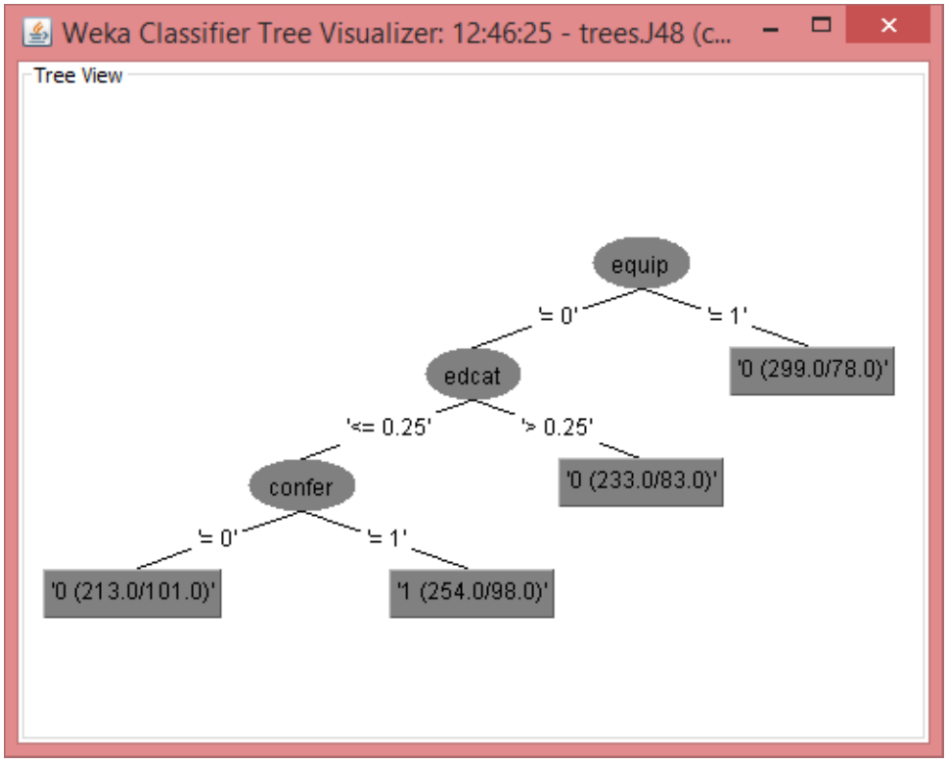



Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) response_01
Start
Stop
Result list (right-click for options)
12:27:06 - bayes.NaiveBayes
12:27:48 - bayes.NaiveBayes
12:32:16 - lazy.IBk
12:40:47 - lazy.IBk
12:41:02 - lazy.IBk
12:41:19 - lazy.IBk
12:46:25 - trees.J48
Classifier output
Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: customer_labThree-weka.filters.unsupervised.attribute.Numeric
Instances: 999
Attributes: 5
edcat
equip
internet
confer
response_01
Test mode:10-fold cross-validation
=== Classifier model (full training set) ===
J48 pruned tree
equip = 0
| edcat <= 0.25
| | confer = 0: 0 (213.0/101.0)
| | confer = 1: 1 (254.0/98.0)
| edcat > 0.25: 0 (233.0/83.0)
equip = 1: 0 (299.0/78.0)
Number of Leaves : 4
Size of the tree : 7
Status
OK
Log
x 0

