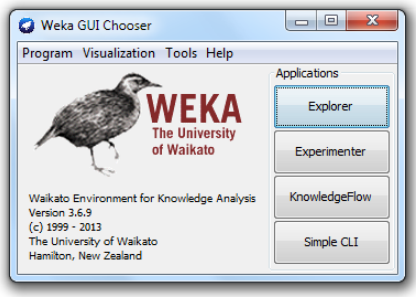
UTILIZANDO O SOFTWARE WEKA

O que é

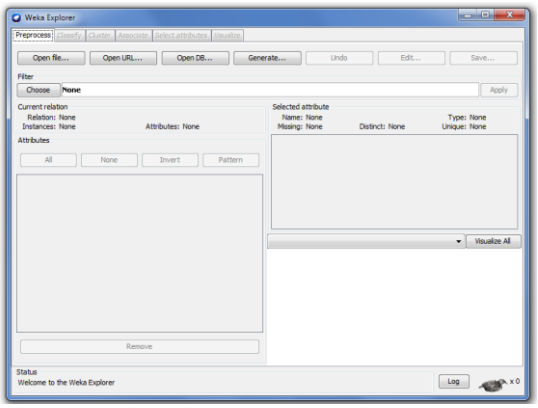
- Weka: software livre para mineração de dados
  - Desenvolvido por um grupo de pesquisadores
    - Universidade de Waikato, Nova Zelândia
    - Também é um pássaro típico da Nova Zelândia
  - Pontos fortes
    - Classificação
    - regras de associação
    - clusters de dados



Weka Explorer

3

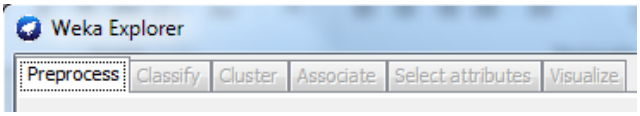
- Interface gráfica que permite a execução dos algoritmos de data mining da Weka de forma interativa



Weka Explorer

4

- Opções disponíveis
  - **Preprocess**: escolhe e modifica os dados utilizados
  - **Classify**: treina e testa sistemas de aprendizagem que classificam ou realizar regressão
  - **Cluster**: análise de clusters
  - **Associate**: permite aprender regras de associação para os dados
  - **Select attributes**: seleciona os atributos mais relevantes nos dados
  - **Visualize**: gráfico 2D interativo dos dados



Weka Explorer

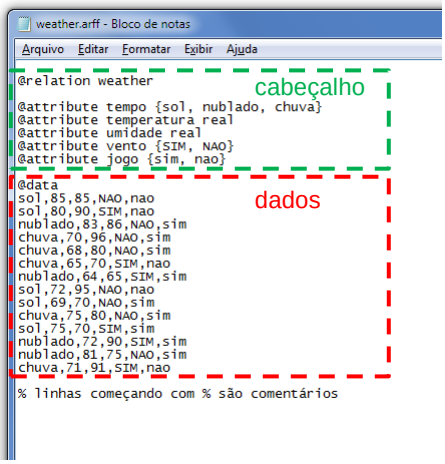
5

- Open File...
  - Abre uma caixa de diálogo que permite que você navegue para os dados arquivo no sistema de arquivos local
  - Opção padrão: arquivos no formato **ARFF**
    - ARFF: Attribute-Relation File Format

Arquivo ARFF

6

- O que é?
  - O formato ARFF é utilizado como padrão para estruturar as bases de dados manipuladas pela Weka
  - É um arquivo de texto ASCII que descreve uma lista de instâncias que compartilham um conjunto de atributos



7

Weka Explorer - Preprocess

Weka Explorer: Preprocess

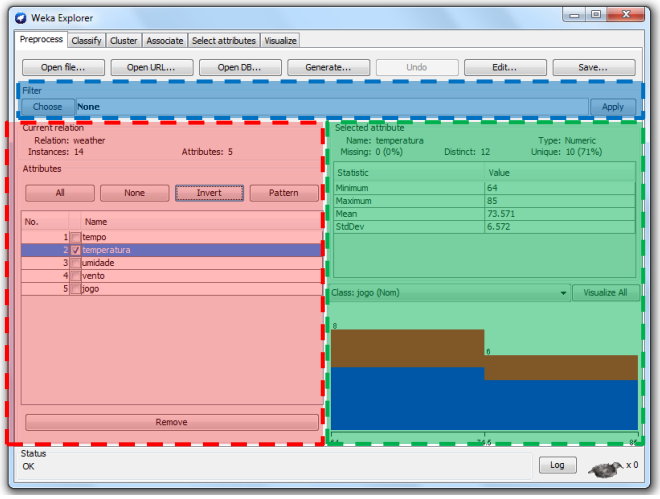
8

- Módulo que permite escolher os dados a serem utilizados. Permite também que se modifique esses dados por meio da aplicação de filtros
- Nele podemos
  - Selecionar conjuntos de dados em diversos formatos
  - Excluir atributos
  - Acessar estatísticas básicas
  - Aplicar um filtro aos dados
    - Ex.: zscore

