



# Redes Neurais (Inteligência Artificial)

Aula 15 – Bibliotecas e Ferramentas para  
Aprendizado de Máquina

Edirlei Soares de Lima  
<edirlei@iprj.uerj.br>



# LibSVM

- Biblioteca com implementação atual e bem otimizada de **Support Vector Machine (SVM)**.
  - É escrita originalmente em C e Java, mas possui versões em C#, Python, Ruby, Perl, Haskell, Lisp, PHP, CUDA...
  - Versão atual 3.22.
- 

# LibSVM

- **LibSVM:**

- <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

- **GNUPlot:**

- <http://sourceforge.net/projects/gnuplot/files/gnuplot/4.4.3/gp443win32.zip/download>

- **Python:**

- <https://www.python.org/downloads/release/python-2713/>

- **Exemplos de Datasets:**

- <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>

# LibSVM

- **Instalação:**
  - **Descompacte:** libsvm-3.22.zip e gp443win32.zip
  - **Instale:** python-2.7.13.msi
  - **Obs:** É necessário usar a versão 2.7 do Python.

# LibSVM

- **Configuração:**

- **Edite o arquivo:** libsvm-3.22\tools\easy.py

Substitua a linha 25:

```
gnuplot_exe = r"c:\tmp\gnuplot\binary\pgnuplot.exe"
```

Pelo caminho do GNUPlot. Exemplo:

```
gnuplot_exe = r" C:\gp443win32\gnuplot\binary\gnuplot.exe"
```

# LibSVM

- **Execução:**

- Considerando os arquivos **train.txt** e **test.txt** como bases de treinamento e teste, o processo de treinamento e teste é executado pela seguinte linha de comando:

```
C:\Python27\python.exe easy.py train.txt test.txt
```

# LibSVM

- **Formato do arquivo de treinamento/testes:**

(Classe) (Atrib<sub>1</sub>ID):(Atrib<sub>1</sub>) (Atrib<sub>2</sub>ID):(Atrib<sub>2</sub>) ... (Atrib<sub>N</sub>ID):(Atrib<sub>N</sub>)

## **Exemplo:**

8 1:47 2:100 3:27 4:81 5:57 6:37 7:26

6 1:100 2:100 3:88 4:99 5:49 6:74 7:17

3 1:50 2:84 3:66 4:100 5:75 6:75 7:51

8 1:48 2:96 3:62 4:65 5:88 6:27 7:21

3 2:83 3:29 4:100 5:88 6:95 7:64

# LibSVM

- **Avaliação dos Resultados:**

- **Precisão geral:** Valor do accuracy exibido no console.

- **Arquivos Gerados:**

**Train.txt.scale** e **Test.txt.scale** – Contém os dados normalizados.

**Train.txt.range** e **Test.txt.range** – Valor mínimo e máximo dos atributos.

**Train.txt.scale.out** – Contém os resultados parciais obtidos com diferentes parâmetros durante o treinamento.

**Train.txt.scale.png** – Gráfico com a variação dos resultados obtidos com diferentes parâmetros durante o treinamento.

**Test.txt.predict** – Resultado da classificação dos exemplos de teste.

**Train.txt.model** – Modelo do classificador treinado.



# Weka

- Ferramenta e biblioteca completa para tarefas de aprendizado de máquina.
- Possui uma grande quantidade de algoritmos de aprendizado de máquina.
- Implementado na linguagem Java.
- Versão atual: 3.6



# Weka

- **Weka:**
  - <http://www.cs.waikato.ac.nz/ml/weka/>
- **Exemplos de Datasets:**
  - “C:\Program Files\Weka-3-6\data\”
  - [http://www.cs.waikato.ac.nz/ml/weka/index\\_datasets.html](http://www.cs.waikato.ac.nz/ml/weka/index_datasets.html)

# Weka

- **Formato do arquivo de treinamento/testes:**

## **Cabeçalho do arquivo:**

@relation (DatasetName)

@attribute (AtribName<sub>1</sub>) (AtribType<sub>1</sub>)

@attribute (AtribName<sub>2</sub>) (AtribType<sub>2</sub>)

.

