Universidade Federal do Paraná (UFPR)
Especialização em Engenharia Industrial 4.0

Introdução ao Weka

Data Mining with Open Source Machine Learning Software in Java

David Menotti
www.inf.ufpr.br/menotti/am-221

Hoje

- Weka
  - Introdução
  - Como instalar
  - Datasets
  - Usando algoritmos de:
    - Classificação
    - Clustering
    - Regressão

Introdução

- Weka é uma coleção de algoritmos de aprendizado de máquina para tarefas de mineração de dados. Ele contém ferramentas para preparação de dados, classificação, regressão, agrupamento, mineração de regras de associação e visualização.

- Encontrada apenas nas ilhas da Nova Zelândia, a **Weka** é uma ave que não voa e tem uma natureza inquisitiva. O nome é pronunciado assim, e o pássaro soa assim.

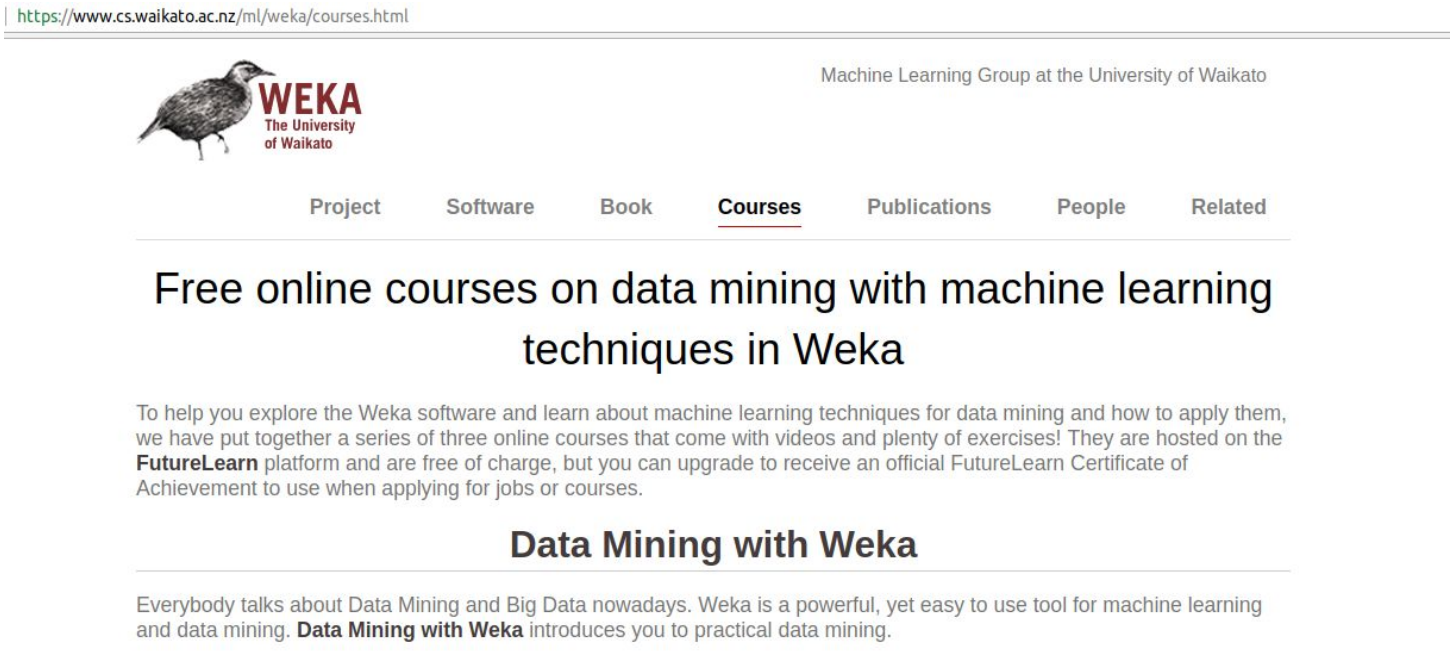
Introdução

- Weka é um software de código aberto emitido sob a ***GNU General Public License***.

- Sim, é possível aplicar a Weka para processar **big data** e realizar aprendizado profundo (**deep learning**)!

Introdução

- No site, existem vários cursos on-line gratuitos que ensinam aprendizado de máquina e mineração de dados usando o Weka. Confira no site os cursos para detalhes sobre quando e como se inscrever. Os vídeos dos cursos estão disponíveis no Youtube.

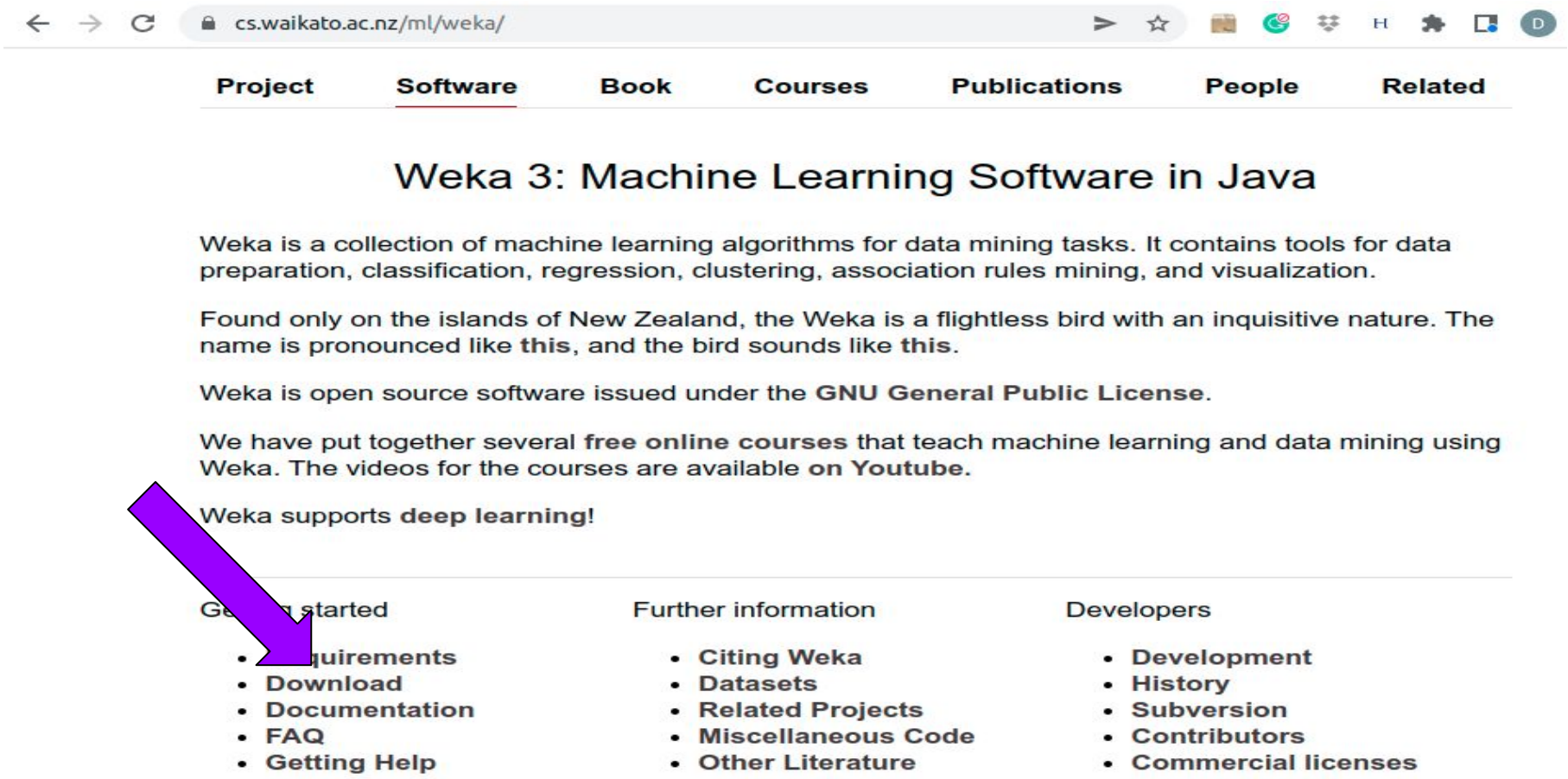


Manual

- Weka Manual
  - (v3-8-1) 19/12/2016
    - https://alvarofpinheiro.eti.br/_files/200000910-c2f11c3ec2/Weka%20Manual-3-8-1.pdf
  - (v3-**7**-8) 21/01/2013
    - https://statweb.stanford.edu/~lpekelis/13_datafest_cart/WekaManual-3-7-8.pdf
  - (v3-**6**-8) 03/05/2012
    - http://www.nilc.icmc.usp.br/elc-ebralc2012/minicursos/WekaManual-3-6-8.pdf

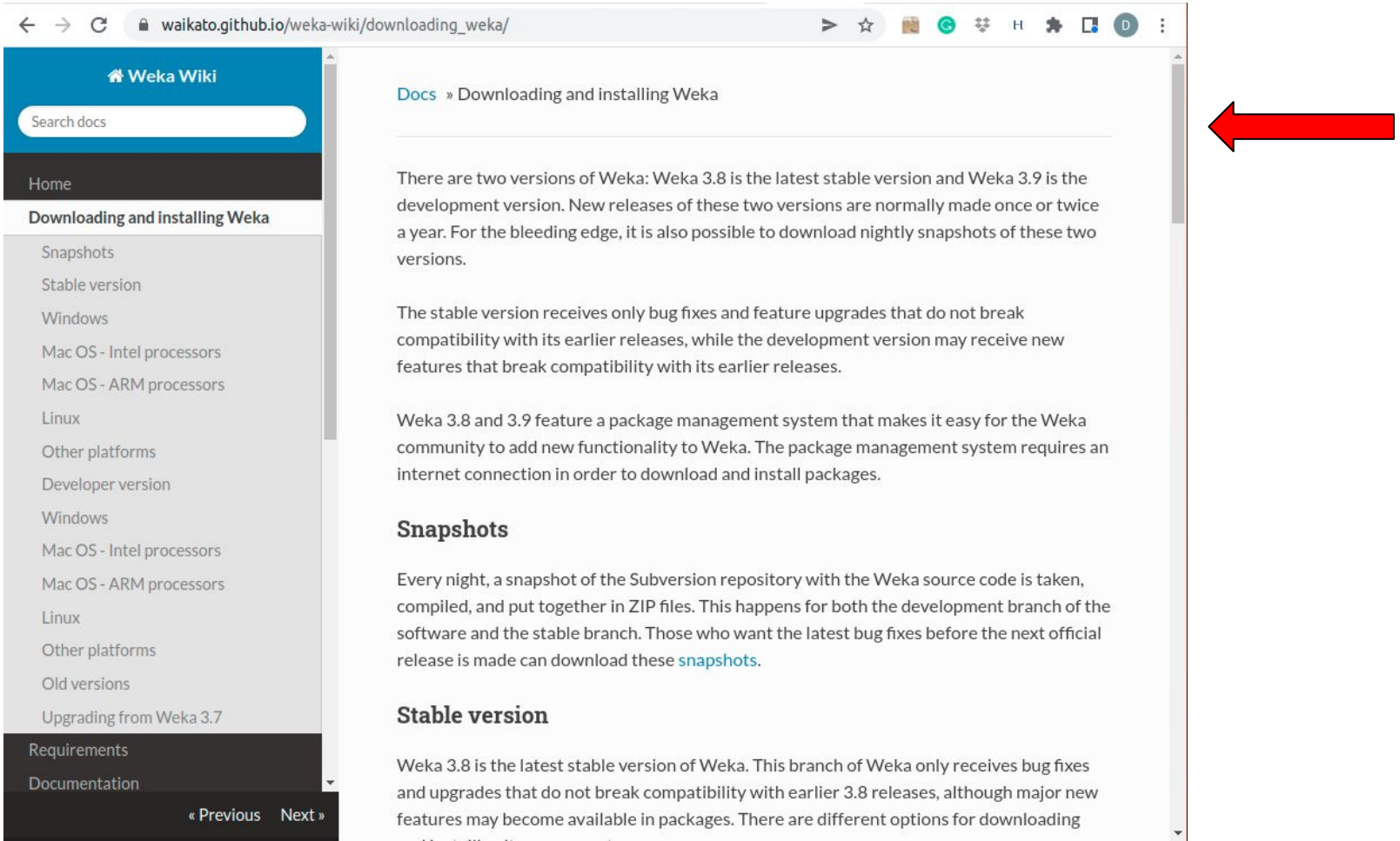
Como Instalar

- Weka website (latest version 3.8/3.9)
  - https://www.cs.waikato.ac.nz/ml/weka/