Weka Explorer: Preprocess

9

- Visão geral



Filtros

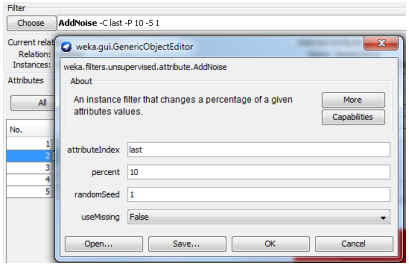
Atributos existentes

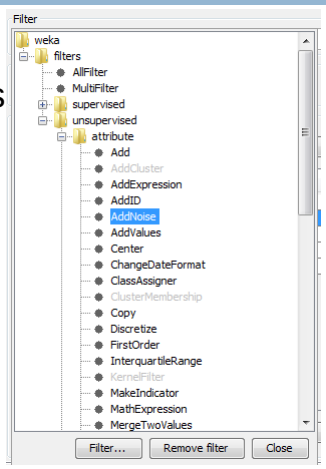
Estatísticas do atributo selecionado

Weka Explorer: Preprocess

10

- Filtros
  - Permitem transformar os dados de várias maneiras
    - Ex: Adicionar ruído
  - Clicando no nome do filtro, podemos configurá-lo

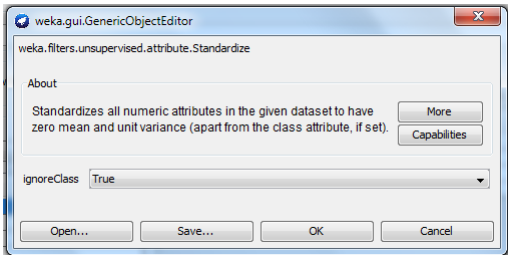




Weka Explorer: Preprocess

11

- zscore
  - normaliza os dados
  - Selecione
    - weka.filters.unsupervised.attribute. Standardize
  - Clicando no nome do filtro, podemos configurá-lo de modo a não normalizar a classe dos dados



12

Weka Explorer - Classify

Weka Explorer: Classify

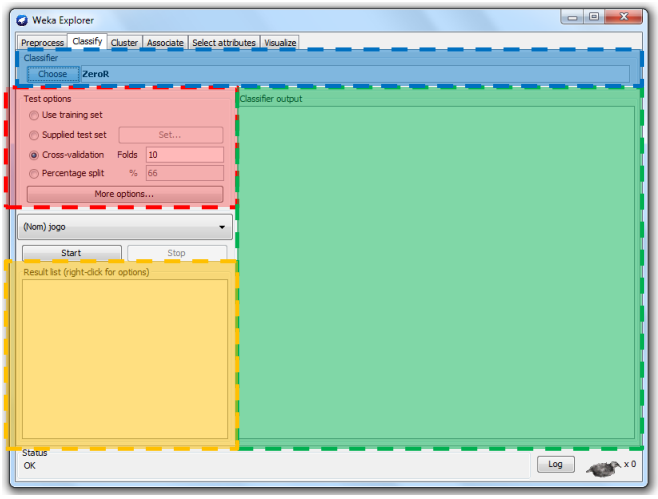
13

- Módulo que permite treinar e testar sistemas de aprendizagem que classificam ou realizar uma regressão dos dados selecionados em **Preprocess**
- Nele podemos
  - Selecionar e configurar diversos classificadores
  - Escolher a metodologia de teste
    - Fornecer arquivo de teste
    - Realizar *cross-validation*
    - Etc.

Weka Explorer: Classify

14

- Visão Geral



Classificadores

Metodologia de teste

Resultados do classificador

Últimos testes

Weka Explorer: Classify

15

- Metodologia de teste
  - Use training set
    - Usa os casos de treino como de teste
  - Supplied test set
    - Permite selecionar um arquivo com os casos de teste
  - Cross-validation
    - Usa validação cruzada do tipo k-fold
  - Percentage split
    - Usa uma certa porcentagem dos dados para teste

Árvores de Decisão

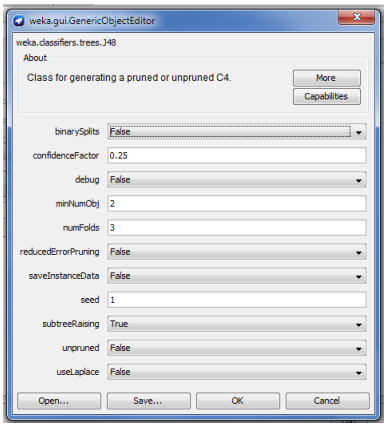
16

- Selecione
  - weka.classifiers.trees
- Algumas árvores disponíveis
  - J48
    - Árvore de decisão C4.5 (com ou sem poda)
  - NBTree (Naive Bayes tree)
    - Árvore de decisão com classificador *naive Bayes nas folhas*
  - Id3
    - Árvore de decisão Id3
  - LMT
    - Árvore de decisão com modelo logistico

Árvores de Decisão

17

- Configurando o classificador
  - Clicando no nome dele, podemos configurá-lo
  - Ex.: árvore J48



Árvores de Decisão

18

- Clicando em “Start” o classificador é executado. Saída:

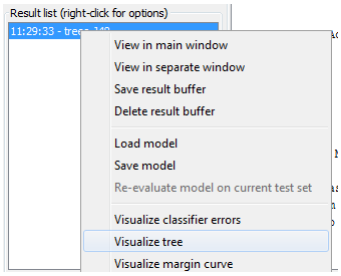
```
J48 pruned tree
------------------

tempo = sol
|   umidade <= 75: sim (2.0)
|   umidade > 75: nao (3.0)
tempo = nublado: sim (4.0)
tempo = chuva
|   vento = SIM: nao (2.0)
|   vento = NAO: sim (3.0)

Number of Leaves  :     5

Size of the tree :      8
```