.

@attribute (AtribName<sub>N</sub>) (AtribType<sub>N</sub>)

@attribute (Class) {(Class<sub>1</sub>, Class<sub>2</sub>, ... Class<sub>N</sub>)}

@data

(Atrib<sub>1</sub>), (Atrib<sub>2</sub>), ... , (Atrib<sub>N</sub>),(Class)

.

.

# Weka

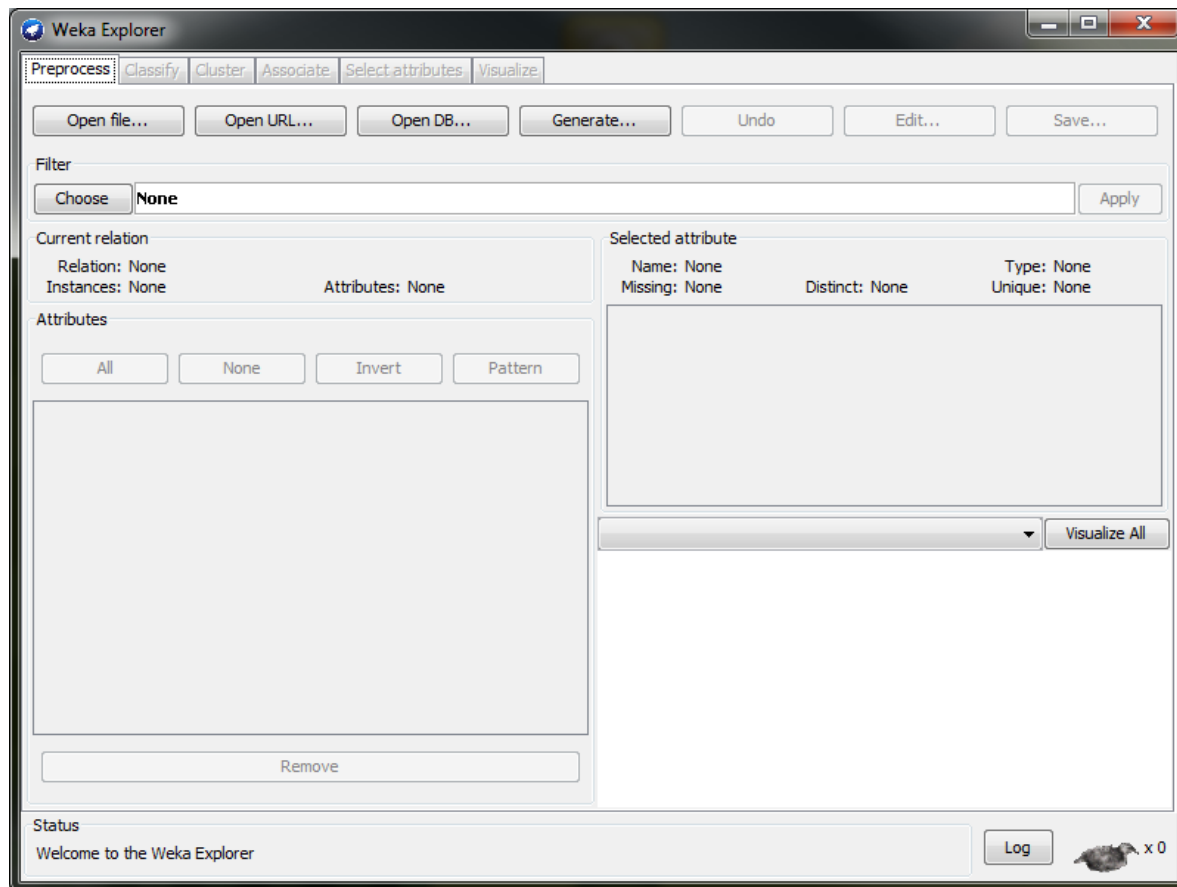
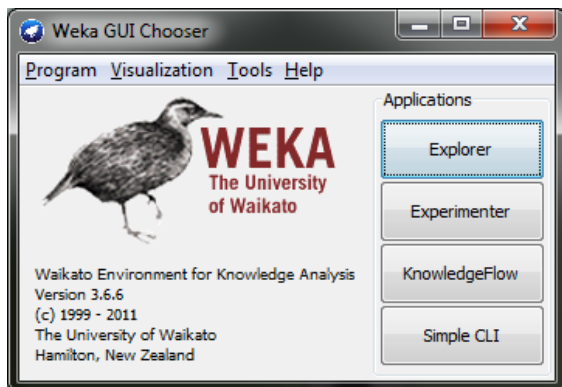
- **Formato do arquivo de treinamento/testes:**