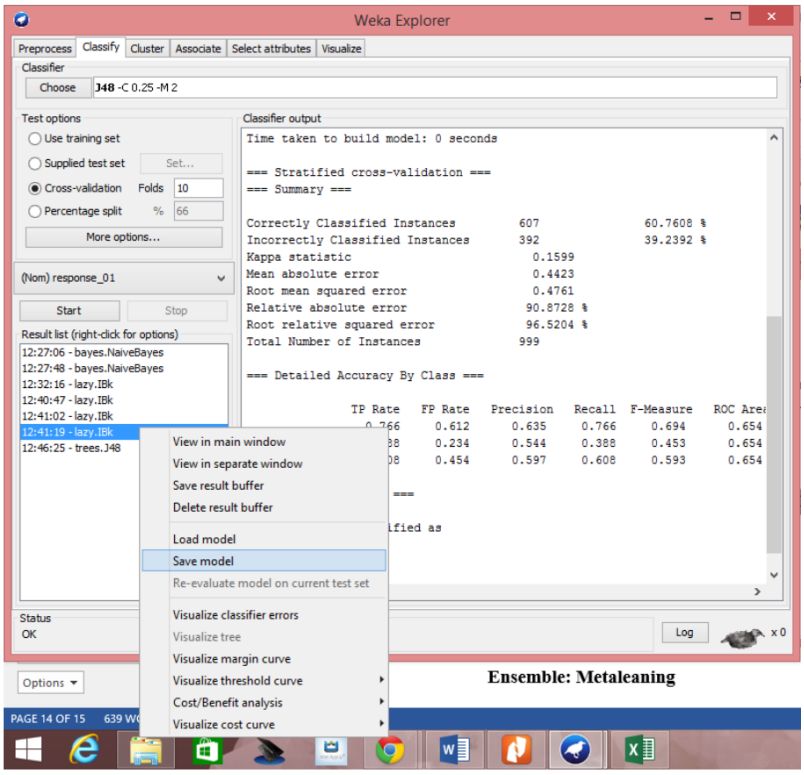

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) response_01
Start
Stop
Result list (right-click for options)
12:27:06 - bayes.NaiveBayes
12:27:48 - bayes.NaiveBayes
12:32:16 - lazy.IBk
12:40:47 - lazy.IBk
12:41:02 - lazy.IBk
12:41:19 - lazy.IBk
12:46:25 - trees.J48
Classifier output
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 613 61.3614 %
Incorrectly Classified Instances 386 38.6386 %
Kappa statistic 0.1693
Mean absolute error 0.4571
Root mean squared error 0.4826
Relative absolute error 93.9202 %
Root relative squared error 97.832 %
Total Number of Instances 999
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure ROC Area
0.781 0.62 0.637 0.781 0.702 0.619
0.38 0.219 0.556 0.38 0.452 0.619
Weighted Avg. 0.614 0.452 0.603 0.614 0.597 0.619
=== Confusion Matrix ===
a b <-- classified as
454 127 | a = 0
259 159 | b = 1
Status
OK
Log
x 0

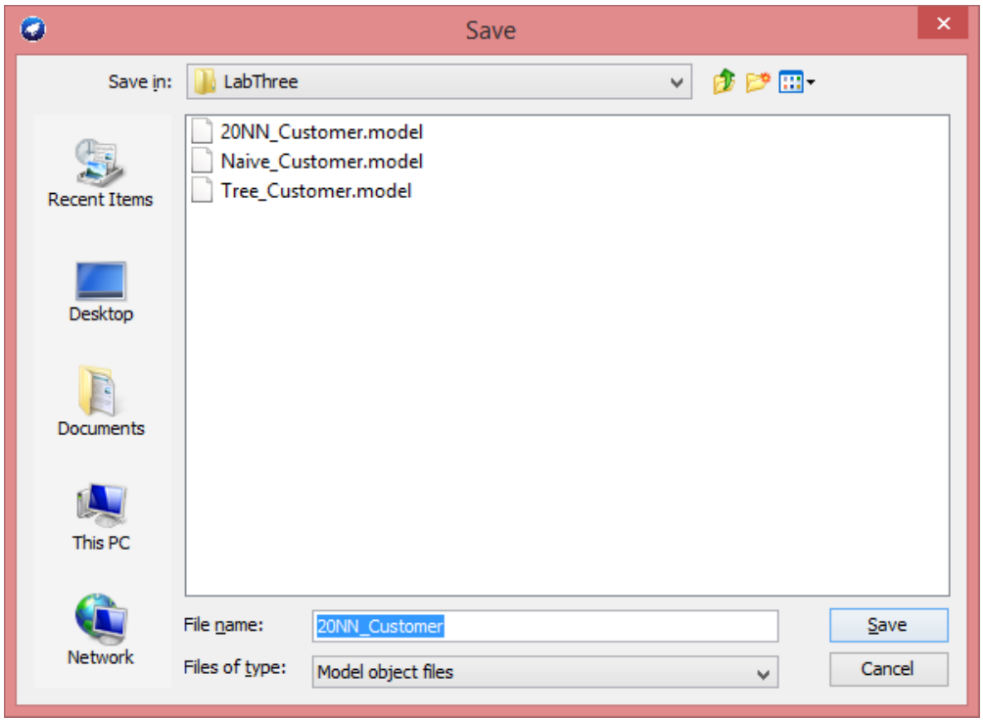
2. Para visualizar a árvore de decisão que construímos, clique com o botão **Direito** no item **trees.J48** da lista de resultados.





3. Os modelos de classificação treinados podem ser salvos clicando com o botão **Direito** nos itens da lista de resultados.





Ensemble (Metaleaning) classifier.meta.Voting

1. Você pode combinar vários classificadores para executar um método conjunto.
   Para isso você deverá escolher: **classifiers** => **meta** => **Vote**.