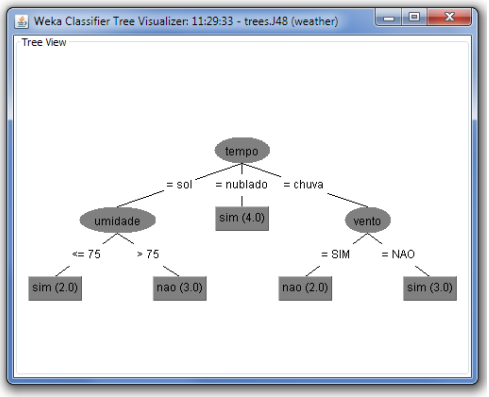
```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances           9               64.2857 %
Incorrectly Classified Instances         5               35.7143 %
Kappa statistic                          0.186
Mean absolute error                      0.2857
Root mean squared error                  0.4818
Relative absolute error                 60      %
Root relative squared error             97.6586 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.778     0.6        0.7       0.778     0.737      0.789    sim
                 0.4       0.222      0.5       0.4       0.444      0.789    nao
Weighted Avg.    0.643     0.465      0.629     0.643     0.632      0.789

=== Confusion Matrix ===

 a b   <-- classified as
 7 2 | a = sim
 3 2 | b = nao
```

Árvores de Decisão

19

- Na lista de resultados, podemos visualizar a árvore gerada





Lazy learning - aprendizado preguiçoso

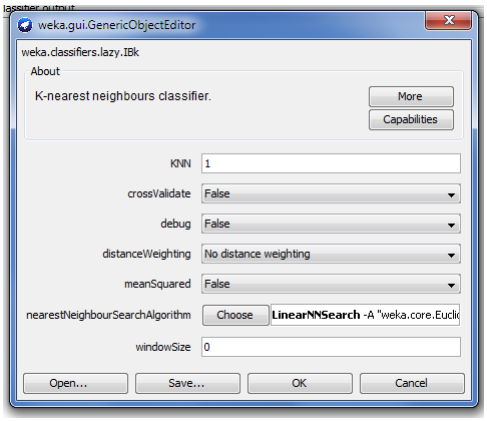
20

- Selecione
  - weka.classifiers.lazy
- Alguns métodos disponíveis
  - IBk
    - K-NN
  - IBk
    - K-NN usando K = 1
  - KStar
    - K-NN com distância com entropia

Lazy learning - aprendizado preguiçoso

21

- Configurando o classificador
  - Clicando no nome dele, podemos configurá-lo
  - Ex.: IBk (K-NN)



Bayeslearning - aprendizado preguiçoso

22

- Clicando em “Start” o classificador é executado. Saída:

```
IB1 instance-based classifier
using 3 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         143               95.3333 %
Incorrectly Classified Instances         7                4.6667 %
Kappa statistic                          0.93
Mean absolute error                      0.04
Root mean squared error                  0.1703
Relative absolute error                  9.0013 %
Root relative squared error             36.1192 %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        Iris-setosa
                 0.94      0.04       0.922     0.94      0.931      0.963    Iris-versicolor
                 0.92      0.03       0.939     0.92      0.929      0.958    Iris-virginica
Weighted Avg.    0.953     0.023      0.953     0.953     0.953      0.974

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 47  3 |  b = Iris-versicolor
  0  4 46 |  c = Iris-virginica
```

Classificadores Bayesianos

23

- Selecione
  - weka.classifiers.bayes
- Alguns métodos disponíveis
  - NaiveBayesSimple
    - Classificação com naive Bayes. Atributos numéricos são modelados por uma distribuição normal
  - NaiveBayes
    - Classificação com naive Bayes utilizando as probabilidades das classes. Permite utilizar estimadores de densidade de kernel no caso dos dados não seguirem a distribuição normal.

Classificadores Bayesianos

24

- Alguns métodos disponíveis
  - NaiveBayesMultinomial
    - Utilizado para classificação de dados de texto (contagem de palavras, etc). Utiliza **distribuição multinomial**.
  - ComplementNaiveBayes
    - Classificação com naive Bayes usando **complemento de classe**: para estimar os parâmetros de uma classe o método utiliza os dados de todas as classes, menos os da classe a ser treinada.

Classificadores Bayesianos

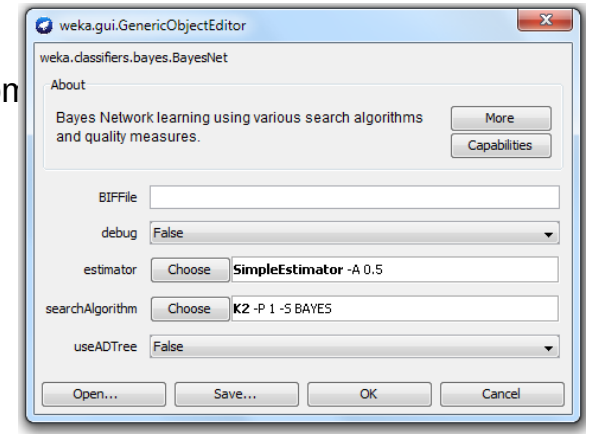
25

- Alguns métodos disponíveis
  - NaiveBayesUpdateable
    - Versão **incremental** do NaiveBayes.
    - Utiliza um método de kernel para calcular a função de probabilidade de variáveis continuas.
  - BayesNet
    - Classificação usando Redes Bayesianas.