**Exemplo:**

```
@relation Activity_Recognition
@attribute 'valor1' real
@attribute 'valor2' real
@attribute 'valor3' real
@attribute 'class' {0,1,2,3,4,5,6,7,8,9,10}
@data
0.24679,0.210083,0.0873606,0
0.546452,0.811992,0.0163704,1
0.745887,0.114372,0.0957822,3
0.245887,0.214372,0.0857822,0
```

# Weka

- Interface para Testes:



# Weka

- **Selecionando uma Base de Treinamento:**

The screenshot shows the Weka Explorer application window. The 'Preprocess' tab is active. The 'Open file...' button is circled in red. The 'Current relation' is 'faceExpressions' with 1575 instances and 37 attributes. The 'Attributes' list shows 'rightEyeMaxWidth' selected. The 'Selected attribute' panel displays statistics for 'rightEyeMaxWidth'.

**Current relation**  
Relation: faceExpressions  
Instances: 1575      Attributes: 37

**Attributes**

No.	Name
1	<input checked="" type="checkbox"/> rightEyeMaxWidth
2	<input type="checkbox"/> rightEyeMaxHeight
3	<input type="checkbox"/> rightEyeAngle
4	<input type="checkbox"/> rightEyeArcLength
5	<input type="checkbox"/> rightEyeArea
6	<input type="checkbox"/> rightEyeHU1
7	<input type="checkbox"/> rightEyeHU2
8	<input type="checkbox"/> rightEyeHU3
9	<input type="checkbox"/> rightEyeHU4
10	<input type="checkbox"/> rightEyeHU5
11	<input type="checkbox"/> rightEyeHU6
12	<input type="checkbox"/> rightEyeHU7
13	<input type="checkbox"/> leftEyeMaxWidth

**Selected attribute**  
Name: rightEyeMaxWidth      Type: Numeric  
Missing: 0 (0%)      Distinct: 1575      Unique: 1575 (100%)