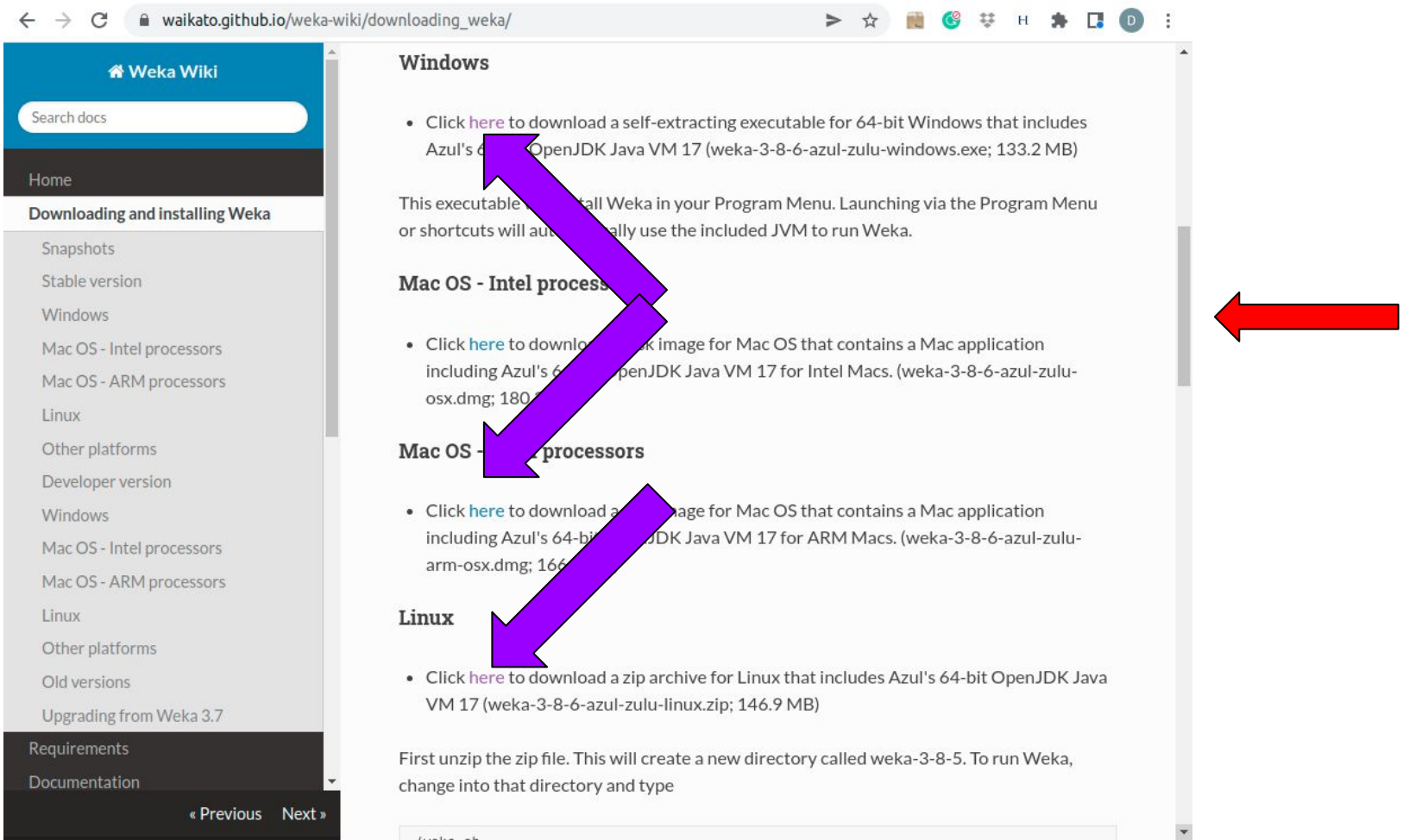
Como Instalar

- Downloading: https://waikato.github.io/weka-wiki/downloading_weka/



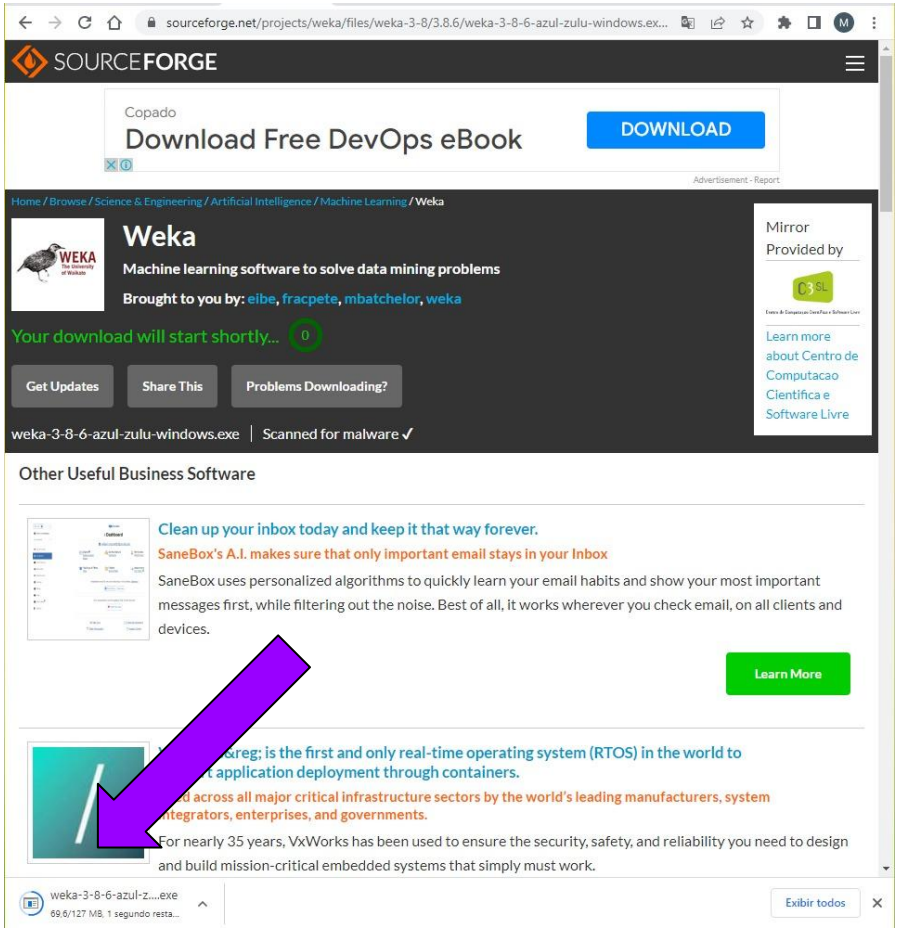
Como Instalar

- Downloading: https://waikato.github.io/weka-wiki/downloading_weka/



Como Instalar

- Downloading: https://waikato.github.io/weka-wiki/downloading_weka/



Como Instalar

- Downloading: https://waikato.github.io/weka-wiki/downloading_weka/

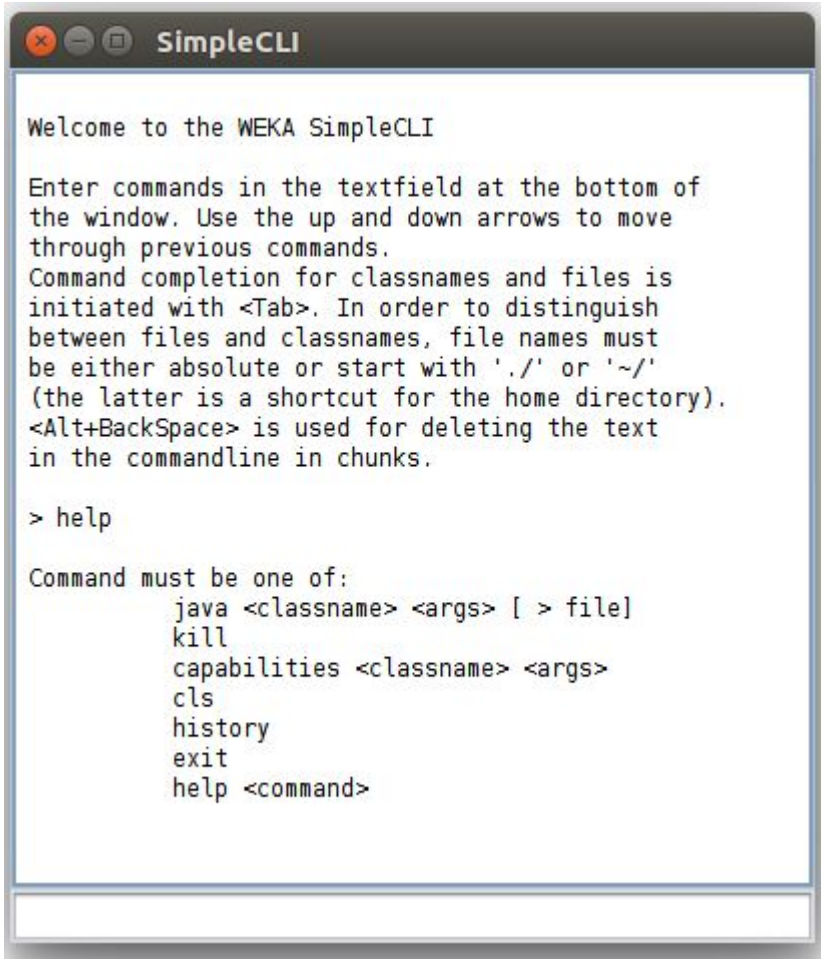
Basta baixar (130Mb) e executar

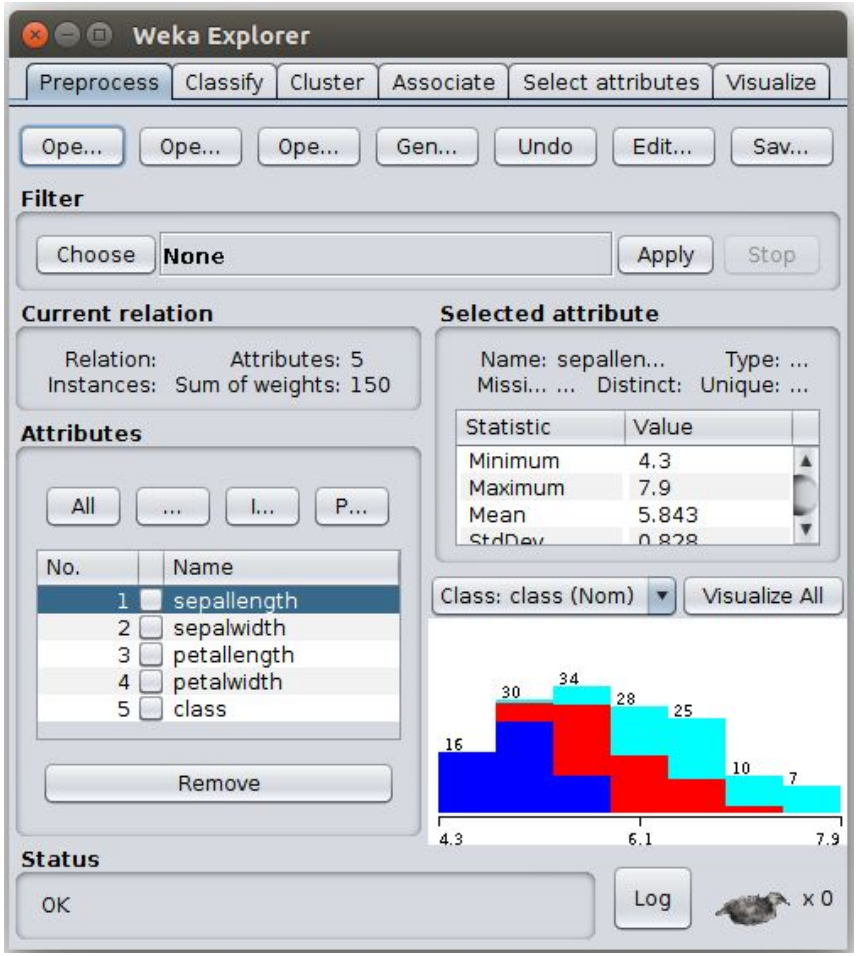
Weka-3-8-6-azul-zulu-windows.exe

CLI vs GUI

- Início

CLI vs GUI


SimpleCLI
Welcome to the WEKA SimpleCLI
Enter commands in the textfield at the bottom of
the window. Use the up and down arrows to move
through previous commands.
Command completion for classnames and files is
initiated with <Tab>. In order to distinguish
between files and classnames, file names must
be either absolute or start with './' or '~/'
(the latter is a shortcut for the home directory).
<Alt+BackSpace> is used for deleting the text
in the commandline in chunks.
> help
Command must be one of:
java <classname> <args> [> file]
kill
capabilities <classname> <args>
cls
history
exit
help <command>


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Ope...
Ope...
Ope...
Gen...
Undo
Edit...
Sav...
Filter
Choose
None
Apply
Stop
Current relation
Relation:
Attributes: 5
Instances:
Sum of weights: 150
Selected attribute
Name: sepallen...
Type: ...
Missi... ...
Distinct:
Unique: ...
Statistic
Value
Minimum
4.3
Maximum
7.9
Mean
5.843
Attributes
All
...
I...
P...
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
Remove
Class: class (Nom)
Visualize All
16
30
34
28
25
10
7
4.3
6.1
7.9
Status
OK
Log
x 0

Atributos

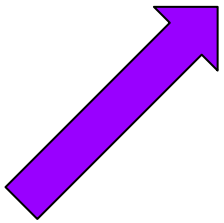
- *Nominal*: um de uma lista predefinida de valores
  - e.g. vermelho, azul, amarelo
- *Numérico*: Um número real ou inteiro
- *String*: delimitada por “aspas duplas”
- *Data*
- *Relational*

Arquivos ARFF

- A representação das instâncias

- Consiste em:
  - Um cabeçalho (header): Descreve os tipos de atributos e seus valores
  - Seção de dados: lista de dados separada por vírgula