Classificadores Bayesianos

26

- Configurando o classificador
  - Clicando no nome dele, podemos configurá-lo
  - Ex.: BayesNet



Classificadores Bayesianos

27

- Clicando em “Start” o classificador é executado. Saída (NaiveBayesSimple)
  - Estatísticas por classe

```
Naive Bayes (simple)

Class Iris-setosa: P(C) = 0.33333333

Attribute sepallength
Mean: 5.006             Standard Deviation: 0.35248969

Attribute sepalwidth
Mean: 3.418             Standard Deviation: 0.3810244

Attribute petallength
Mean: 1.464             Standard Deviation: 0.17351116

Attribute petalwidth
Mean: 0.244             Standard Deviation: 0.1072095
```

```
Class Iris-virginica: P(C) = 0.33333333

Attribute sepallength
Mean: 6.588             Standard Deviation: 0.63587959

Attribute sepalwidth
Mean: 2.974             Standard Deviation: 0.32249664

Attribute petallength
Mean: 5.552             Standard Deviation: 0.5518947

Attribute petalwidth
Mean: 2.026             Standard Deviation: 0.27465006
```

```
Class Iris-versicolor: P(C) = 0.33333333

Attribute sepallength
Mean: 5.936             Standard Deviation: 0.51617115

Attribute sepalwidth
Mean: 2.77              Standard Deviation: 0.31379832

Attribute petallength
Mean: 4.26              Standard Deviation: 0.46991098

Attribute petalwidth
Mean: 1.326             Standard Deviation: 0.19775268
```

Classificadores Bayesianos

28

- Clicando em “Start” o classificador é executado. Saída (NaiveBayesSimple)
  - Resultado da classificação

```
Correctly Classified Instances         143               95.3333 %
Incorrectly Classified Instances         7                4.6667 %
Kappa statistic                          0.93
Mean absolute error                      0.0375
Root mean squared error                  0.1541
Relative absolute error                  8.4391 %
Root relative squared error             32.6809 %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        Iris-setosa
                 0.94      0.04       0.922     0.94      0.931      0.991    Iris-versicolor
                 0.92      0.03       0.939     0.92      0.929      0.991    Iris-virginica
Weighted Avg.    0.953     0.023      0.953     0.953     0.953      0.994

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 47  3 |  b = Iris-versicolor
  0  4 46 |  c = Iris-virginica
```

Regressão

29

- Selecione
  - weka.classifiers.functions
- Métodos de regressão disponíveis
  - SimpleLinearRegression
    - Modelo de regressão linear simples
    - Escolhe o atributo que resulta no menor erro quadrado
    - Os valores em falta não são permitidos
    - Trabalha apenas com atributos numéricos

Regressão

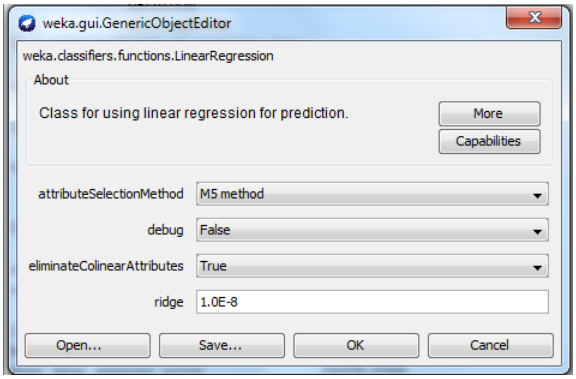
30

- Métodos de regressão disponíveis
  - LinearRegression
    - Funciona como o SimpleLinearRegression
    - Usa o critério de Akaike (medida da qualidade relativa) para seleção do modelo de regressão (linear ou múltipla)
    - É capaz de lidar com casos ponderados

Regressão

31

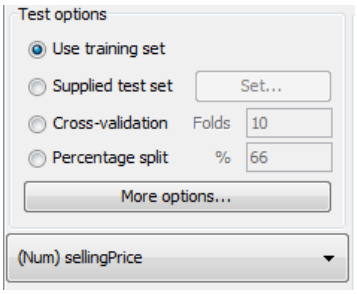
- Configurando o classificador
  - Clicando no nome dele, podemos configurá-lo
  - Ex.: LinearRegression



Regressão

32

- Devemos usar sempre “Use training set” em “Test options”
  - A regressão será calculada em cima dos dados de treinamento
- Definir variável dependente
  - Aquela que os dados irão predizer
    - Ex: sellingPrice



Regressão

33

- Clicando em “Start” o classificador é executado.