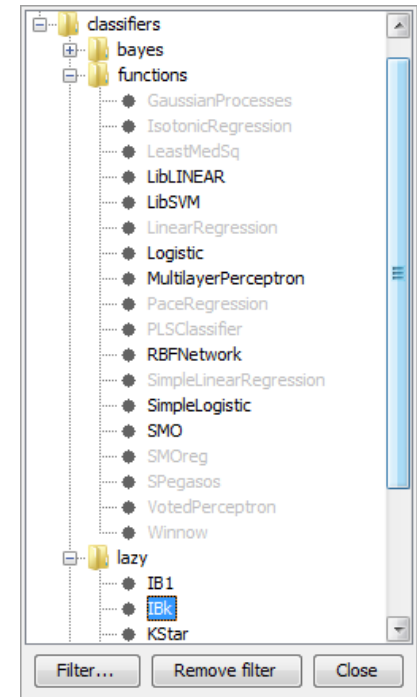
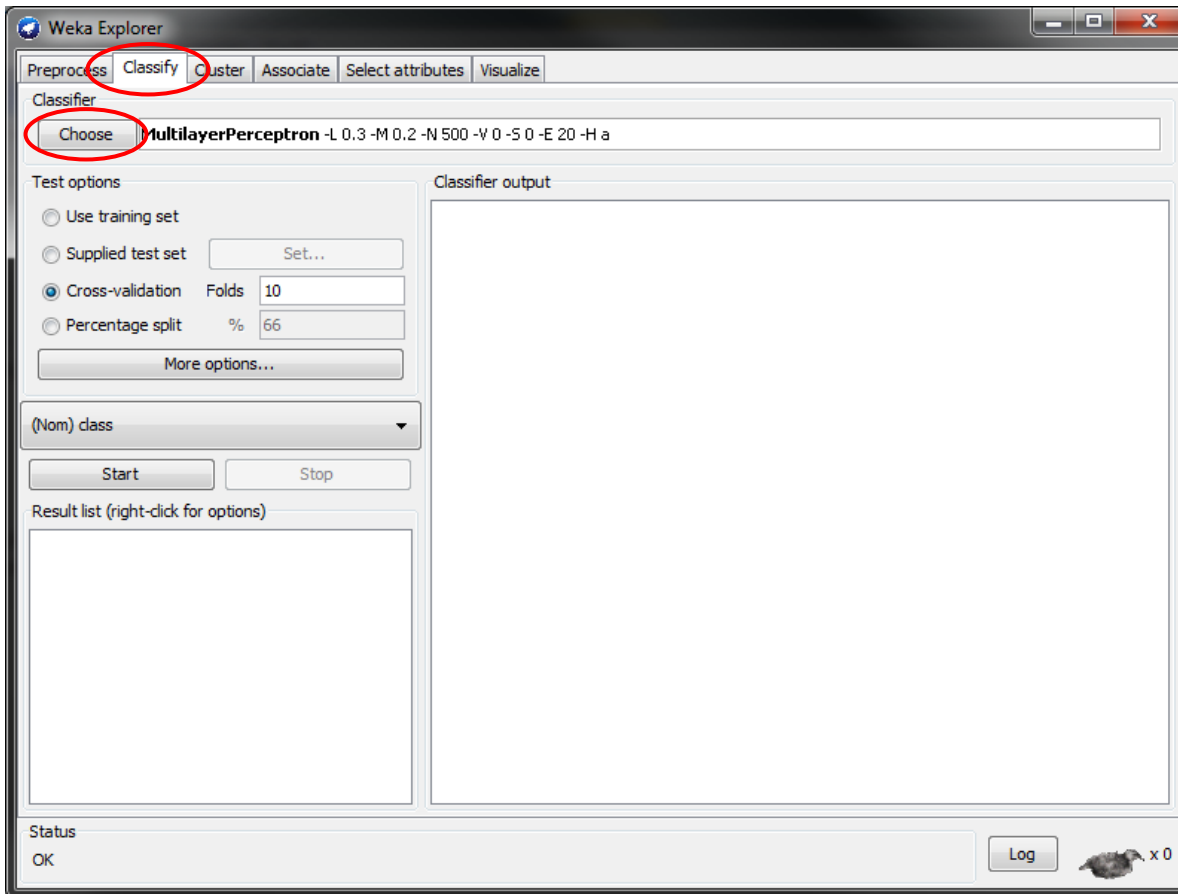
Statistic	Value
Minimum	5.266
Maximum	55.685
Mean	30.703
StdDev	7.467