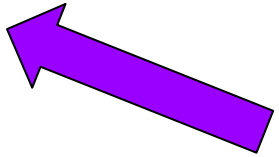
Exemplo de Arquivo ARFF



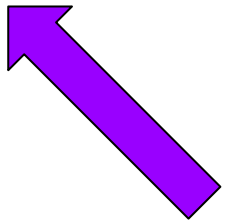
Comment

```
% This is a toy example, the UCI weather dataset
% Any relation to real weather is purely coincidental

@relation weather.symbolic

@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
```

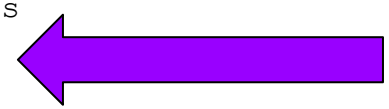
Nome do Dataset



Atributos

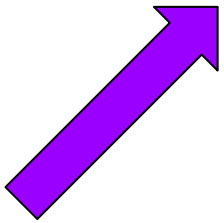


Classe / Meta



Dados/Valores

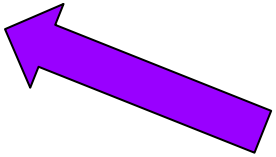
ARFF



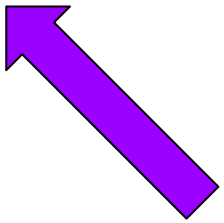
Comment

```
% This is a toy example, the UCI weather dataset
% Any relation to real weather is purely coincidental

@relation weather.symbolic

@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
```