LinearRegression

```
Linear Regression Model

sellingPrice =

    -26.6882 * houseSize +
      7.0551 * lotSize +
  43166.0767 * bedrooms +
  42292.0901 * bathroom +
 -21661.1208

Time taken to build model: 0.06 seconds

=== Evaluation on training set ===
=== Summary ===

Correlation coefficient                  0.9945
Mean absolute error                   4053.821
Root mean squared error               4578.4125
Relative absolute error                 13.1339 %
Root relative squared error             10.51   %
Total Number of Instances               7
```

SimpleLinearRegression

```
Linear regression on lotSize

9.24 * lotSize + 114992.8


Time taken to build model: 0 seconds

=== Evaluation on training set ===
=== Summary ===

Correlation coefficient                  0.7859
Mean absolute error                  21597.2469
Root mean squared error              26939.2269
Relative absolute error                 69.9726 %
Root relative squared error             61.8405 %
Total Number of Instances               7
```

Redes Neurais

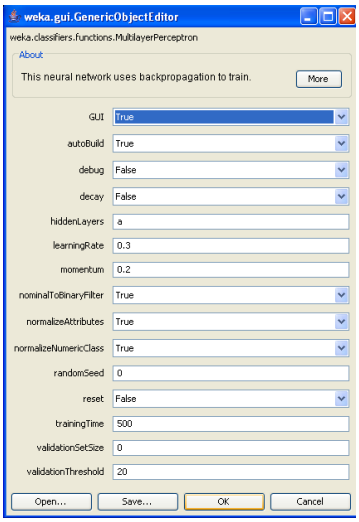
34

- Selecione
  - weka.classifiers.functions
- O único método disponível será
  - MultiLayerPreceptron
- Apesar de possuir apenas essa rede, é possível encontrar pacotes com outras redes implementadas na internet
  - Self-Organizing Maps
  - Learning Vector Quantizer
  - Elman Recurrent Network
  - etc

Redes Neurais

35

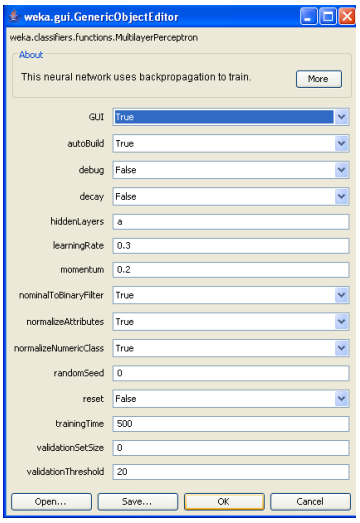
- Configurando o classificador
  - training time
    - Nro de iterações
  - learning rate
    - Incremento do ajuste de pesos no back propogation
  - momentum
    - Controla as mudanças nas variações dos incrementos



Redes Neurais

36

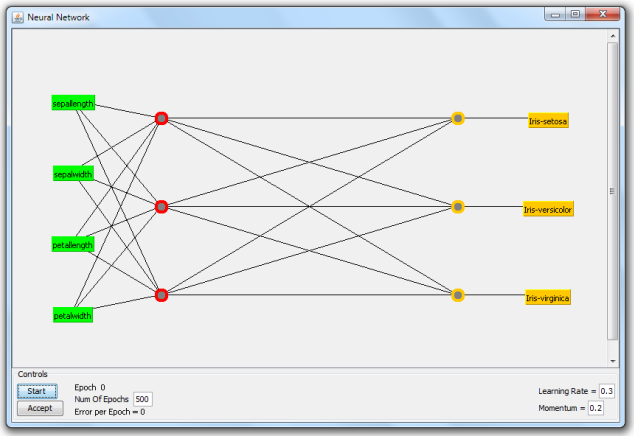
- Configurando o classificador
  - hiddenLayers
    - Nro de camadas ocultas. O valor 0 indica que não possui camadas ocultas
    - Existem também alguns curingas que definem automaticamente o nro de camadas
      - 'a' = (número de atributos + número de classes) / 2
      - 'i' = número de atributos
      - 'o' = número de classes
      - 't' = número de atributos + número de classes.



Redes Neurais

37

- Configurando o classificador
  - GUI: Exibe a rede gerada



Redes Neurais

38

- Clicando em “Start” o classificador é executado.

```
Time taken to build model: 1.64 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         146               97.3333 %
Incorrectly Classified Instances         4                2.6667 %
Kappa statistic                          0.96
Mean absolute error                      0.0327
Root mean squared error                  0.1291
Relative absolute error                  7.3555 %
Root relative squared error             27.3796 %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        Iris-setosa
                 0.96      0.02       0.96      0.96      0.96       0.996    Iris-versicolor
                 0.96      0.02       0.96      0.96      0.96       0.996    Iris-virginica
Weighted Avg.    0.973     0.013      0.973     0.973     0.973      0.998

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 48  2 |  b = Iris-versicolor
  0  2 48 |  c = Iris-virginica
```

SVM

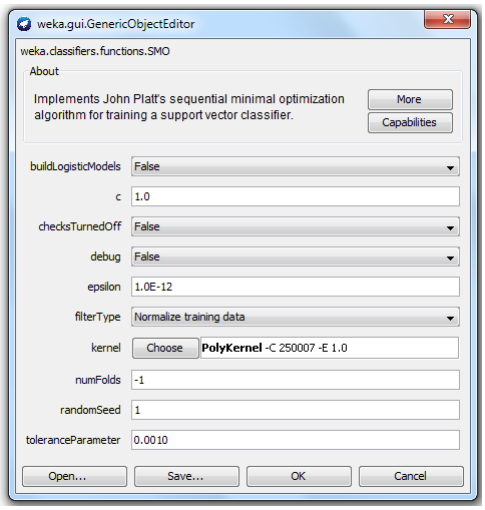
39

- Selecione
  - weka.classifiers.functions
- Método disponíveis
  - SMO
    - Implementa o algoritmo de otimização mínima sequencial de John Platt para treinar uma SVM
  - LibSVM
    - Pacote com implementações mais robustas e eficientes de diferentes SVM