Class: class (Nom)      Visualize All

Status: OK      Log      x 0

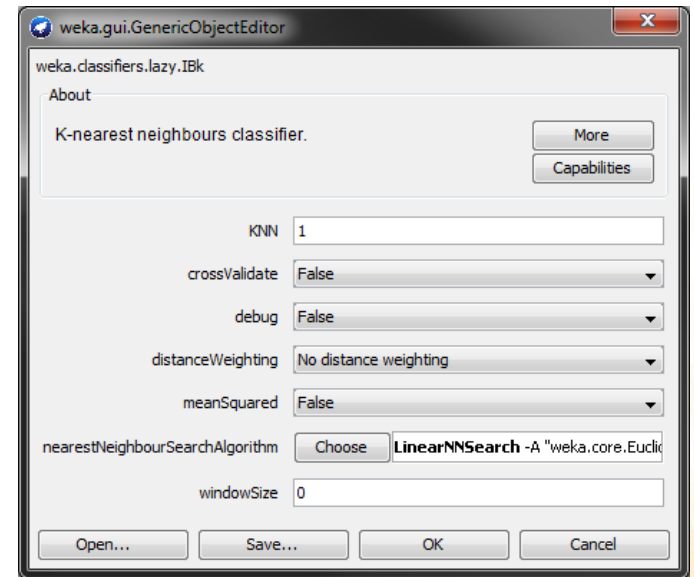
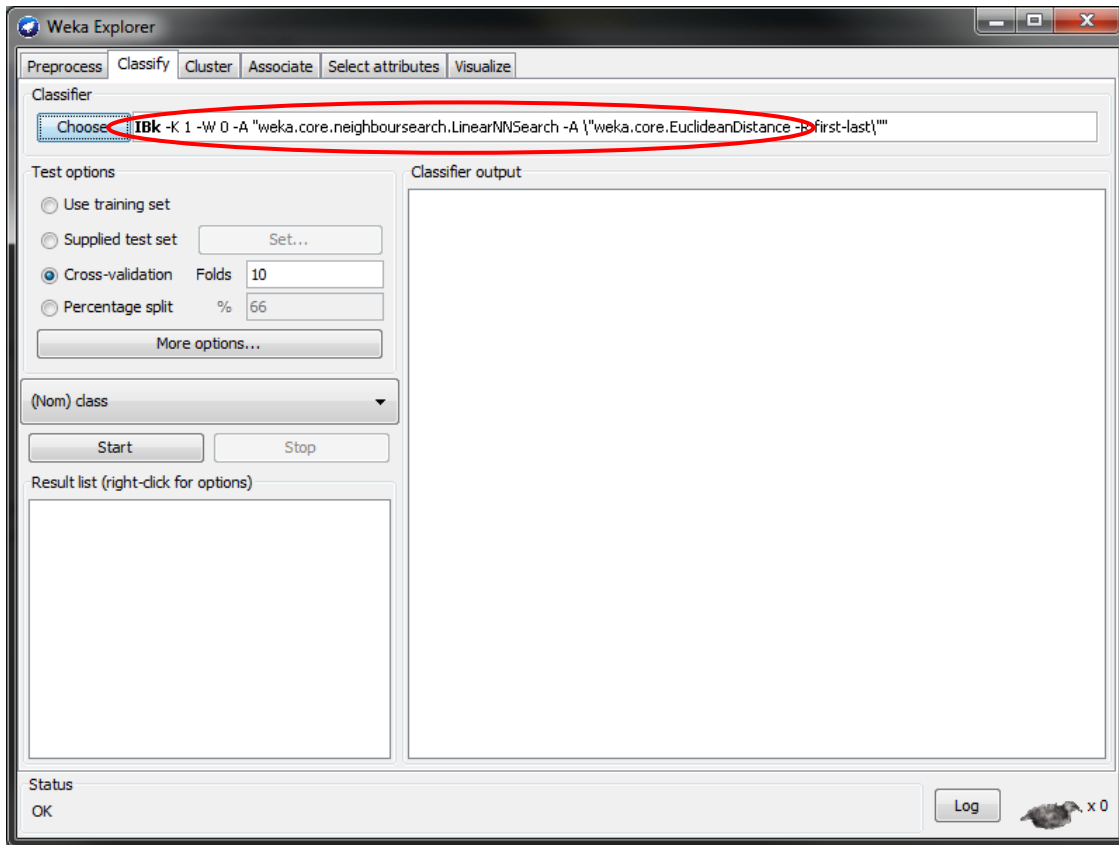
# Weka