Nome do Dataset



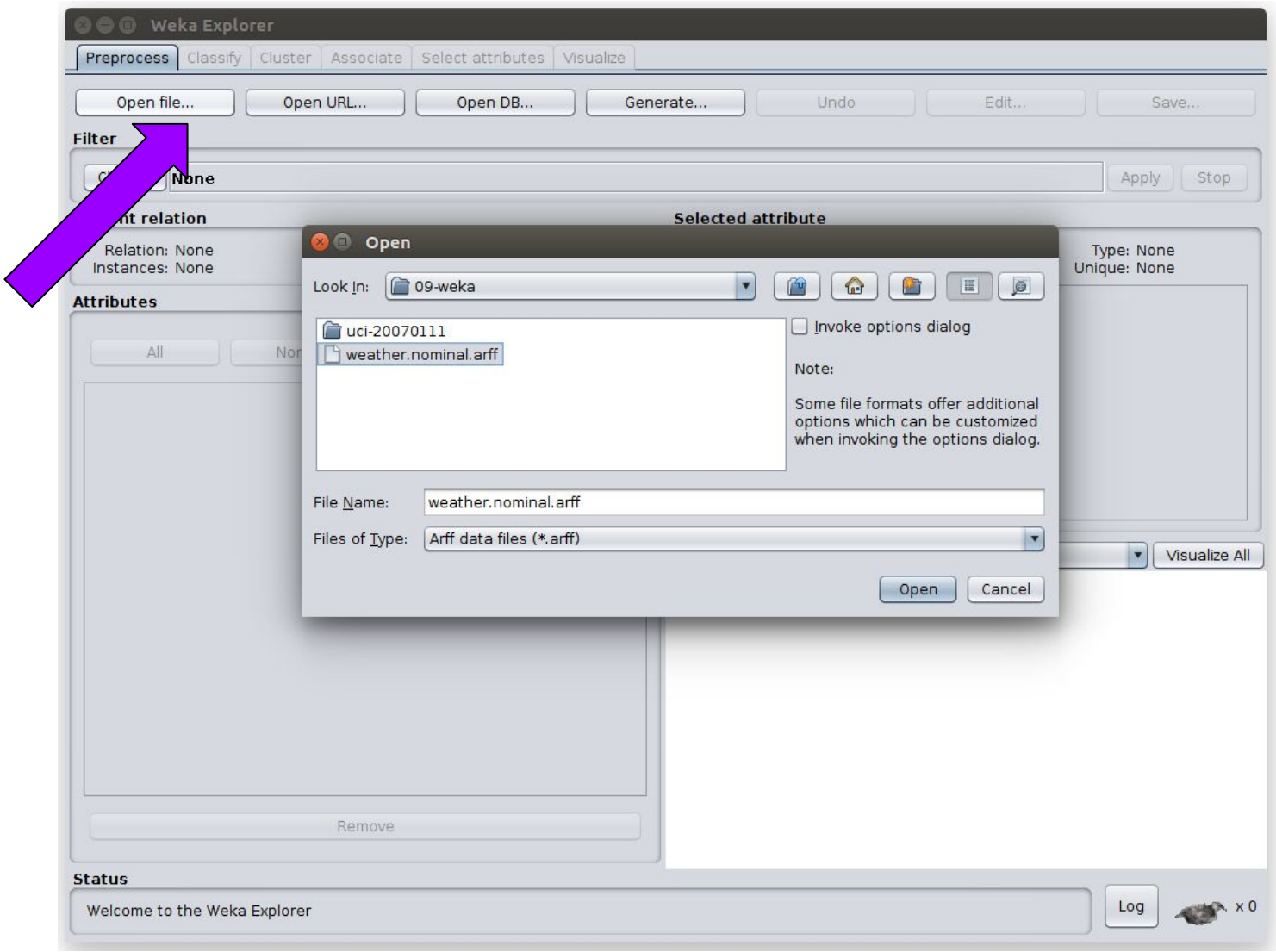
Atributos



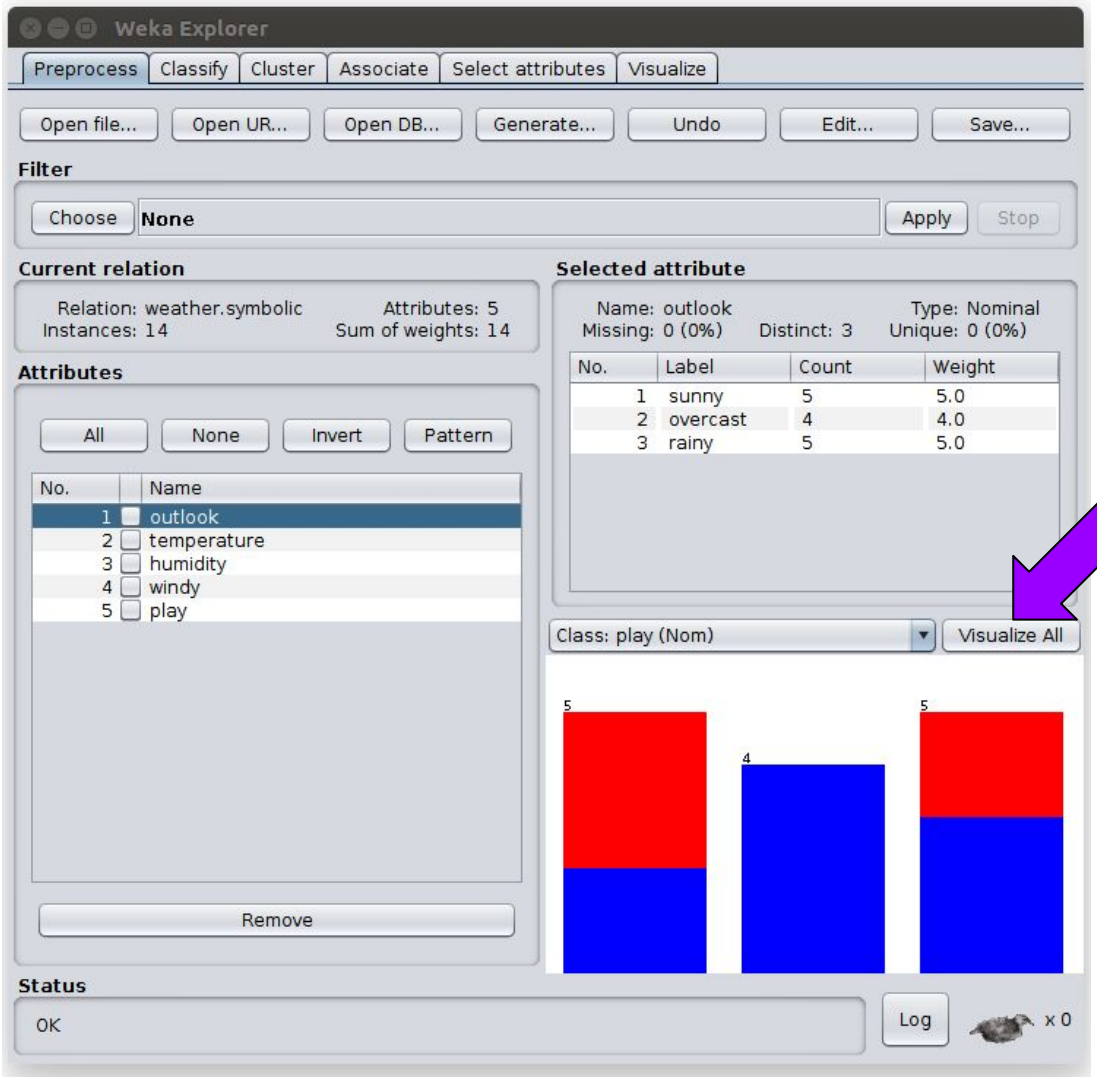
Classe / Meta

Dados/Valores

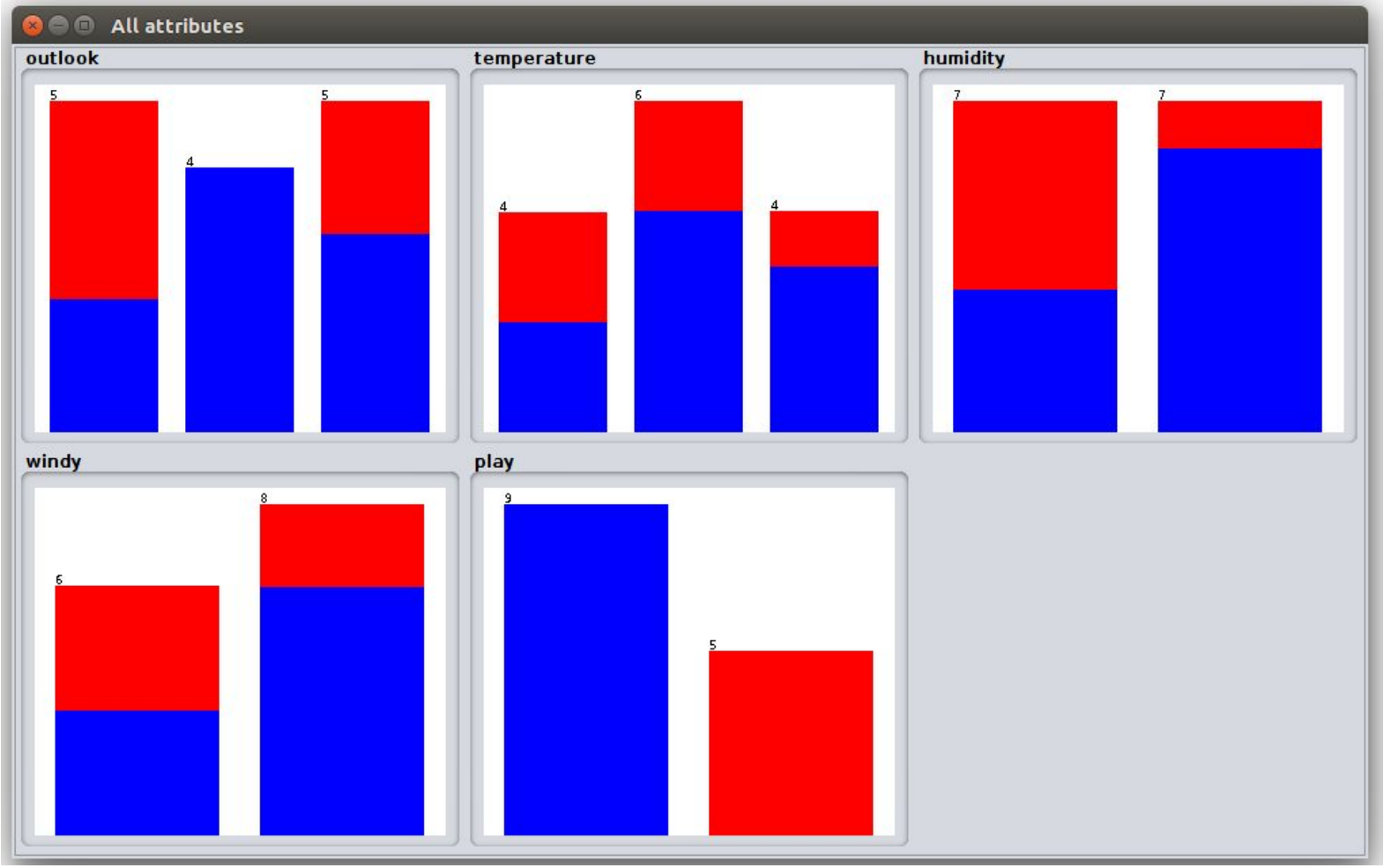
Abrindo um Dataset



Visualizando



Visualizando



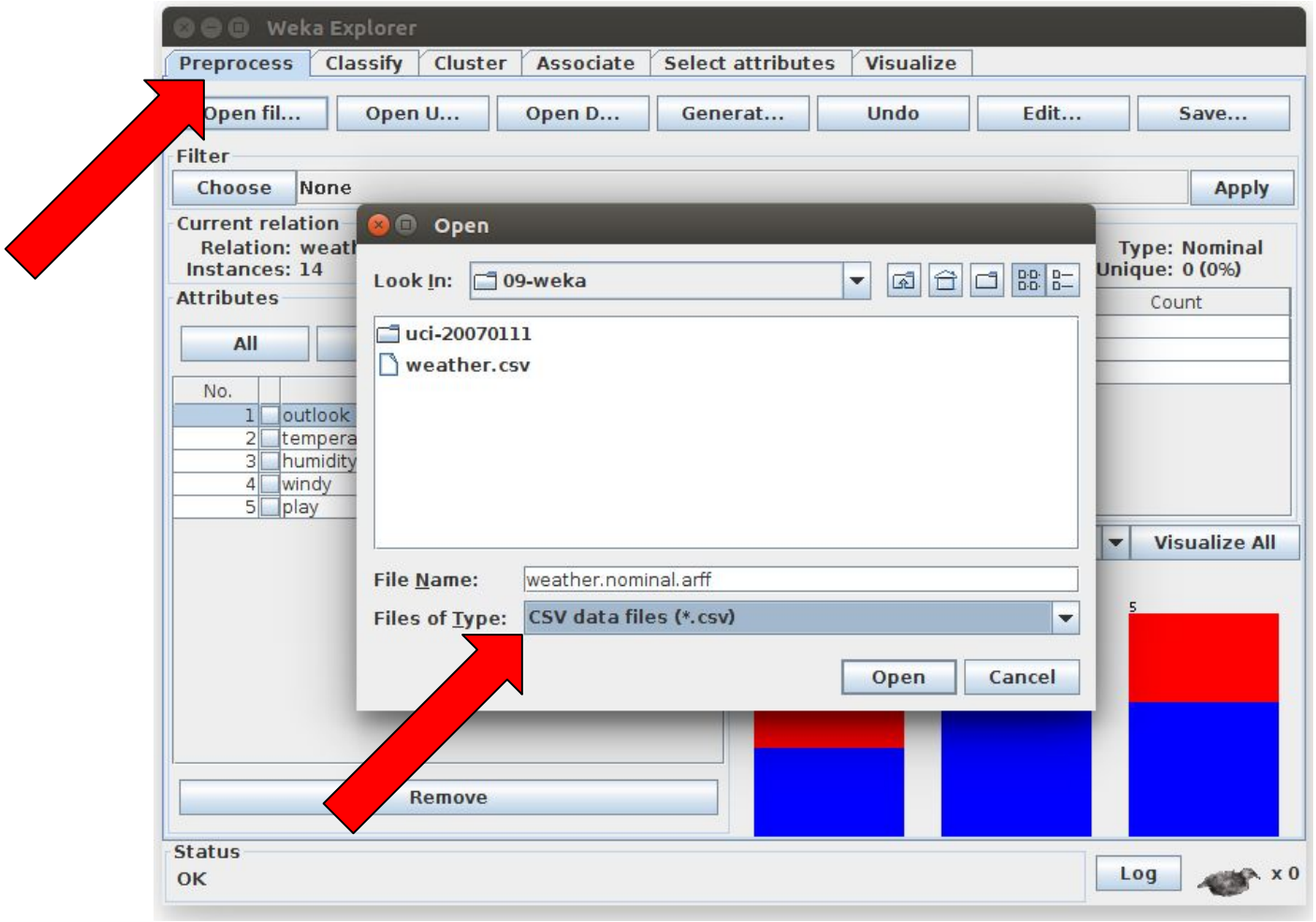
Excel => CSV

	A	B	C	D	E
1	outlook	temperature	humidity	windy	play
2	sunny	hot	high	FALSE	no
3	sunny	hot	high	TRUE	no
4	overcast	hot	high	FALSE	yes
5	rainy	mild	high	FALSE	yes
6	rainy	cool	normal	FALSE	yes
7	rainy	cool	normal	TRUE	no
8	overcast	cool	normal	TRUE	yes
9	sunny	mild	high	FALSE	no
10	sunny	cool	normal	FALSE	yes
11	rainy	mild	normal	FALSE	yes
12	sunny	mild	normal	TRUE	yes
13	overcast	mild	high	TRUE	yes
14	overcast	hot	normal	FALSE	yes
15	rainy	mild	high	TRUE	no

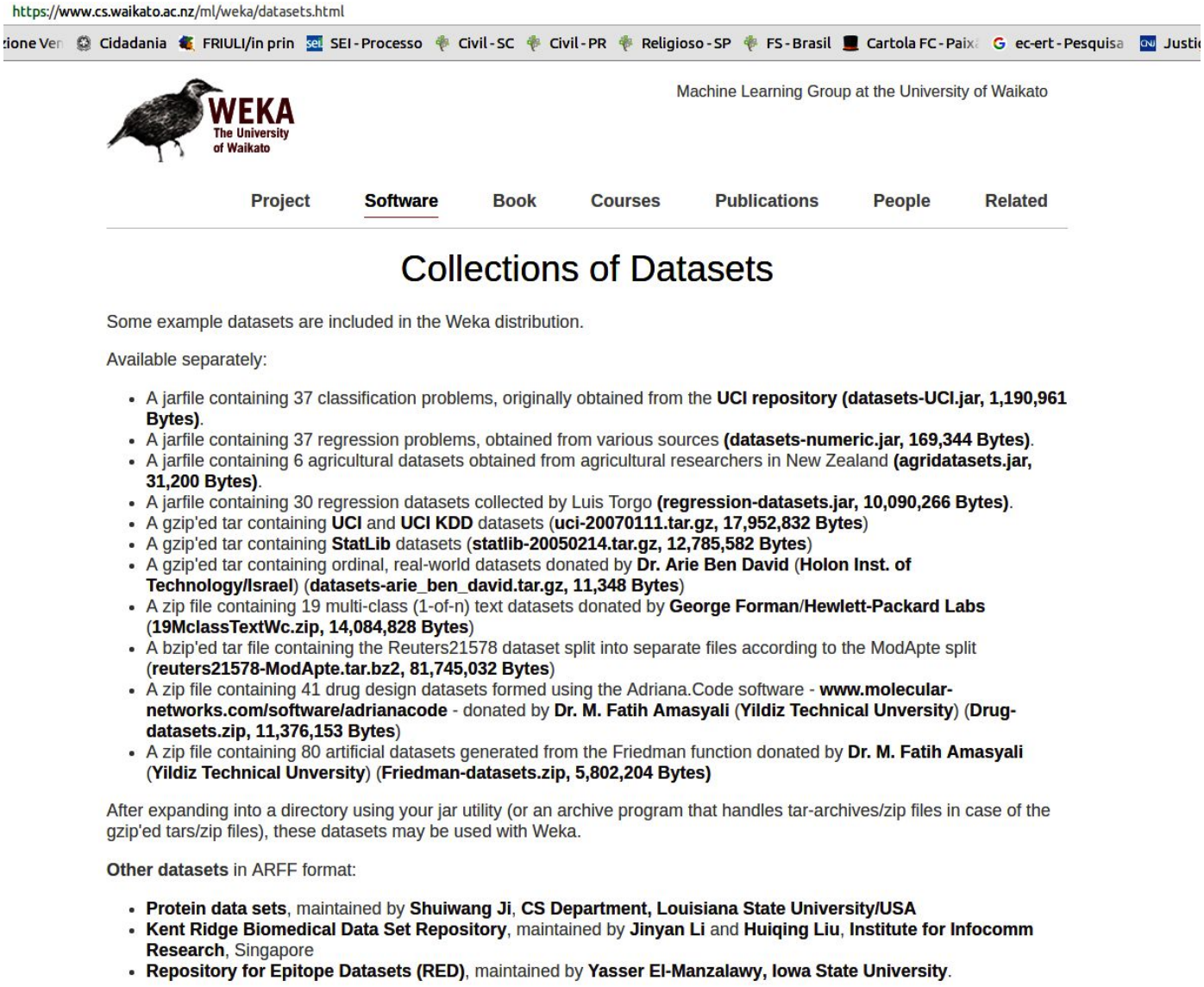
weather.csv

```
outlook,temperature,humidity,windy,play
sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
```

Excel => CSV


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open fil...
Open U...
Open D...
Generat...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Current relation
Relation: weath
Instances: 14
Attributes
All
No.
1 outlook
2 tempera
3 humidity
4 windy
5 play
Type: Nominal
Unique: 0 (0%)
Count
Visualize All
Remove
Status
OK
Log
x 0
Open
Look In:
09-weka
uci-20070111
weather.csv
File Name:
weather.nominal.arff
Files of Type:
CSV data files (*.csv)
Open
Cancel

Software > Datasets



WEKA Datasets

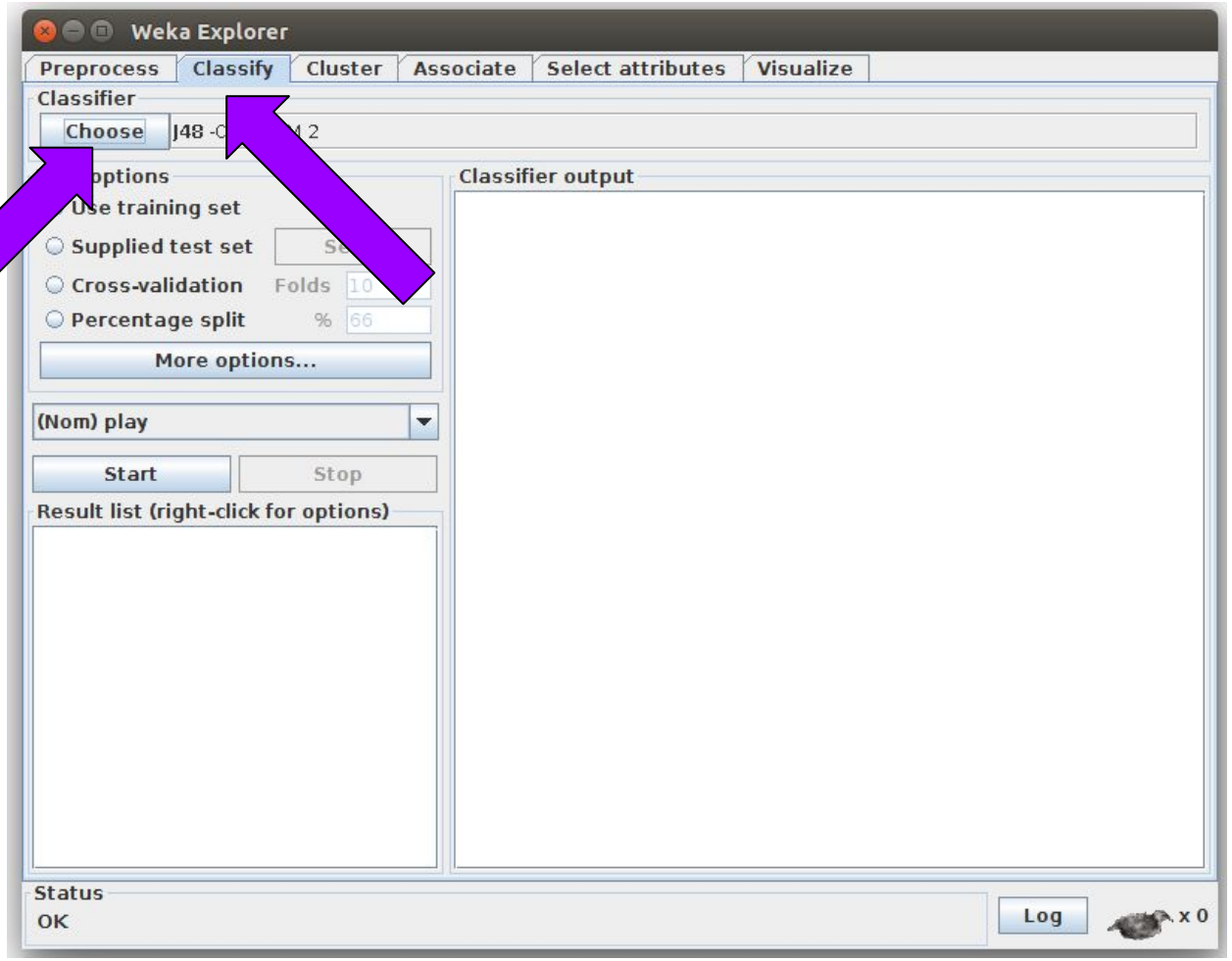
- Alguns datasets em formato **ARFF**
  http://storm.cis.fordham.edu/~gweiss/data-mining/datasets.html

- contact-lens.arff
- cpu.arff
- cpu.with-vendor.arff
- diabetes.arff
- glass.arff
- ionospehre.arff
- iris.arff
- labor.arff
- ReutersCorn-train.arff
- ReutersCorn-test.arff
- ReutersGrain-train.arff
- ReutersGrain-test.arff
- segment-challenge.arff
- segment-test.arff
- soybean.arff
- supermarket.arff
- vote.arff
- weather.arff
- weather.nominal.arff