SVM

40

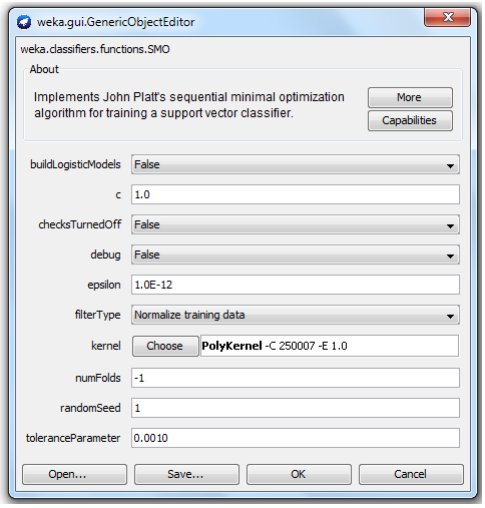
- Configurando SMO
  - filterType
    - Determina como/se os dados serão transformados
  - Kernel
    - Define o kernel a ser usado



SVM

41

- Configurando SMO
  - numFolds
    - Número de folds da validação cruzada
    - -1 significa que os dados de treinamento serão usados
  - Não modificar
    - toleranceParameter
    - epsilon
    - checksTurnedOff



SVM

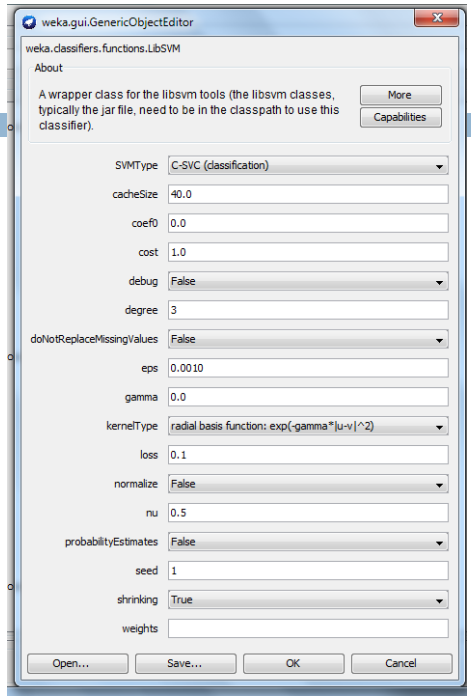
42

- LibSVM
  - Características
    - Diferentes formulações SVM
    - Classificação multi-classes mais eficiente
    - Validação cruzada para seleção de modelos
    - Estimativas de probabilidade
    - Vários kernels (incluindo matriz de kernel pre-calculado)
    - SVM ponderada para dados desbalanceados

SVM

43

- Configurando o libSVM
  - svm_type
    - Seleciona o tipo de SVM
  - kernel_type
    - Seleciona a função kernel
  - Demais parâmetros
    - Funcionamento semelhante ao do SMO…
    - … ou sua configuração depende do tipo de SVM usada



SVM

44

- Clicando em “Start” o classificador é

```
SMO

Kernel used:
  Linear Kernel: K(x,y) = <x,y>

Classifier for classes: Iris-setosa, Iris-versicolor

BinarySMO

Machine linear: showing attribute weights, not support vectors.

        0.6829 * (normalized) sepallength
 +     -1.523  * (normalized) sepalwidth
 +      2.2034 * (normalized) petallength
 +      1.9272 * (normalized) petalwidth
 -      0.7091

Number of kernel evaluations: 352 (70.32% cached)

Classifier for classes: Iris-setosa, Iris-virginica

BinarySMO

Machine linear: showing attribute weights, not support vectors.

        0.5886 * (normalized) sepallength
 +     -0.5782 * (normalized) sepalwidth
 +      1.6429 * (normalized) petallength
 +      1.4777 * (normalized) petalwidth
 -      1.1668

Number of kernel evaluations: 284 (68.996% cached)

Classifier for classes: Iris-versicolor, Iris-virginica

BinarySMO

Machine linear: showing attribute weights, not support vectors.

        0.3176 * (normalized) sepallength
 +     -0.863  * (normalized) sepalwidth
 +      3.0543 * (normalized) petallength
 +      4.0815 * (normalized) petalwidth
 -      4.5924

Number of kernel evaluations: 453 (61.381% cached)
```

SVM

45

- Clicando em “Start” o classificador é executado

```
Time taken to build model: 0.03 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         144               96      %
Incorrectly Classified Instances         6                4      %
Kappa statistic                          0.94
Mean absolute error                      0.2311
Root mean squared error                  0.288
Relative absolute error                 52      %
Root relative squared error             61.101   %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        Iris-setosa
                 0.98      0.05       0.907     0.98      0.942      0.965    Iris-versicolor
                 0.9       0.01       0.978     0.9       0.938      0.97     Iris-virginica
Weighted Avg.    0.96      0.02       0.962     0.96      0.96       0.978

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 49  1 |  b = Iris-versicolor
  0  5 45 |  c = Iris-virginica
```