- **Selecionando um Algoritmo:**



# Weka

- **Alterando Parâmetros do Algoritmo:**





# Weka

- Realizando Testes:

The screenshot shows the Weka Explorer interface. The 'Classifier' tab is active, displaying the selected classifier: `IBk -K 5 -W 0 -A "weka.core.neighboursearch.KDTree -A \"weka.core.EuclideanDistance -R first-last\" -S weka.core.neighboursearch.kdtrees.SlidingMidP`. The 'Test options' section has 'Percentage split' selected with a value of 66%. The 'Start' button is circled in red. The 'Classifier output' pane shows the following results:

Root relative squared error: 80.4803 %  
Total Number of Instances: 535

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.563	0.039	0.69	0.563	0.62	0.911	0
	0.439	0.126	0.294	0.439	0.352	0.782	1
	0.633	0.051	0.613	0.633	0.623	0.885	2
	0.565	0.062	0.574	0.565	0.569	0.866	3
	0.833	0.009	0.932	0.833	0.88	0.994	4
	0.514	0.104	0.442	0.514	0.475	0.827	5
	0.403	0.072	0.424	0.403	0.413	0.796	6
	0.579	0.031	0.759	0.579	0.657	0.916	7
Weighted Avg.	0.568	0.06	0.599	0.568	0.579	0.874	

=== Confusion Matrix ===

	a	b	c	d	e	f	g	h	<-- classified as
40	12	8	0	0	4	6	1		a = 0
3	25	2	3	1	11	11	1		b = 1
7	3	38	1	0	6	4	1		c = 2
1	7	2	39	1	8	3	8		d = 3
1	1	5	4	55	0	0	0		e = 4
2	23	1	1	0	38	6	3		f = 5
4	14	4	3	0	12	25	0		g = 6

Status: OK

# Weka

- Analizando os Resultados:

Time taken to build model: 0.07 seconds

=== Evaluation on test split ===

=== Summary ===

Correctly Classified Instances	304	56.8224 %
Incorrectly Classified Instances	231	43.1776 %
Kappa statistic	0.5067	
Mean absolute error	0.1275	
Root mean squared error	0.2664	
Relative absolute error	58.2602 %	
Root relative squared error	80.4803 %	
Total Number of Instances	535	