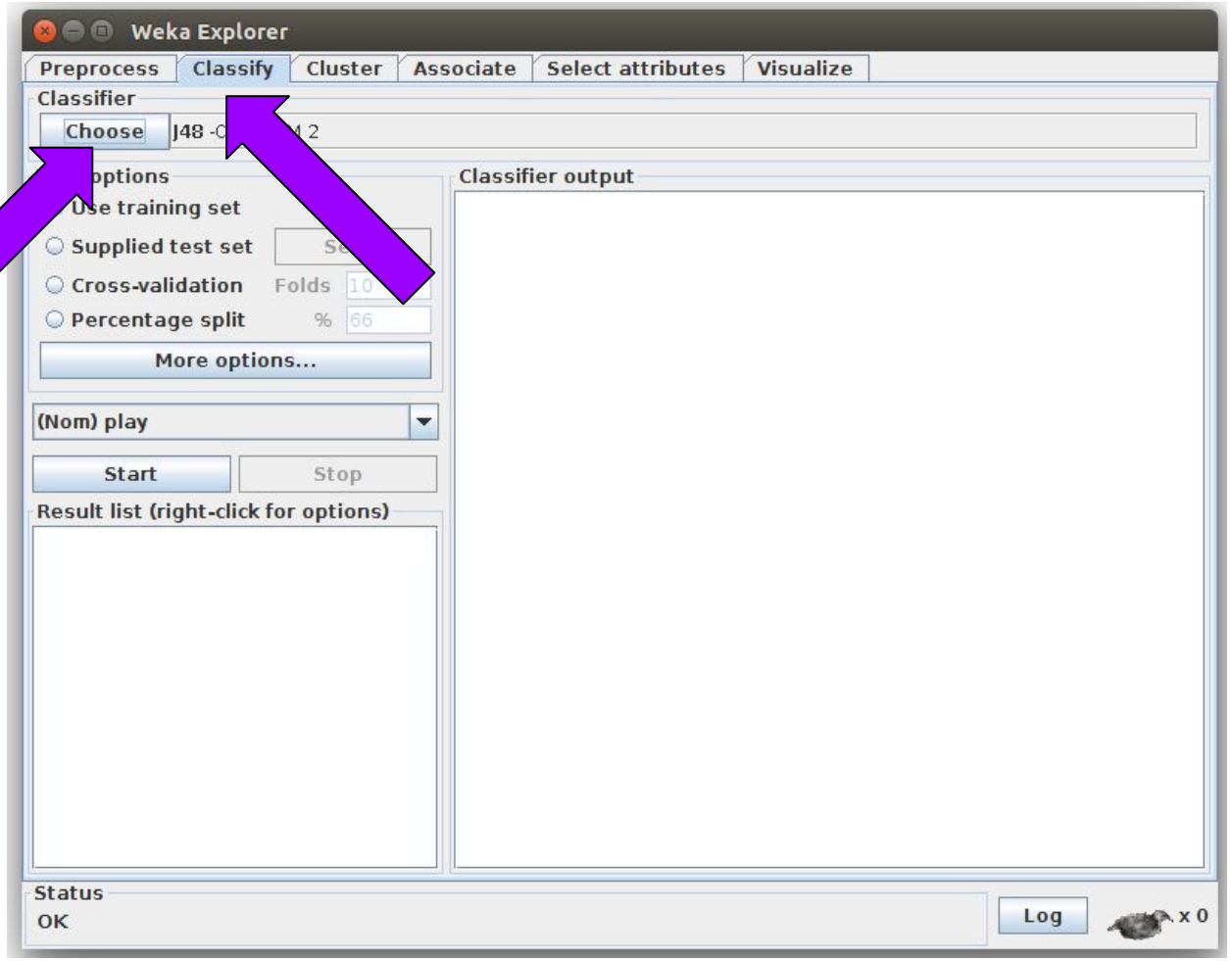
Classificação

- Como gerar:
  - uma árvore de decisão J48
  - um k-NN
  - Naive Bayes classifier
  - MLP
  - SVM
  - PCA