46

Weka Explorer - Cluster

Weka Explorer: Cluster

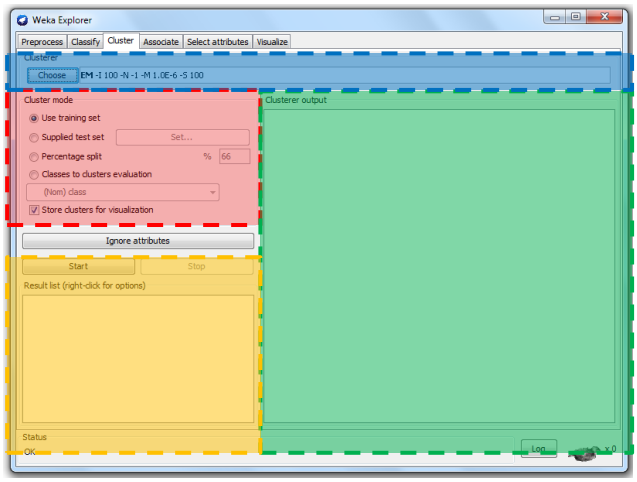
47

- Módulo que permite analisar os ***clusters*** ou ***agrupamentos*** dos dados selecionados em **Preprocess**
- Nele podemos
  - Selecionar e configurar diversos métodos de agrupamentos
  - Escolher a metodologia de avaliação do agrupamento
    - Os próprios dados
    - Fornecer arquivo de teste
    - Etc.

Weka Explorer: Cluster

48

- Visão Geral



Weka Explorer: Cluster

49

- Metodologia de avaliação
  - Use training set
    - Classifica os dados de treinamento nos clusters e calcula a percentagem de casos em cada cluster
  - Supplied test set
    - Permite selecionar um arquivo com os casos de teste para avaliar o agrupamento, se este for probabilístico
  - Percentage split
    - Usa uma certa porcentagem dos dados para avaliar o agrupamento, se este for probabilístico

Weka Explorer: Cluster

50

- Metodologia de avaliação
  - Classes to clusters evaluation
    - Ignora a classe e calcula o agrupamento.
    - Atribui classes aos clusters, de acordo com a as amostras dentro do cluster: *classe mais frequente*
    - Em seguida, calcula o erro de classificação e mostra a matriz de confusão correspondente.

Análise de Clusters

51

- Selecione
  - weka.clusterers
- Método disponíveis
  - SimpleKMeans
    - K-means
  - EM
    - *Expectation maximization* ou maximização de expectativa
    - Gera descrições probabilísticas dos clusters em termos de média e desvio padrão para os atributos numéricos

Análise de Clusters

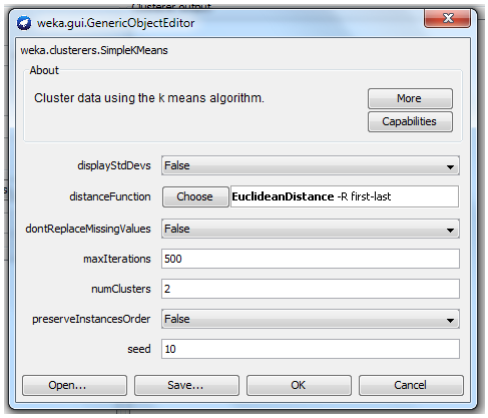
52

- Método disponíveis
  - Cobweb
    - Gera agrupamento hierárquico, onde os grupos são descritos probabilisticamente
  - HierarchicalClusterer
    - Implementa uma série de métodos clássicos hierárquicos e tipos de linkage (Single, Complete, Average, Mean, Centroid, Ward,...)

Análise de Clusters

53

- Configurando o método (Ex.: SimpleKMeans)
  - distanceFunction
    - Função de distância
  - maxIterations
    - Nro de iterações máximas
  - numClusters
    - Nro de clusters
  - Seed
    - Nro de sementes iniciais