=== Confusion Matrix ===

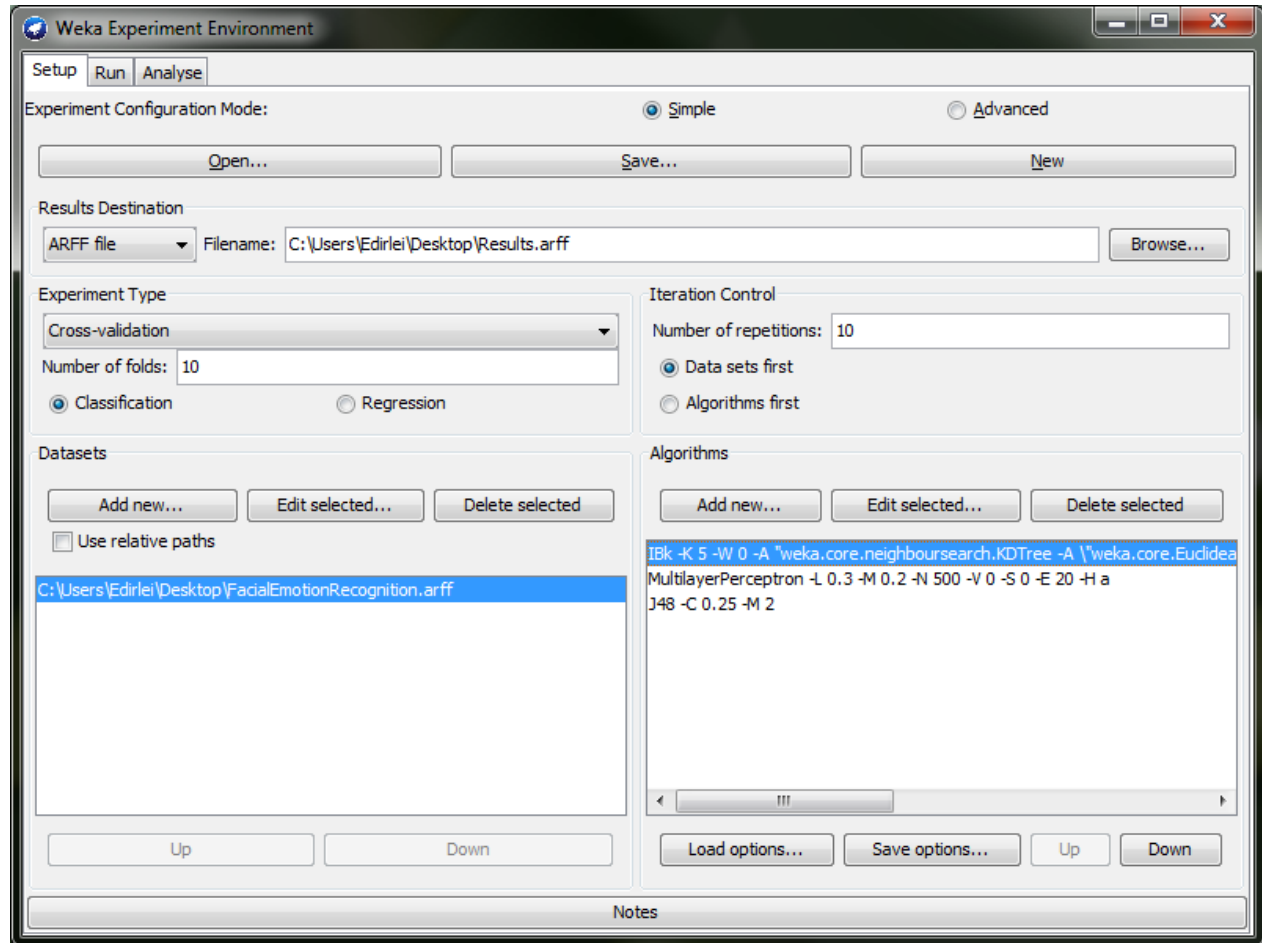
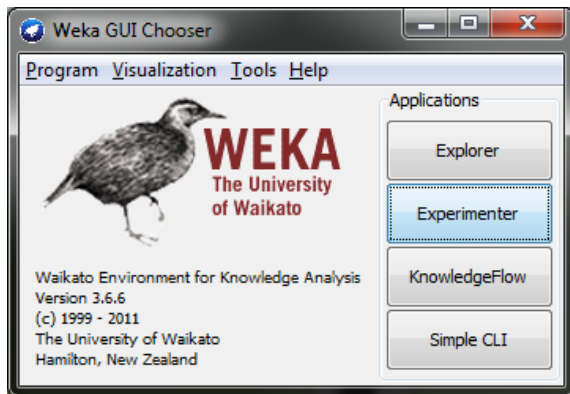
	a	b	c	d	e	f	g	h	<-- classified as
40	12	8	0	0	4	6	1	1	a = 0
3	25	2	3	1	11	11	1	1	b = 1
7	3	38	1	0	6	4	1	1	c = 2
1	7	2	39	1	8	3	8	1	d = 3
1	1	5	4	55	0	0	0	1	e = 4
2	23	1	1	0	38	6	3	1	f = 5
4	14	4	3	0	12	25	0	1	g = 6
0	0	2	17	2	7	4	44	1	h = 7