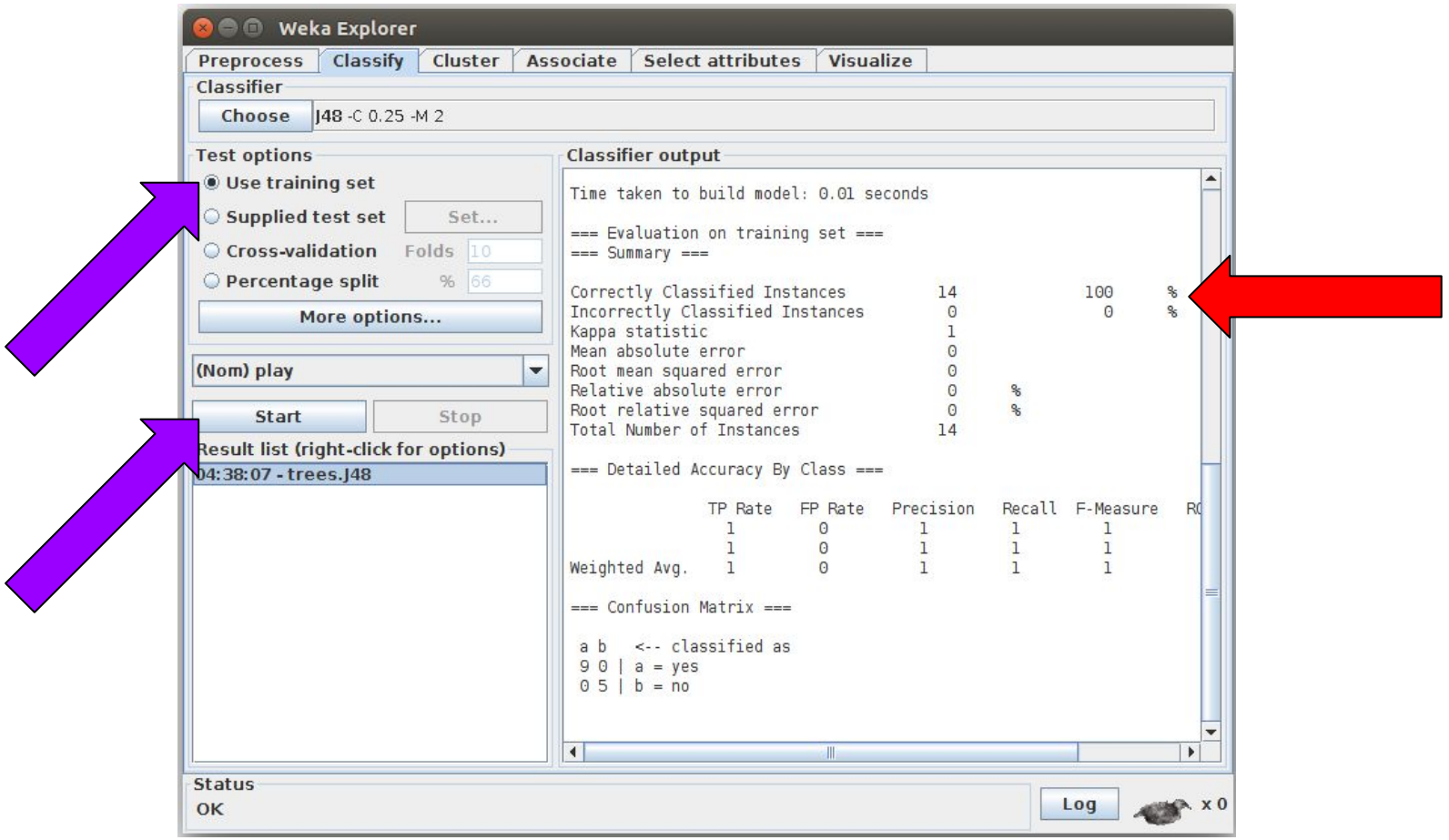
Classify > Choose



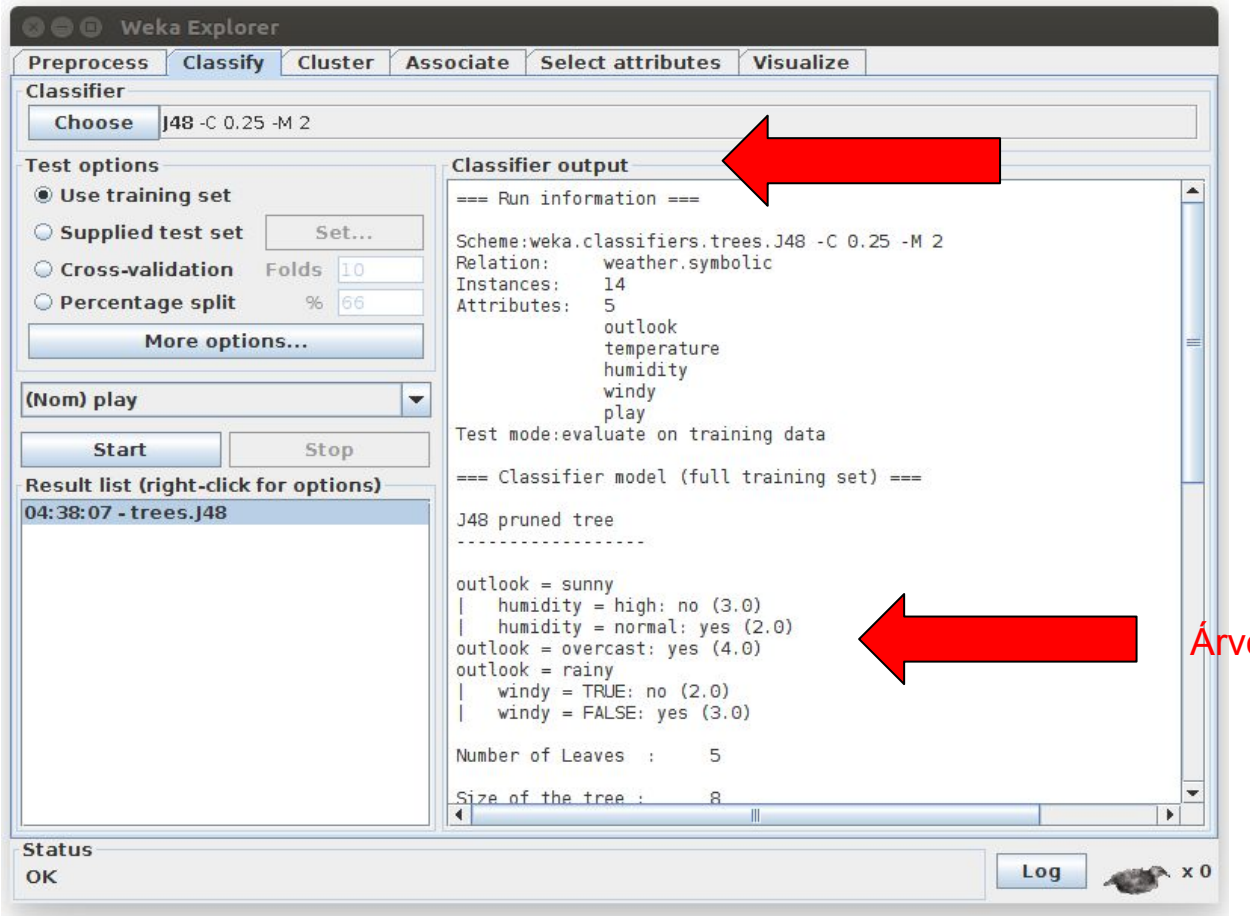
Classify > tree > J48



Classify > tree > J48

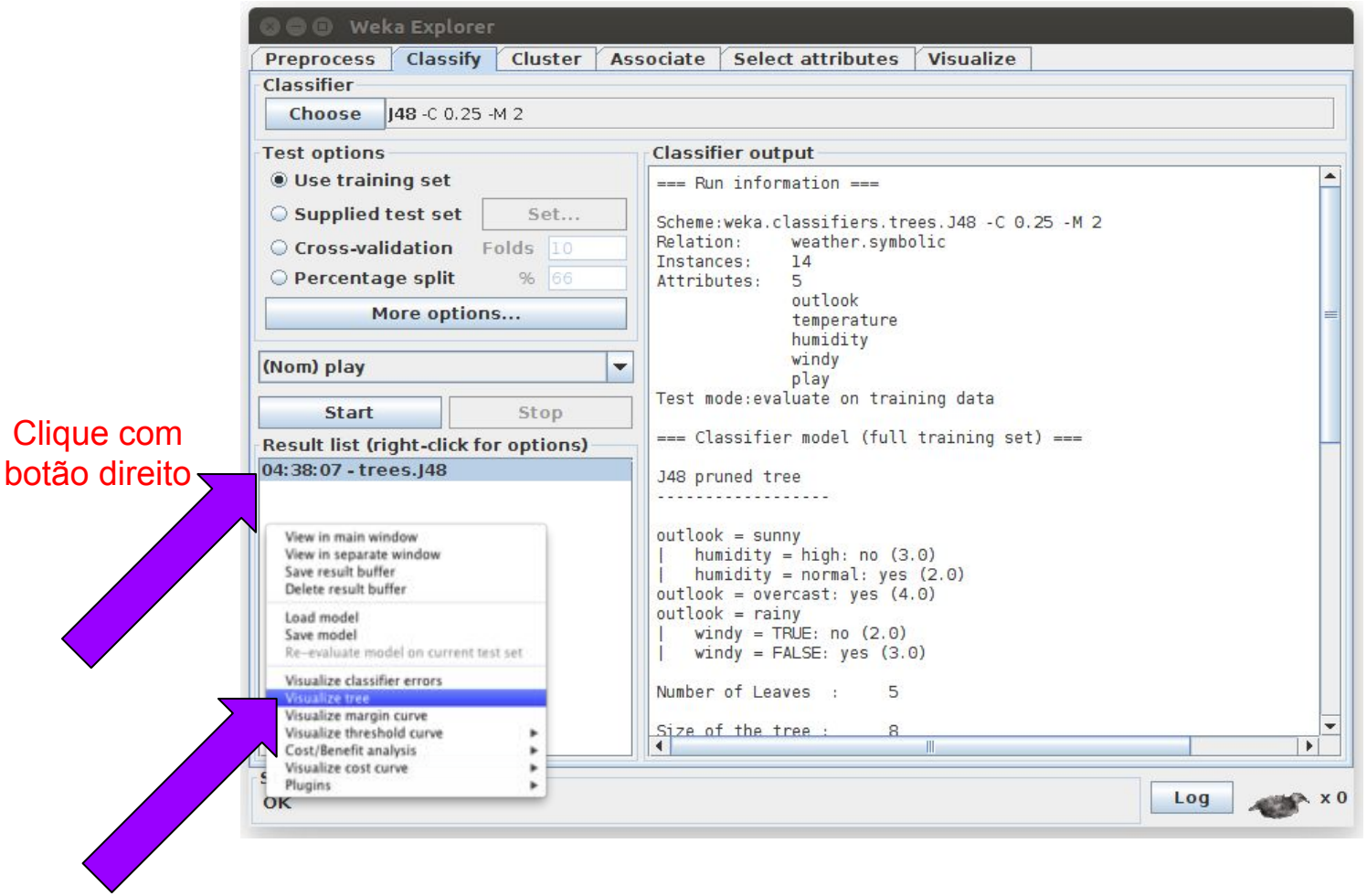


Classifier output

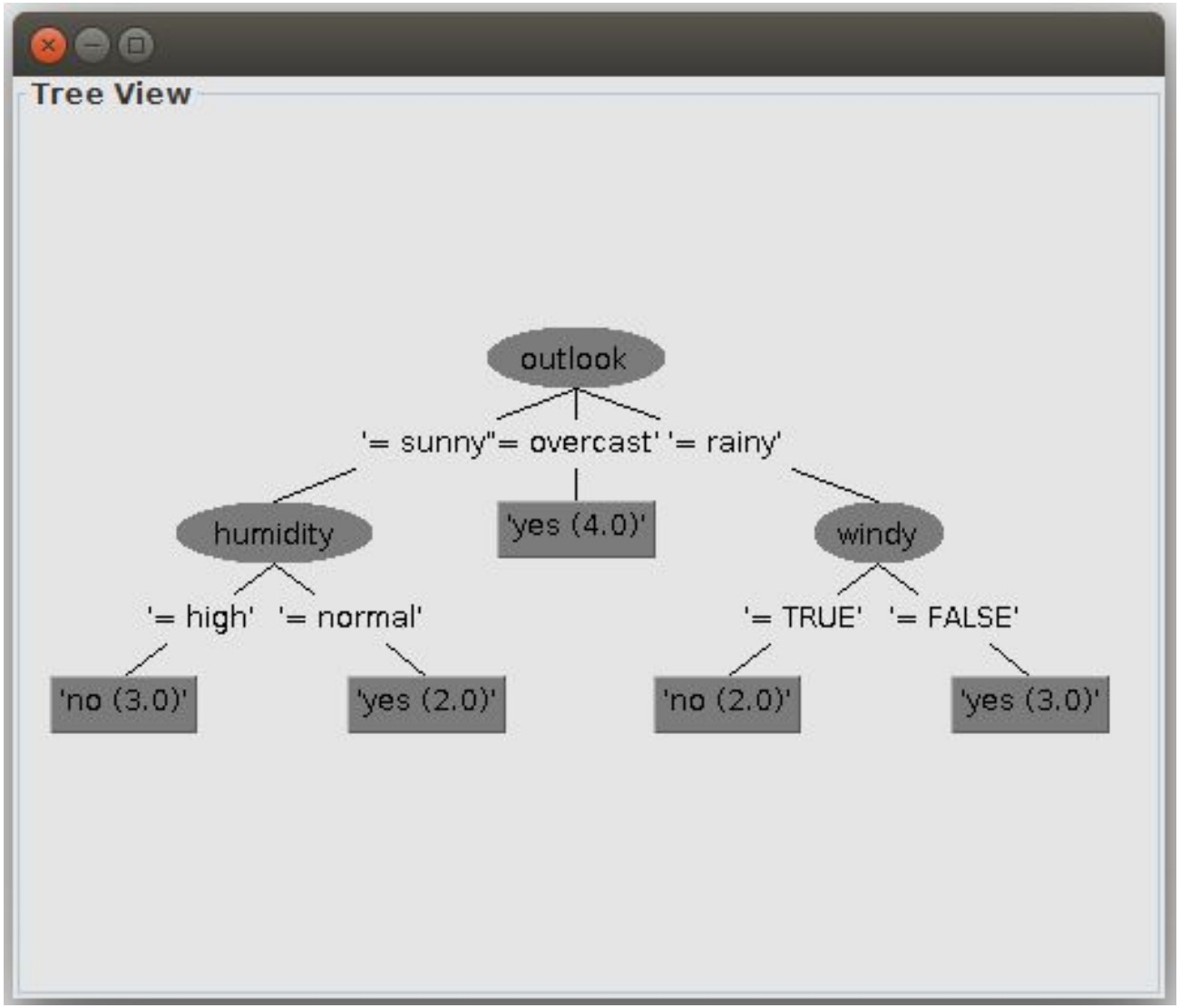


Árvore / Regras Geradas

Visualize


Clique com botão direito
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) play
Start
Stop
Result list (right-click for options)
04:38:07 - trees.J48
View in main window
View in separate window
Save result buffer
Delete result buffer
Load model
Save model
Re-evaluate model on current test set
Visualize classifier errors
Visualize tree
Visualize margin curve
Visualize threshold curve
Cost/Benefit analysis
Visualize cost curve
Plugins
Classifier output
=== Run information ===
Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: weather.symbolic
Instances: 14
Attributes: 5
outlook
temperature
humidity
windy
play
Test mode:evaluate on training data
=== Classifier model (full training set) ===
J48 pruned tree
outlook = sunny
| humidity = high: no (3.0)
| humidity = normal: yes (2.0)
outlook = overcast: yes (4.0)
outlook = rainy
| windy = TRUE: no (2.0)
| windy = FALSE: yes (3.0)
Number of Leaves : 5
Size of the tree : 8
OK
Log
x 0

Visualize


Tree View
outlook
'= sunny"= overcast' '= rainy'
humidity
'yes (4.0)'
windy
'= high' '= normal'
'= TRUE' '= FALSE'
'no (3.0)'
'yes (2.0)'
'no (2.0)'
'yes (3.0)'

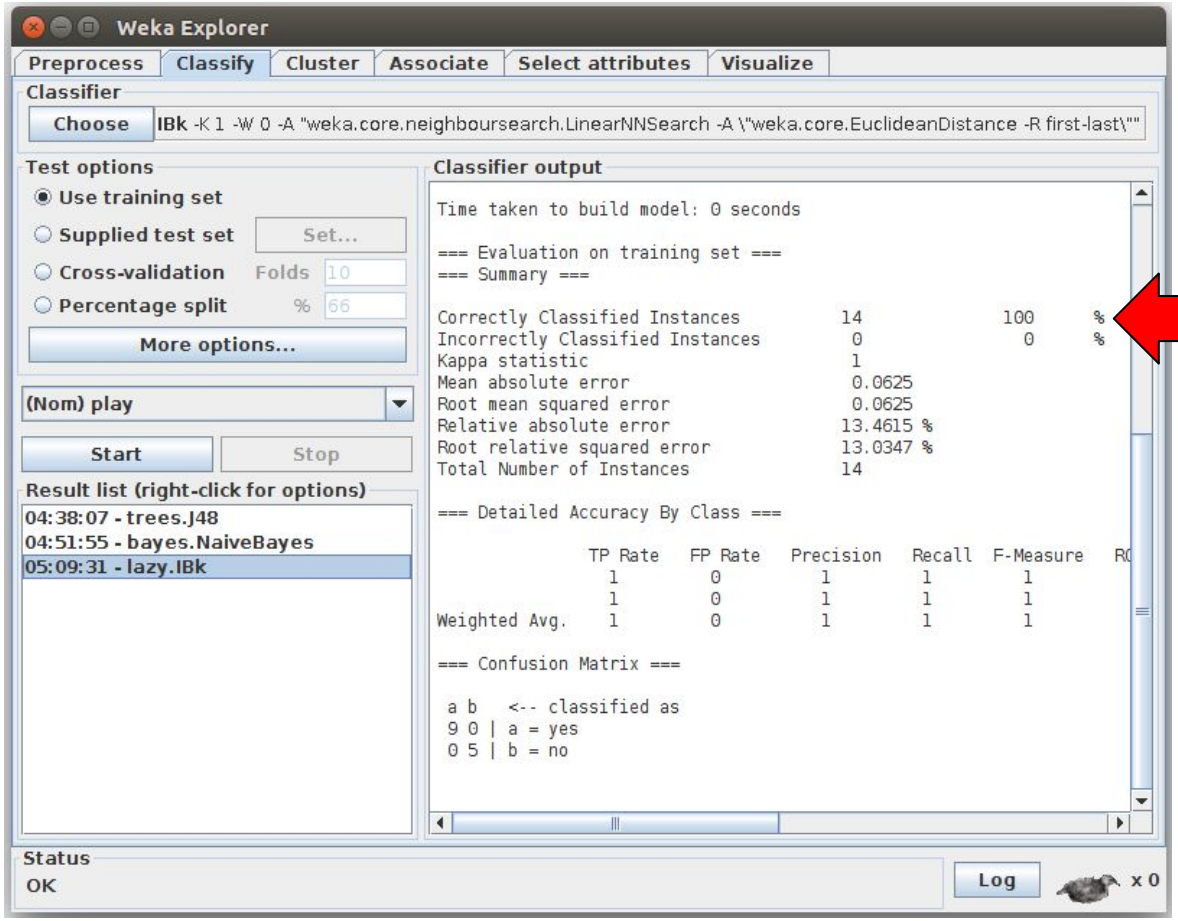
Código em Java

```
import java.awt.BorderLayout;
import java.io.BufferedReader;
import java.io.FileReader;

import weka.classifiers.*;
import weka.classifiers.trees.J48;
import weka.core.Instances;
import weka.gui.treevisualizer.PlaceNode2;
import weka.gui.treevisualizer.TreeVisualizer;

public class WekaJ48 {
public static void main(String args[]) throws Exception {
    // train classifier
    J48 cls = new J48();
    Instances data = new Instances(new BufferedReader(new FileReader("D:\\sample.arff")));
    data.setClassIndex(data.numAttributes() - 1);
    cls.buildClassifier(data);
```

Código em Java

```
    // display classifier
    final javax.swing.JFrame jf =
      new javax.swing.JFrame("Weka Classifier Tree Visualizer: J48");
    jf.setSize(500,400);
    jf.getContentPane().setLayout(new BorderLayout());
    TreeVisualizer tv = new TreeVisualizer(null,
        cls.graph(),
        new PlaceNode2());
    jf.getContentPane().add(tv, BorderLayout.CENTER);
    jf.addWindowListener(new java.awt.event.WindowAdapter() {
      public void windowClosing(java.awt.event.WindowEvent e) {
        jf.dispose();
      }
    });

    jf.setVisible(true);
    tv.fitToScreen();
  }
}
```

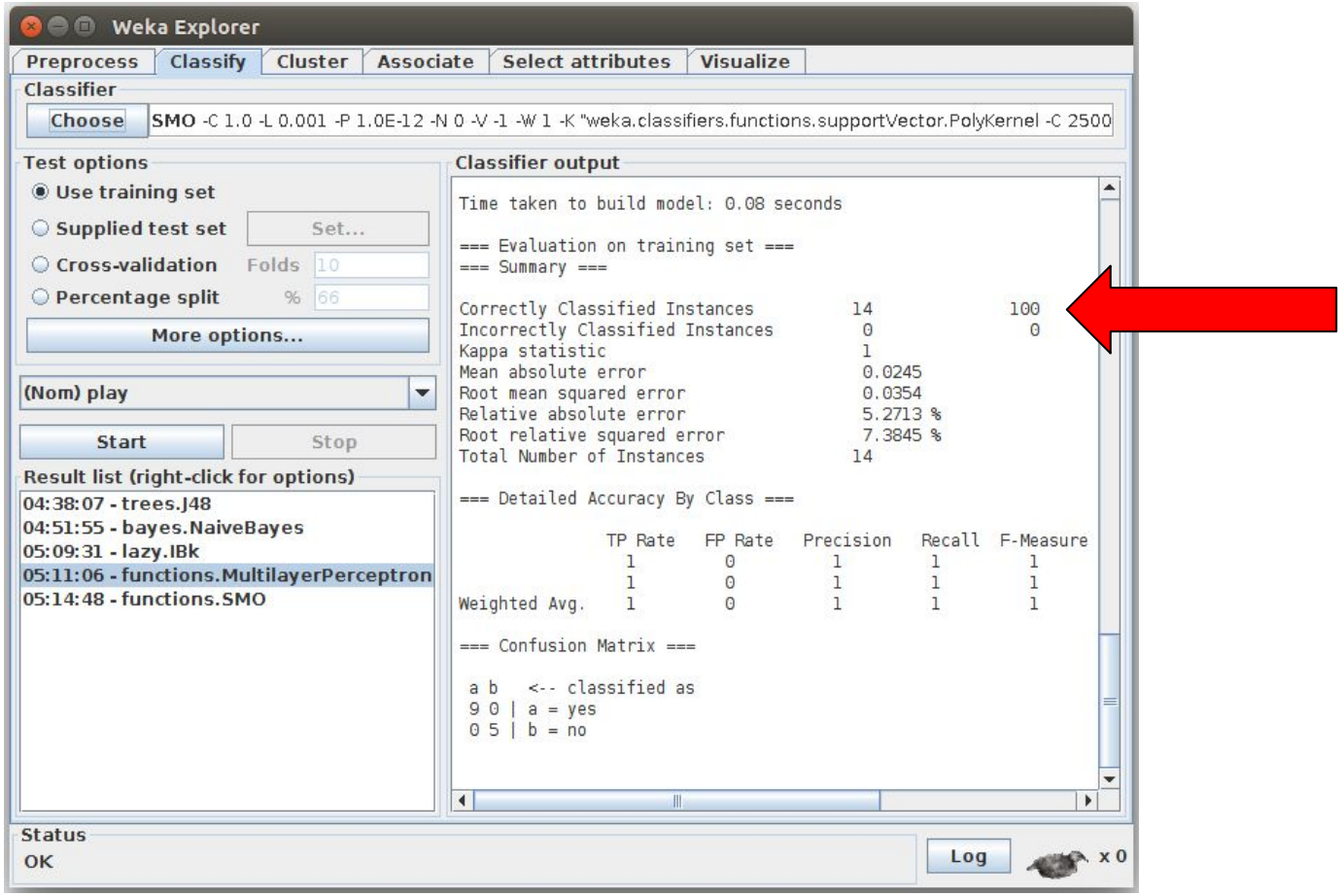
Classify > Lazy > k-NN (IBk)



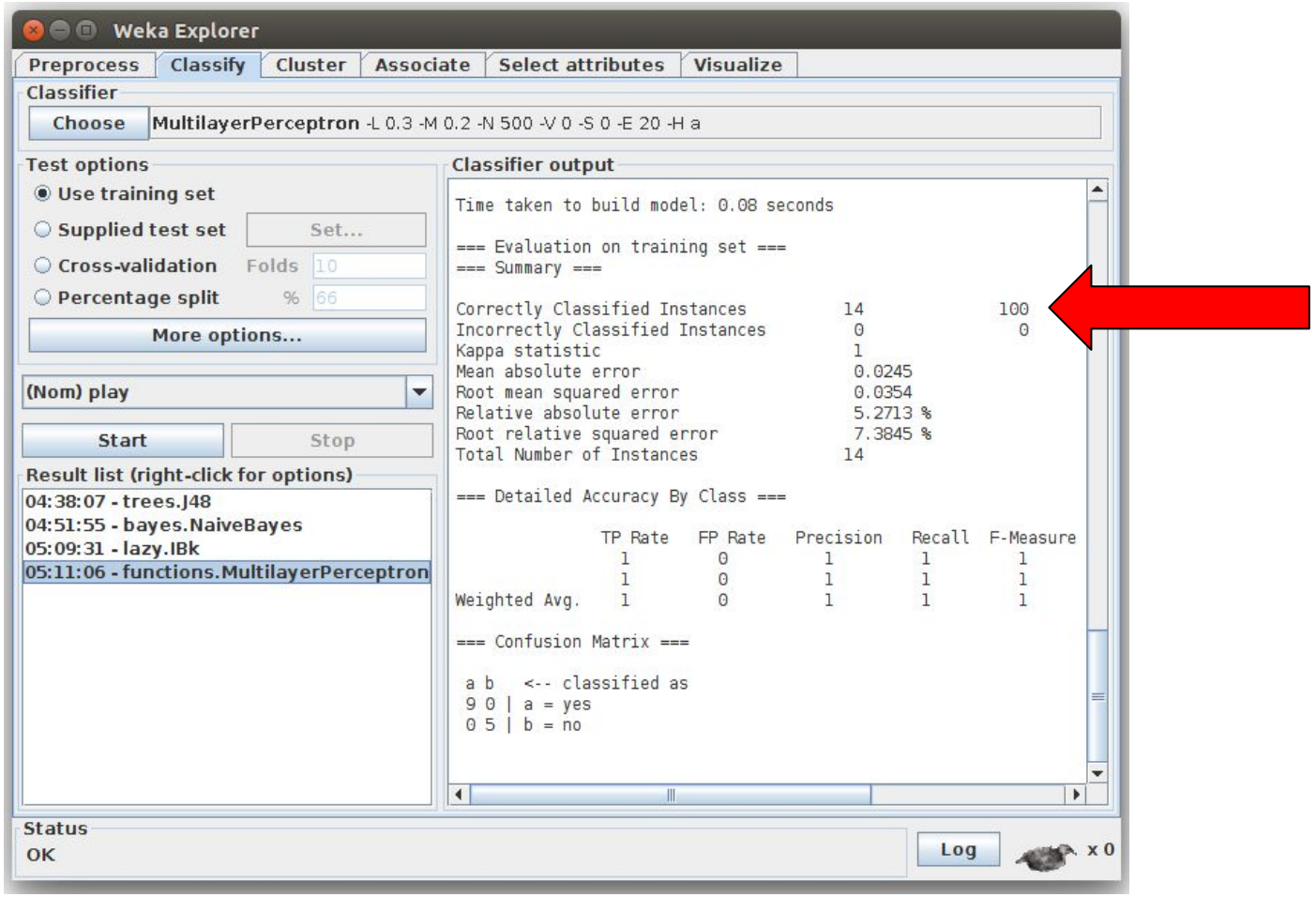
Classify > function > SVM

- Deve-se instalar o LibSVM
  - LIBSVM - A Library for Support Vector Machines
  - https://www.csie.ntu.edu.tw/~cjlin/libsvm/

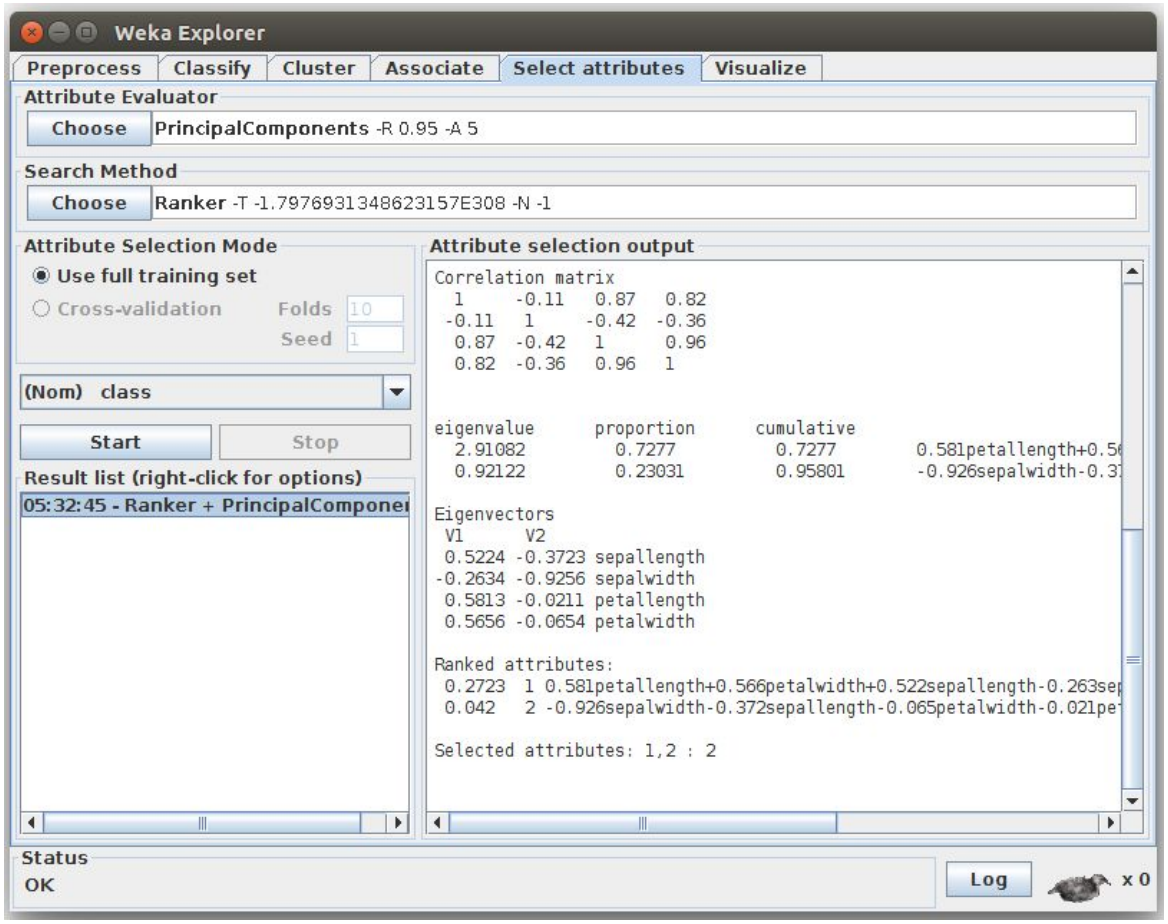
Classify > function > SMO



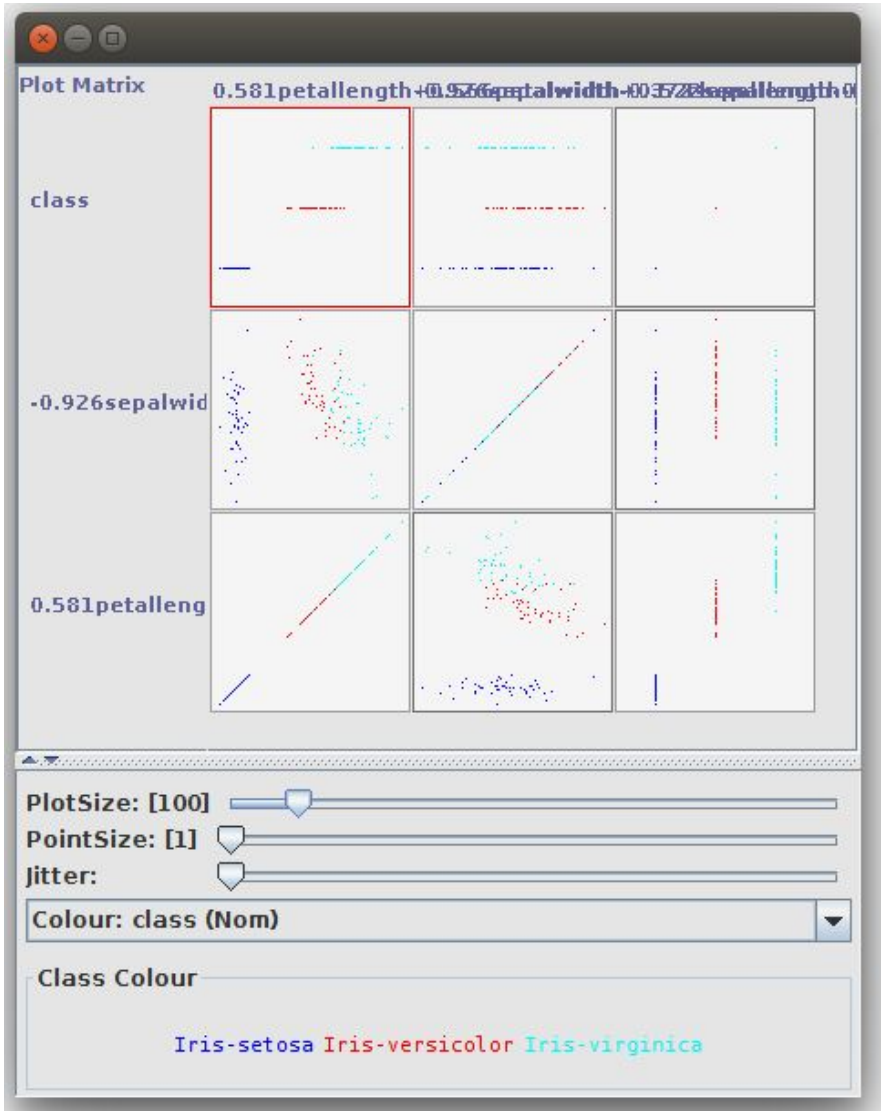
Classify > function > MLP



Select Attributes > PCA



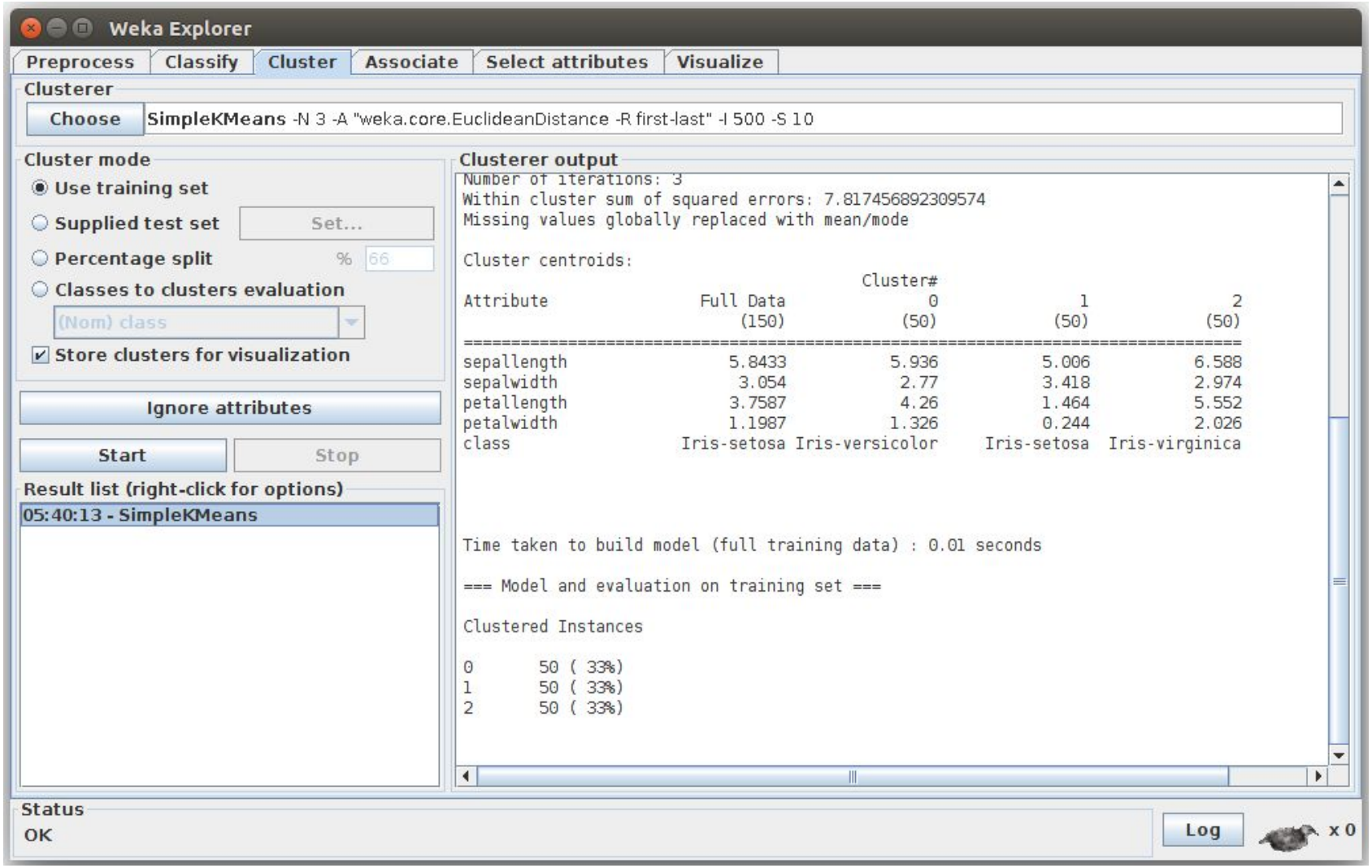
Visualizar


Plot Matrix
class
-0.926sepalwid
0.581petalleng
PlotSize: [100]
PointSize: [1]
Jitter:
Colour: class (Nom)
Class Colour
Iris-setosa Iris-versicolor Iris-virginica

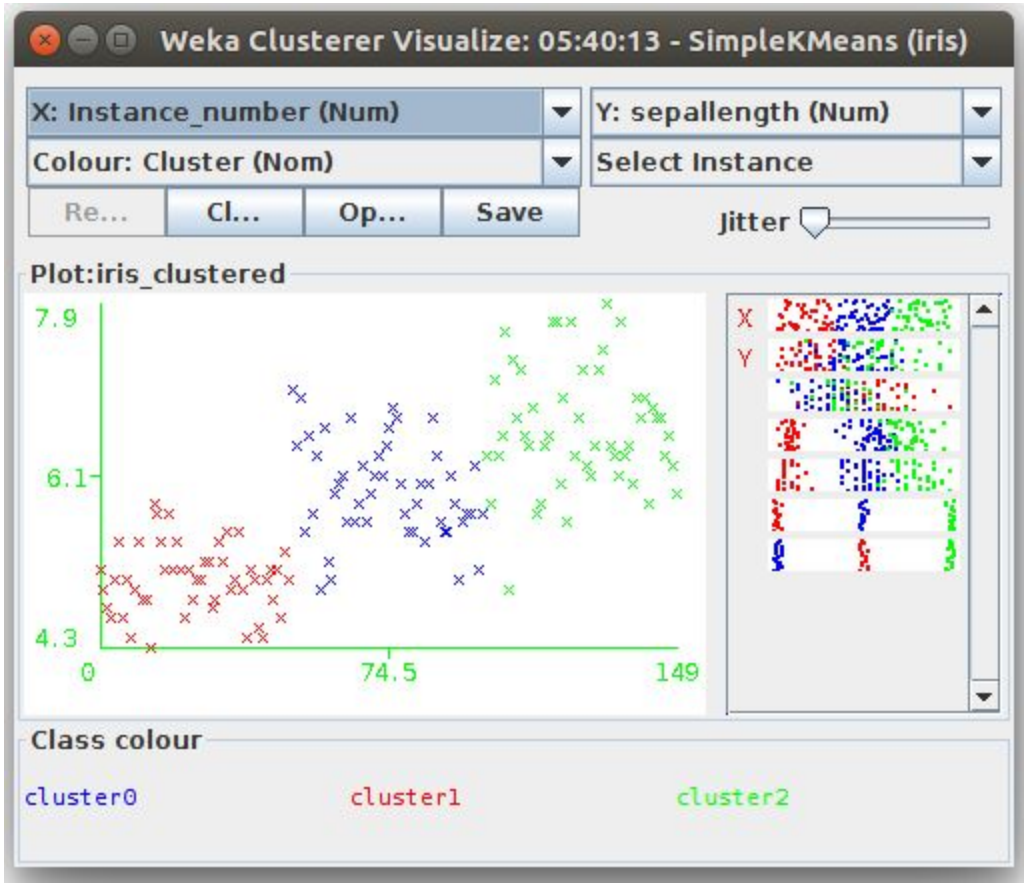
Clustering & Regressão

- Como gerar:
  - Um kMeans
  - Uma regressão linear

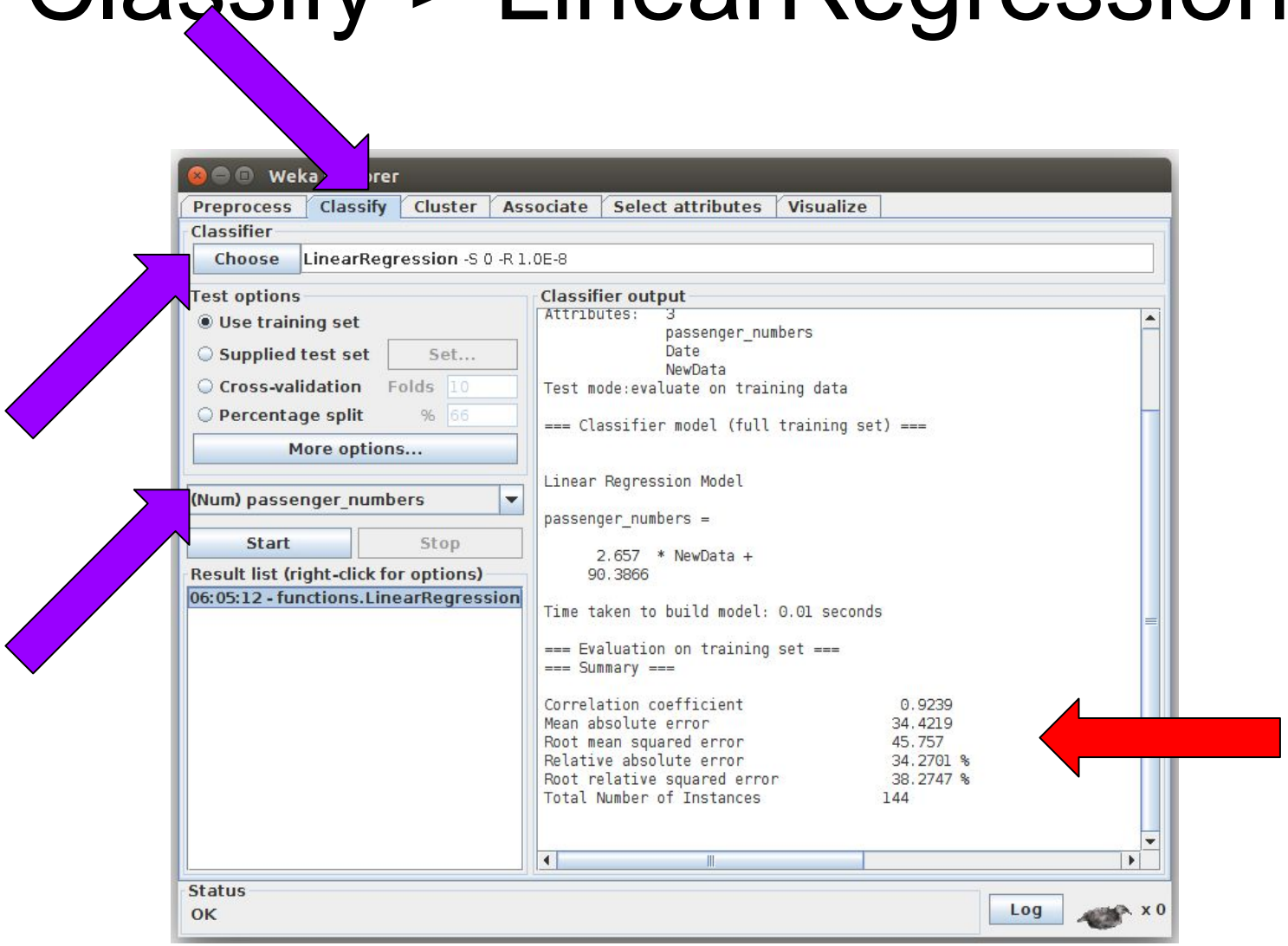
Cluster > SimpleKmeans



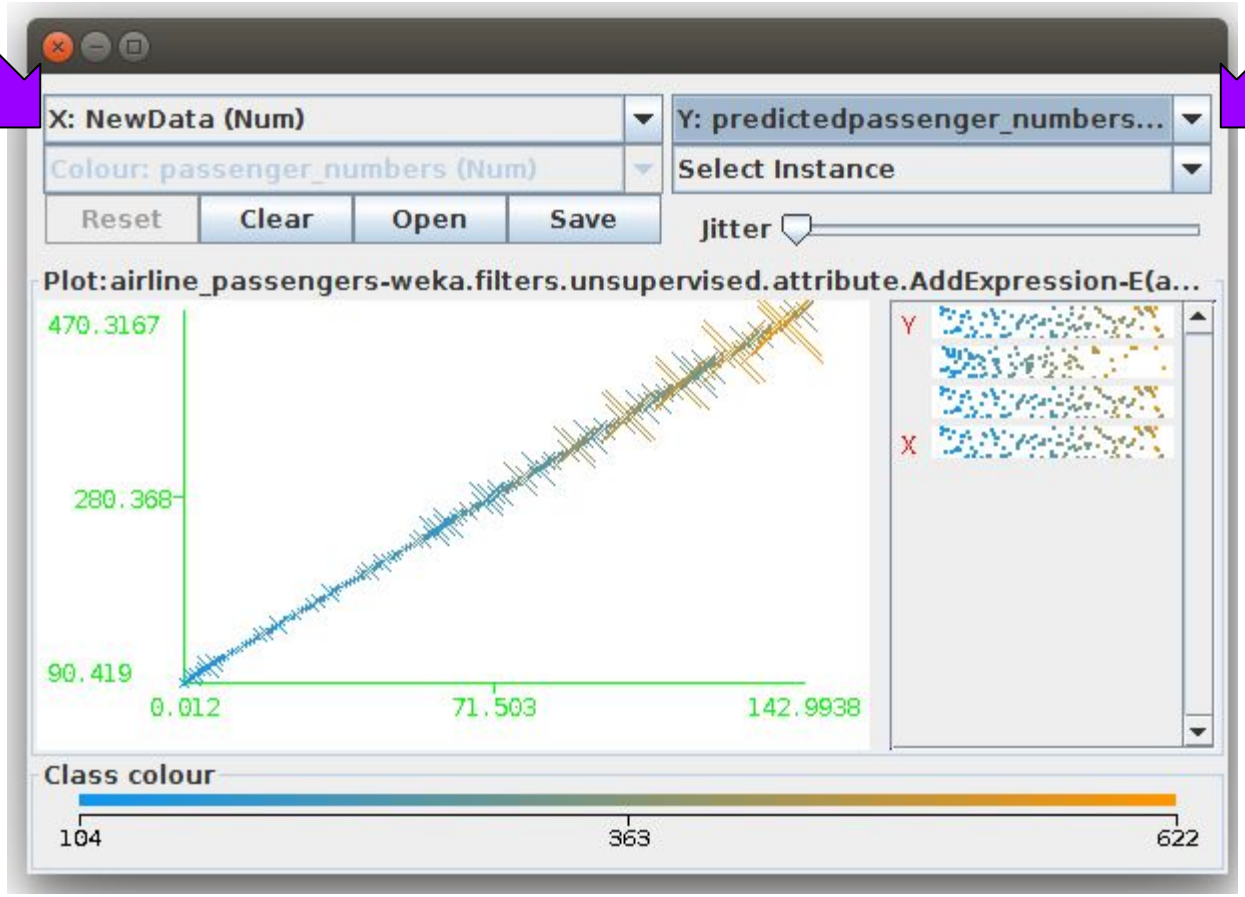
Cluster > SimpleKmeans



Classify > LinearRegression



Classify > LinearRegression



References

- Weka 3: Data Mining Software in Java
  - https://www.cs.waikato.ac.nz/ml/weka/

- Weka Datasets
  - http://storm.cis.fordham.edu/~gweiss/data-mining/datasets.html