Análise de Clusters

54

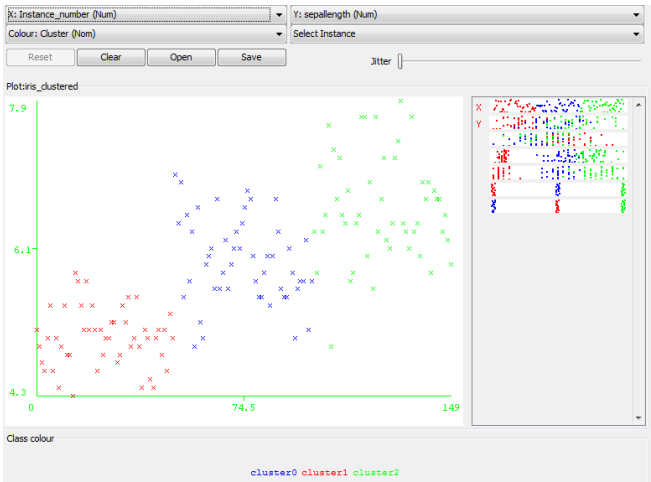
- Clicando em “Start” o método é executado

```
kMeans
======

Number of iterations: 3
Within cluster sum of squared errors: 7.817456892309574
Missing values globally replaced with mean/mode

Cluster centroids:
                                        Cluster#
Attribute               Full Data             0             1             2
                          (150)            (50)          (50)          (50)
=============================================================================
sepallength               5.8433           5.936         5.006         6.588
sepalwidth                 3.054            2.77         3.418         2.974
petallength               3.7587            4.26         1.464         5.552
petalwidth                1.1987           1.326         0.244         2.026
class                Iris-setosa Iris-versicolor    Iris-setosa  Iris-virginica



Time taken to build model (full training data) : 0.04 seconds

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

Clustered Instances

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

Análise de Clusters

55

- Podemos ainda visualizar os clusters formados



56

Weka Explorer – Select Attributes

Weka Explorer: Select Attributes

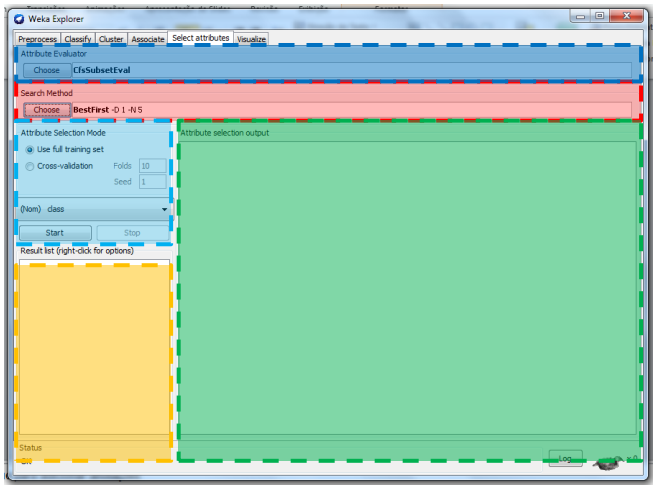
57

- Módulo que permite investigar quais atributos são mais preditivos
  - Seleção em 2 etapas:
    - Um método de busca:
    - Um método de avaliação
  - Flexibilidade: (quase) qualquer combinação de busca/avaliação

Weka Explorer: Select Attributes

58

- Visão Geral



Método de busca

Modo de seleção dos atributos

Últimos testes

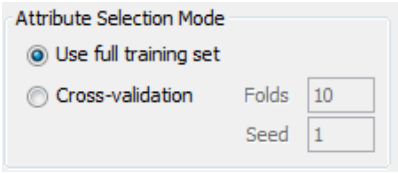
Métodologia de avaliação

Resultados da seleção

Weka Explorer: Select Attributes

59

- Modo de seleção dos atributos
  - Use full training set
    - A importância do atributo é determinada usando todo o conjunto de treinamento
  - Cross-validation
    - A importância do atributo é determinada usando cross-validation



Metodologia de avaliação

60

- Selecione
  - weka.attributeSelection
- Alguns métodos disponíveis
  - CfsSubsetEval
    - Seleciona os atributos medindo sua capacidade preditiva e grau de redundância
  - ChiSquaredAttributeEval
    - Seleciona os atributos atráves do cálculo do qui-quadrado
  - ConsistencySubsetEval
    - Seleciona os atributos medindo o seu nível de consistência

Metodologia de avaliação

61

- Alguns métodos disponíveis
  - GainRatioAttributeEval
    - Seleciona os atributos medindo a taxa de ganho do atributo em relação a classe
  - InfoGainAttributeEval
    - Utiliza o ganho de informação para selecionar os atributos
  - PrincipalComponents
    - Transformação dos dados usando PCA
  - SVMAttributeEval
    - Avalia os atributos usando uma SVM

Método de busca

62

- Selecione
  - weka.attributeSelection
- Alguns métodos disponíveis
  - BestFirst
    - Inicia com nenhum atributo e inclui um atributo por vez no conjunto
  - ExhaustiveSearch
    - Busca exaustiva por todo o conjunto de atributos

Método de busca

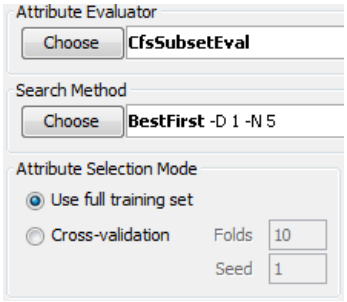
63

- Alguns métodos disponíveis
  - GeneticSearch
    - Realiza a busca utilizando um algoritmo genético simples (Goldberg,1989)
  - Ranker
    - Classifica os atributos usando suas avaliações individuais (e.g., entropia, taxa de ganho etc)

Weka Explorer: Select Attributes

64

- Exemplo: iris.arff
  - Metodologia de avaliação
    - CfsSubsetEval
  - Método de busca
    - BestFirst
  - Modo de seleção dos atributos
    - Use full training set



Weka Explorer: Select Attributes

- Clicando em “Start” o método é executado

```
=== Run information ===

Evaluator:     weka.attributeSelection.CfsSubsetEval
Search:weka.attributeSelection.BestFirst -D 1 -N 5
Relation:      iris
Instances:     150
Attributes:    5
               sepallength
               sepalwidth
               petallength
               petalwidth
               class
Evaluation mode:evaluate on all training data
```

```
=== Attribute Selection on all input data ===

Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 12
        Merit of best subset found:    0.887

Attribute Subset Evaluator (supervised, Class (nominal): 5 class):
        CFS Subset Evaluator
        Including locally predictive attributes

Selected attributes: 3,4 : 2
                     petallength
                     petalwidth
```