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.563	0.039	0.69	0.563	0.62	0.911	0
	0.439	0.126	0.294	0.439	0.352	0.782	1
	0.633	0.051	0.613	0.633	0.623	0.885	2
	0.565	0.062	0.574	0.565	0.569	0.866	3
	0.833	0.009	0.932	0.833	0.88	0.994	4
	0.514	0.104	0.442	0.514	0.475	0.827	5
	0.403	0.072	0.424	0.403	0.413	0.796	6
	0.579	0.031	0.759	0.579	0.657	0.916	7
Weighted Avg.	0.568	0.06	0.599	0.568	0.579	0.874	

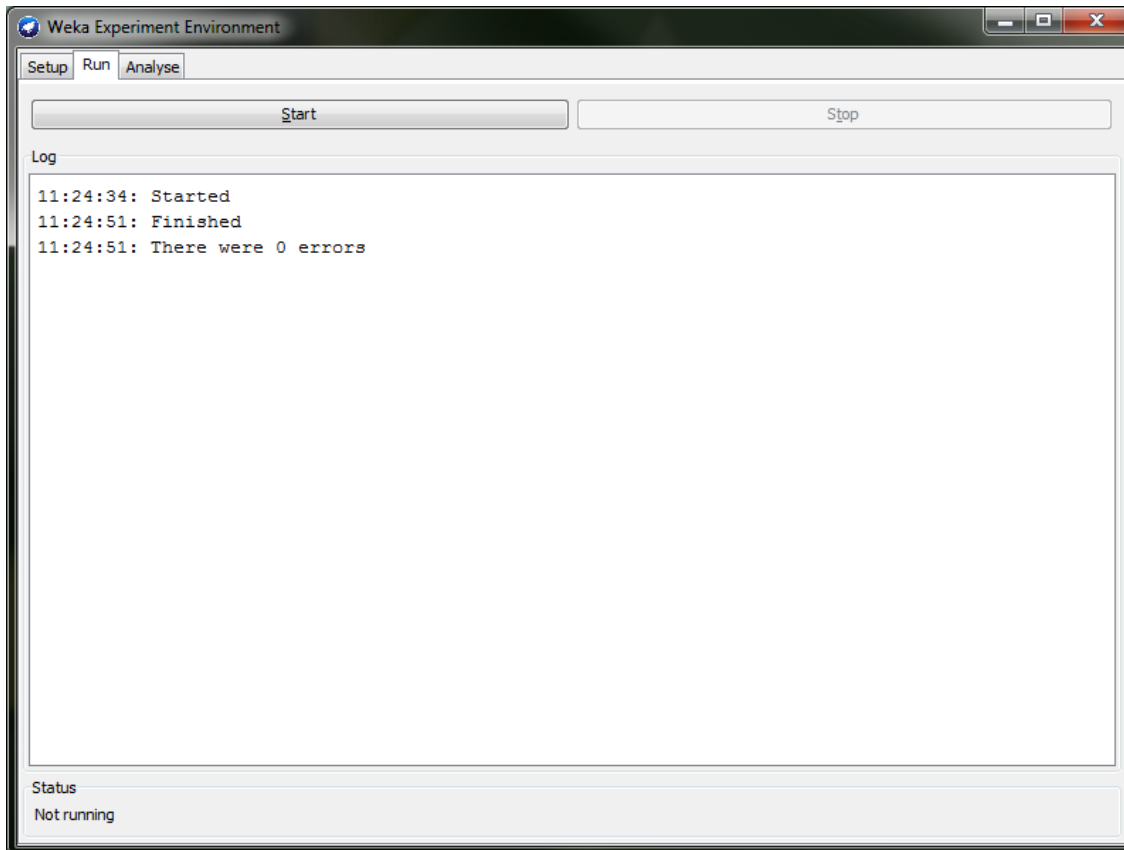
# Weka

- Realizando Experimentos:



# Weka

- **Realizando Experimentos:**



# Weka

- Comparando os Resultados dos Experimentos:

