

# WEKA ([www.cs.waikato.ac.nz/ml/weka/](http://www.cs.waikato.ac.nz/ml/weka/))

- WEKA: *Waikato Environment for Knowledge Analysis*
- WEKA es un conjunto de herramientas sobre *Machine Learning*
- Está escrito en Java y distribuido con licencia GNU pública
- Ejecución:
  - desde el menú inicio o haciendo click en el ejecutable
  - desde línea de comandos: `java -jar weka.jar`
  - o haciendo doble click en un archivo `.arff`
  - si hay errores por falta de memoria aumentar el tamaño del heap de la máquina virtual de java:
    - `java -Xmx1024M -jar weka.jar`

# WEKA

- Los algoritmos pueden ser aplicados a un fichero (dataset)
  - Desde la GUI: explorer, experimenter, (knowledge flow, simpleCLI)
  - Desde línea de comandos del S.O.  

```
java weka.classifiers.trees.J48 -t C:\ejemplos\zoo.arff
```
  - Desde un programa Java
- Contiene herramientas para:
  - Filtros de Preprocesamiento de datos: organiza los datos
  - Clasificación (aprendizaje supervisado)
  - Regresión
  - Agrupamiento (*clustering*) (aprendizaje no supervisado)
  - Reglas de asociación
  - Visualización

# WEKA

Weka - Log

---Registering Weka Editors---  
Trying to add database driver (JDBC):  
RmiJdbc.RJDriver - Error, not in CLASSPATH?  
Trying to add database driver (JDBC): jdbc.idbDriver  
max. Size 100.000 currently: 455 Clear Close

ARFF-Viewer- C:\A-Z\Acad\IAIC\...

File Edit View

zoo.arff

Relation: zoo

No.	animal	hair	feathers	eggs	milk	airborn
1	aardvark	true	false	false	true	false
10	cavy	true	false	false	true	false
100	worm	false	false	true	false	false
101	wren	false	true	true	false	true
11	cheetah	true	false	false	true	false
12	chicken	false	true	true	false	true
13	chub	false	false	true	false	false
14	clam	false	false	true	false	false

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Open file... Open URL... Open DB... Generate... Undo Edit...

Filter Choose None

Current relation  
Relation: zoo  
Instances: 101  
Attributes: 18

Selected attribute  
Name: animal  
Missing: 0 (0%)  
Distinct: 100

Attributes  
All None Invert Pattern

No.	Name
1	animal
2	hair
3	feathers
4	eggs
5	milk
6	airborne
7	aquatic
8	predator
9	toothed
10	backbone

Weka Experiment Environment

Setup Run Analyse

Experiment Configuration Mode: Simple Advanced

Open... Save... New

Results Destination  
ARFF file Filename: Browse

Experiment Type  
Cross-validation  
Number of folds: 10  
Classification Regression

Iteration Control  
Number of repetitions: 10  
Data sets first Algorithms first

Weka KnowledgeFlow Environment


DataSources DataSinks Filters Classifiers Clusterers Associations Evaluation Visualization

Arff Loader C45 Loader CSV Loader Database Loader LibSVM Loader Serialized InstancesLoader

Knowledge Flow Layout  
CSVLoader C45Loader Add Classification Attribute Selection BayesNet CrossValidation FoldMaker

Weka GUI Chooser

Program Visualization Tools Help



**WEKA**  
The University of Waikato

Waikato Environment for Knowledge Analysis  
Version 3.6.6  
(c) 1999 - 2011  
The University of Waikato  
Hamilton, New Zealand

Applications  
Explorer  
Experimenter  
KnowledgeFlow  
Simple CLI

IA 2017 – 2018

ISIA – FDI – UCM

Weka - 3

# Conceptos Básicos para utilizar WEKA

- **Instancia** : es un ejemplo definido con atributos
  - Tipos: Nominal, numeric, string o fecha/hora
- **Dataset**: Fichero con ejemplos
  - Fichero de entrenamiento o Fichero de test o de ambos
- **ARFF**: es el formato habitual de estos ficheros
  - Una cabecera describiendo los atributos
  - Cada ejemplo con sus atributos separados por comas.
  - También se pueden leer de una URL
    - también de una BD (resultado de una consulta SQL) usando JDBC
- **Filtros**: Para preprocesar los datos del dataset antes de usar
  - Sobre ejemplos: quitar, reordenar
  - Sobre atributos: discretización, normalización, quitar, añadir, transformar
  - Pueden ser supervisados y sin supervisar
  - Muchos de los algoritmos los usan internamente
- **Clasificador**: Aplicar un algoritmo de los usados en Weka a un dataset
  - Con unos parámetros escogidos

# Componentes de un Experimento

- Objetivo de los experimentos:
  - Entrenar con ejemplos un algoritmo, validarlo (test) con otros ejemplos
  - para que clasifique ejemplos desconocidos (ej.: ID3)
  - o sacar ciertas conclusiones de los resultados (ej.: Clustering)
- Weka se usa para hacer varias ejecuciones y estudiar los resultados
  - Cambiando los ejemplos, los atributos, los algoritmos y sus parámetros
- Cada ejecución es un **experimento** que consta de
  - Dataset con los datos de ejemplos o instancias a procesar
  - Filtros para preparar los datos del dataset
  - Clasificador
    - Algoritmo
    - Parámetros del algoritmo
  - Opciones de Ejecución
  - Opciones de salida: resultados y datos auxiliares
- Para preparar un experimento se necesitan varios pasos

# PASO 0 para Experimentar

- PASO 0: Conseguir ejemplos, etiquetarlos, formatear en ARFF
  - El arff es un formato de fichero texto, sin embargo no es muy popular
  - El csv (comma separated values) suele ser muy frecuente.
  - Puedes crearte un conversor (es sencillo) o buscar alguno ya programado
  - Aquí hay un conversor online que podría ayudarte:
    - <http://ikuz.eu/csv2arff/>

# Formato de archivo ARFF: Ejemplo jugar-tenis

@relation tiempo

@attribute pronostico {soleado, nublado, lluvioso}

@attribute temperatura real

@attribute humedad real

@attribute viento {SI, NO}

@attribute jugar-tenis {si, no}

@data

soleado,85,85,NO,no

soleado,80,90,SI,no

nublado,83,86,NO,si

lluvioso,70,96,NO,si

lluvioso,68,80,NO,si

lluvioso,65,70,SI,no

nublado,64,65,SI,si

soleado,72,95,NO,no

soleado,69,70,NO,si

lluvioso,75,80,NO,si

soleado,75,70,SI,si

nublado,72,90,SI,si

nublado,81,75,NO,si

lluvioso,71,91,SI,no

Nombre que se asigna al conjunto de datos

Atributo nominal

Atributo numérico

El último atributo es sobre el que se construye el clasificador. Es el atributo que queremos predecir, la "clase".

# PASO 1 para Experimentar

- PASO 1: Preparar Datos: Explorer + solapa “Preprocess”
  - ① Open file: Cargar el dataset con los ejemplos (formato ARFF)
  - ② Escoger Filtro: “Filter” + “Choose” + nombre
    - Dentro de la lista: click para seleccionar uno, ej.: Remove
      - Otro click sobre el nombre del filtro ya escogido: dar parámetros  
Ej.: attributeIndices, ...
    - Se puede salvar el filtro, o abrir otro filtro : “save”, “open”
    - Hay explicación de cada filtro pulsando en: “more”



## PASO 1: Explorer, solapa "Preprocess"

**Weka Explorer**

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open fil... | Open U... | Open DB... | Generat... | Undo | Edit... | Save...

Filter: Choose None Apply

Current relation: Relation: tiempo Instances: 14 Attributes: 5

Attributes: All None Invert Pattern

No.	Name
1	<input checked="" type="checkbox"/> pronostico
2	<input type="checkbox"/> temperatura
3	<input type="checkbox"/> humedad
4	<input type="checkbox"/> viento
5	<input type="checkbox"/> jugar-tenis

Remove

Selected attribute: Name: pronostico Type: Nomi... Missing: 0 (...) Distinct: ... Unique: 0 (0%)

No.	Label	Count
1	soleado	5
2	nublado	4
3	lluvioso	5

Class: jugar-tenis (Nom) Visualize All

ver todas las relaciones entre ejemplos

Status: OK Log x 0

# solapa “Preprocess” FILTRO Remove

Quitar Temperatura, es el atrib"2"

Hay muchos filtros

Info sobre ese atributo

Preprocess

Classify

Class

Open fil...

Open U...

Open DB...

Filter

Choose

Remove

Current relation

Relation: tiempo

Instances: 14

Attributes: 5

Attributes

All

None

Invert

Pattern

No.	Name
1	pronostico
2	temperatura
3	humedad
4	viento
5	jugar-tenis

Remove

Status

OK

weka.filters.unsupervised.attribute.Remove

About

A filter that removes a range of attributes from the dataset.

attributeIndices2

invertSelectionFalse

Open...Save...OK

Name: temperatura

Missing: 0 (...)

Distinct: ...

Statistic	Value
Minimum	64
Maximum	85
Mean	73.57
StdDev	6.572

Class: jugar-tenis (Nom)

8

6

64

74.5

85

Obfuscate

PartitionedMultiFilter

PKIDiscretize

PrincipalComponents

PropositionalToMultiInstance

RandomProjection

RandomSubset

RELAGGS

Remove

RemoveType

RemoveUseless

Reorder

ReplaceMissingValues

Standardize

StringToNominal

StringToWordVector

SwapValues

TimeSeriesDelta

TimeSeriesTranslate

Wavelet

instance

Filter...

Remove filter

Close

## PASOS 2 para Experimentar

- PASO 2: Construir un clasificador: Explorer + solapa “Classify”
  - ③ Escoger Algoritmo: “Classifier” + “Choose” + nombre de la lista
  - ④ Escoger parámetros para ese algoritmo: click en el nombre del “classifier”
    - Se abre ventana con parámetros

## PASO 2: construir un clasificador, cual?

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 % 65

More options...

(Nom) jugar-tenis

Start Stop

Result list (right-click for options)

Status OK

weka

- classifiers
- bayes
- functions
- lazy
- meta
- mi
- misc
- rules
- trees
  - ADTree
  - BFTree
  - DecisionStump
  - FT
  - Id3
  - J48**
  - J48graft
  - LADTree
  - LMT
  - MSP
  - NBTree

(ID3 es sólo para valores nominales)  
J48 es una implementación de C4.5

Filter... Remove filter Close

# PASO 2 parámetros

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. The 'Classifier' section displays 'J48 -C 0.25 -M 2'. A red circle with the number '4' is next to the 'Test options' section, which includes radio buttons for 'Use training set', 'Supplied test set', 'Cross-validation' (selected), and 'Percentage split'. The 'Cross-validation' section shows 'Folds' set to 10. Below this is a 'More options...' button. The 'Class' section shows '(Nom) jugar-tenis'. The 'Status' section shows 'OK'.

The 'weka.classifiers.trees.J48' settings dialog is open, showing various parameters:

- binarySplits: False
- confidenceFactor: 0.25
- debug: False
- minNumObj: 2
- numFolds: 3
- reducedErrorPruning: False
- saveInstanceData: False
- seed: 1
- subtreeRaising: True
- unpruned: False
- useLaplace: False

Buttons at the bottom of the dialog include 'Open...', 'Save...', 'OK', and 'Cancel'.

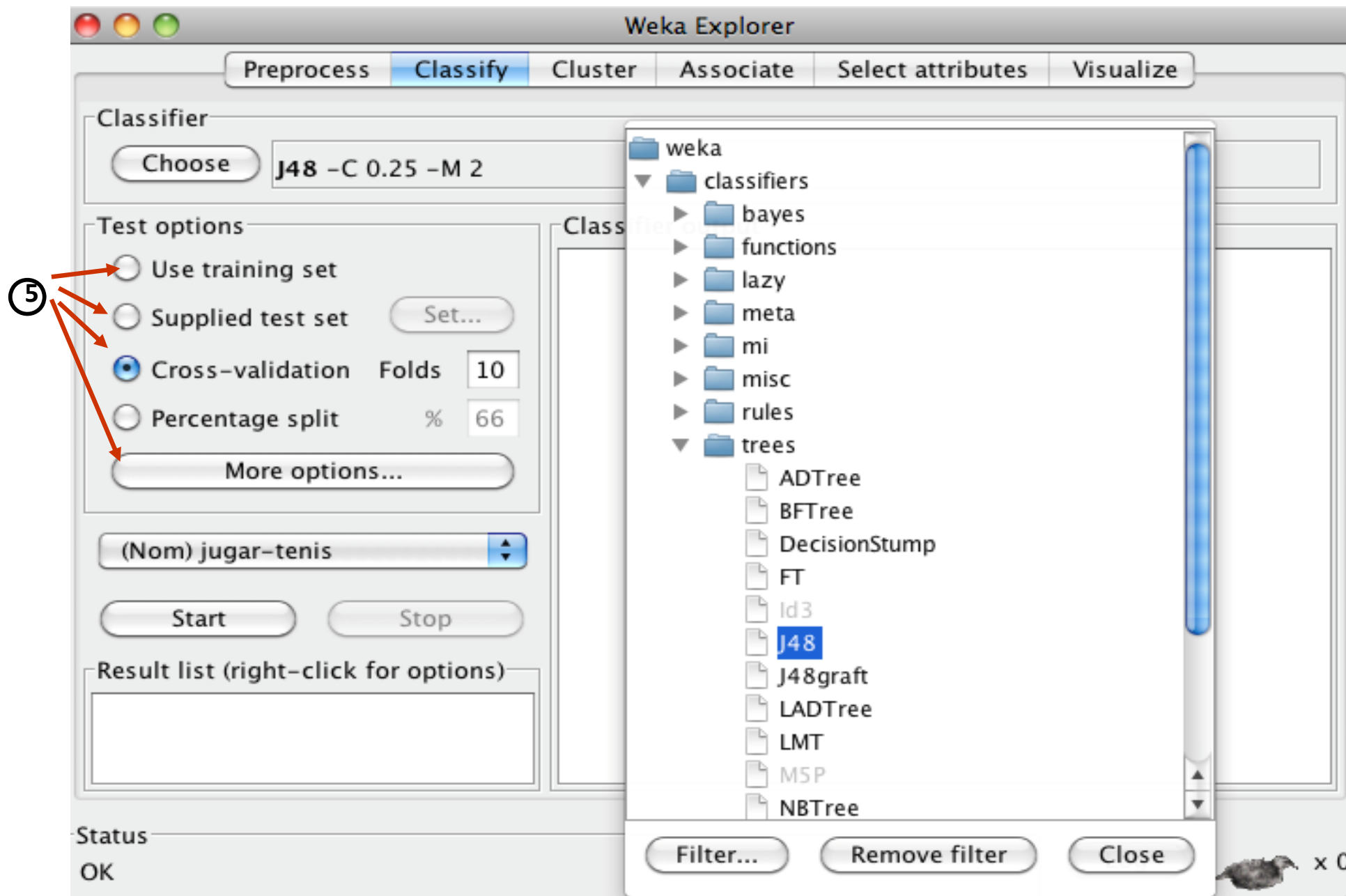
# PASO 3 para Experimentar

## ● PASO 3: Preparar Experimento

### ⑤ Definir las opciones:

- “Test options” para Validar el Clasificador
  - Uso del conjunto de entrenamiento (todo el dataset)
  - Conjunto de test con datos distintos (pide otro fichero)
  - cross-validation
    - » Se divide el dataset en 10 Folds
      - Se usan 9 partes para entrenar y 1 parte como test
      - Se repite el experimento 10 veces con las distintas combinaciones
      - Resultados medio
  - porcentaje de entrenamiento y resto de test (66%)
    - » 1 experimento, 2 tercios para entrenar, un tercio como test
- “More options” :
  - Qué datos de salida: modelo, matriz confusión, predicciones
  - Preservar el orden en particiones, **sacar el código fuente**

## PASO 3 opciones



## PASO 4 para Experimentar

### ● PASO 4: Ejecutarlo y ver resultados

⑥ Botón “Start” . Se puede parar si lleva demasiado tiempo

⑦ Resultados: en la “Result list” se almacenan todos los experimentos

Click dcho y se abre ventana con opciones sobre:

- Resultados: Ver, Salvar, Borrar
- Modelo: salvar, cargar (el clasificador entrenado después del experimento)
- Visualizar: árbol, errores clasificación, curvas (para interpretar resultado)



# PASO 4 ver resultados

**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...

**Classifier output**

=== 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
	0.778	0.6	0.7	0.778	0.737	0.789
	0.4	0.222	0.5	0.4	0.444	0.789
Weighted Avg.	0.643	0.465	0.629	0.643	0.632	0.789

=== Confusion Matrix ===

```
a b  <-- classified as
7 2  | a = si
3 2  | b = no
```

**Result list (right-click for options)**

- 23:15:16 - trees.J48
- 01:17:28 - trees.J48
- 01:17:49 - trees.J48

Annotations:  
- circled 6 points to the Start button  
- arrow labeled 'clase' points to the '(Nom) jugar-tenis' dropdown  
- arrow points to the 'Result list' with text 'Resultados de distintos clasificadores o experimentos'

# PASO 4 ver resultados

Preprocess

Classif

Classifier

Choose J48 -C 0.25 -M 2

Test options

Use training set

Supplied test set

Cross-validation

Percentage split

Folds 10

% 66

More options

(Nom) jugar-tenis

Start

Result list (right-click for context menu)

23:15:16 - trees.J48

01:17:28 - trees.J48

01:17:49 - trees.J48

Status

OK

Weka Classifier Tree Visualizer: 22:58:38 - trees.J48 (tiempo)

Tree View

pronostico

= soleado

= nublado

= lluvioso

humedad

si (2.0)

no (3.0)

si (4.0)

viento

= SI

= NO

no (2.0)

si (3.0)

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

Log

x 0

7 Botón Dcho

# Resultados : Información de la Ejecución

## === Run information ===

**Scheme:** weka.classifiers.trees.J48 -C 0.25 -M 2

**Relation:** tiempo

**Instances:** 14

**Attributes:** 5

pronostico

temperatura

humedad

viento

jugar-tenis

**Test mode:** evaluate on training data

## === Classifier model (full training set) ===

**J48 pruned tree**

```

-----
pronostico = soleado
| humedad <= 75: si (2.0)
| humedad > 75: no (3.0)
pronostico = nublado: si (4.0)
pronostico = lluvioso
| viento = SI: no (2.0)
| viento = NO: si (3.0)
  
```

**Number of Leaves :** 5

**Size of the tree :** 8

## === Stratified cross-validation ===

# PASO 5: Validar interpretando los datos

- PASO 5: interpretar, validar el clasificador

- ⑧ Errores

- ⑨ Estadísticas (explicadas más adelante)

- TP Rate FP Rate Precision Recall F-Measure ROC Area Class**

- ⑩ Matriz de confusión

- PASO 6: Preparar otro experimento para comparar

## PASO 5: Validar , Matriz confusión, estadísticas

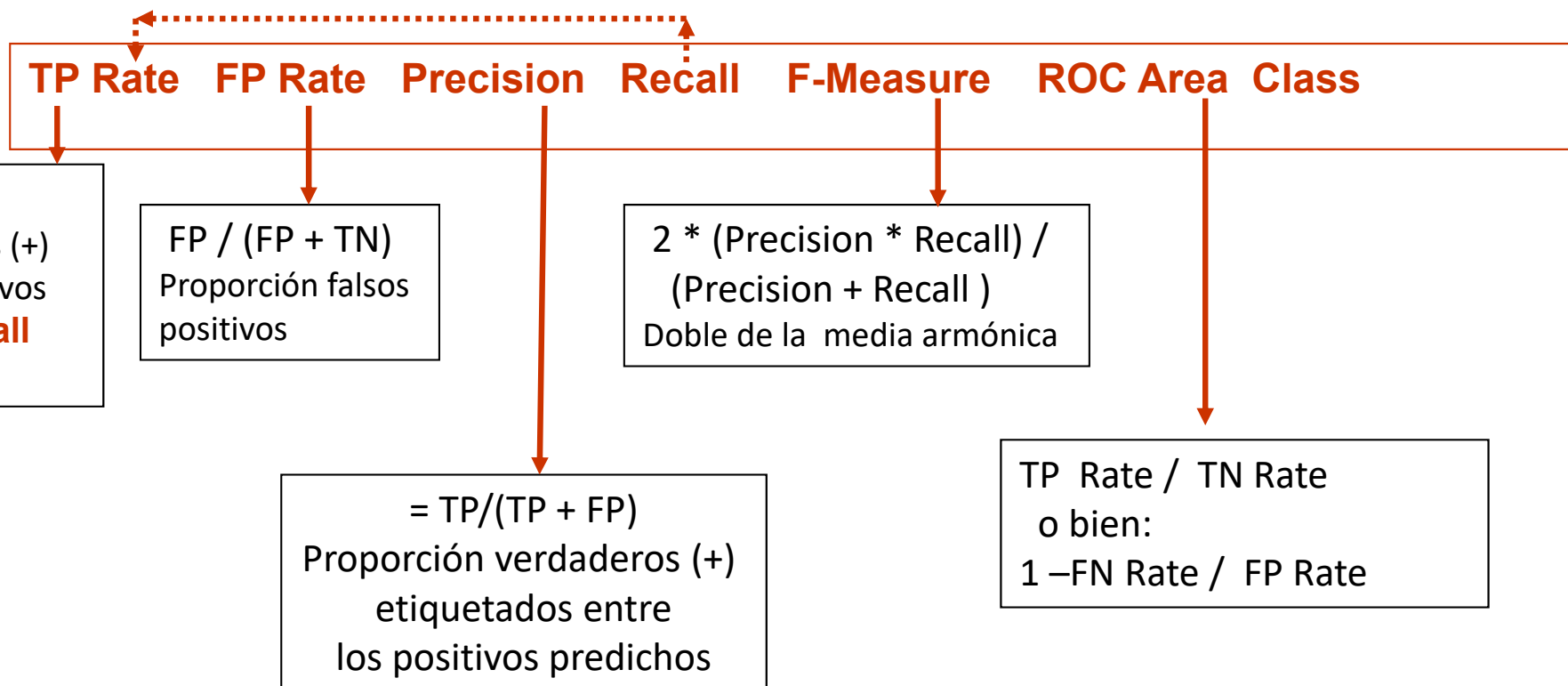
		Predichos	
		0 (+)	1 (-)
Etiquetados	0 (+)	TP	FN
	1 (-)	FP	TN

TP : True + Verdaderos positivos

TN : True - Verdaderos negativos

FP : False + Falsos positivos

FN : False - Falsos negativos



→ Precision, Recall y F-Measure cuanto más cerca de 1 mejor

→ Para cada problema hay ciertas medidas más importantes

# PASO 5: Validar el Clasificador

=== Stratified **cross-validation** ===

=== Summary ===

⑧

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 ===

⑨

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.778	0.6	0.7	0.778	0.737	0.789	si
	0.4	0.222	0.5	0.4	0.444	0.789	no
Weighted Avg.	0.643	0.465	0.629	0.643	0.632	0.789	

=== **Confusion Matrix** ===

⑩

a b <-- classified as

7	2	a = si
3	2	b = no

si => 7 TP, 3 FP, 2 FN, 2 TN  
no => 2 TP, 2 FP, 3 FN, 7 TN

Recall = TP Rate =  $TP / (TP + FN)$

Precision =  $TP / (TP + FP)$

FP Rate =  $FP / (FP + TN)$

## Otro ejemplo con varias clases (zoo)

- ❑ Zoo database (zoo.arff)
  - ❑ 101 instances, 7 types
  - ❑ Number of Attributes: 18 (animal name, 15 Boolean attributes, 2 numerics)
  - ❑ animal name:Unique for each instance, hair:Boolean, feathers:Boolean, eggs:Boolean, milk:Boolean, airborne:Boolean, aquatic:Boolean, predator:Boolean, toothed:Boolean, backbone:Boolean, breathes:Boolean, venomous:Boolean, fins:Boolean, legs: Numeric (set of values: {0,2,4,5,6,8}), tail:Boolean, domestic:Boolean, catsize:Boolean, type:{mammal, bird, ... }

## PASO 5: Zoo.arff ejemplo con varias clases

- Resultado de ejecutar J48 con la misma configuración que el experimento anterior

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	93	92.0792 %
Incorrectly Classified Instances	8	7.9208 %
Kappa statistic	0.8955	
Mean absolute error	0.0225	
Root mean squared error	0.14	
Relative absolute error	10.2478 %	
Root relative squared error	42.4398 %	
Coverage of cases (0.95 level)	96.0396 %	
Mean rel. region size (0.95 level)	15.4173 %	
Total Number of Instances	101	



## PASO 5: Zoo.arff ejemplo con varias clases

=== Confusion Matrix ===

a	b	c	d	e	f	g	<--	classified as		
41	0	0	0	0	0	0		a	=	mammal
0	20	0	0	0	0	0		b	=	bird
0	0	3	1	0	1	0		c	=	reptile
0	0	0	13	0	0	0		d	=	fish
0	0	1	0	3	0	0		e	=	amphibian
0	0	0	0	0	5	3		f	=	insect
0	0	0	0	0	2	8		g	=	invertebrate

→ Reptiles c :

TP = 3 (diagonal)

FP = 1 (resto vertical)

FN = 2 (resto horizontal)

TN = Todos-TP-FP-FN=95

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
1	0	1	1	1	1	mammal
1	0	1	1	1	1	bird
0.6	0.01	0.75	0.6	0.667	0.793	reptile
1	0.011	0.929	1	0.963	0.994	fish
0.75	0	1	0.75	0.857	0.872	amphibian
0.625	0.032	0.625	0.625	0.625	0.92	insect
0.8	0.033	0.727	0.8	0.762	0.986	invertebrate
0.921	0.008	0.922	0.921	0.92	0.976	Weig.Avg.

→ Reptiles c :

TP Rate =  $3/5 = 0,6$ FP Rate =  $1/96 = 0,01$ Precision =  $3/4 = 0,75$ 

Recall=TP Rate

# Otros algoritmos: Clustering Jerárquico

- Solapa “cluster”
- “Choose” selecciona el algoritmo, HierarchicalClusterer
- Parámetros de ese algoritmo: click en su nombre:
  - Función para la distancia entre instancias, ej.: EuclideanDistance
  - Máximo número de iteraciones antes de parar
  - Número de clusters que se quiere:
    - A) escoge número de clases que tenías
      - Así puede comparar si hay instancias demasiado parecidas=> clasifica mal
    - B) escoge 3: ver cómo agrupa en menos clusters que clases
- Opciones del experimento:
  - Classes to clusters evaluation
  - Validación: usar fichero de entrenamiento, o de test o % de entrenamiento
  - Ignorar atributos para que no entren en el algoritmo

# Ejemplo del zoo

-A-

Clustered Instances	
0	41 ( 41%)
1	13 ( 13%)
2	21 ( 21%)
3	17 ( 17%)
4	7 ( 7%)
5	1 ( 1%)
6	1 ( 1%)

Classes to Clusters:								
0	1	2	3	4	5	6	<--	assigned to cluster
41	0	0	0	0	0	0		mammal
0	0	20	0	0	0	0		bird
0	0	1	0	3	0	1		reptile
0	13	0	0	0	0	0		fish
0	0	0	0	4	0	0		amphibian
0	0	0	8	0	0	0		insect
0	0	0	9	0	1	0		invertebrate

Cluster 0 <--	mammal
Cluster 1 <--	fish
Cluster 2 <--	bird
Cluster 3 <--	insect
Cluster 4 <--	amphibian
Cluster 5 <--	invertebrate
Cluster 6 <--	reptile
Incorrectly clustered instances	
13.0	12.8713 %

-B-

Clustered Instances	
0	99 ( 98%)
1	1 ( 1%)
2	1 ( 1%)
Cluster 0 <-- mammal	
Cluster 1 <-- invertebrate	
Cluster 2 <-- reptile	
Incorrectly clustered instances : 58.0 57.4257 %	

Classes to Clusters:			
0	1	2	<-- assigned to cluster
41	0	0	mammal
20	0	0	bird
4	0	1	reptile
13	0	0	fish
4	0	0	amphibian
8	0	0	insect
9	1	0	invertebrate

# Enlaces sobre materiales de Weka

- Documentación en /weka-3-6-9/documentation.html
  - WekaManual.pdf
  - Home <http://www.cs.waikato.ac.nz/ml/weka/>
  - WekaWiki (HOWTOs, code snippets, etc.) <http://weka.wikispaces.com/>
  - weka desde java : <http://weka.wikispaces.com/Use+WEKA+in+your+Java+code>
- [Data mining \[Recurso electrónico\] : practical machine learning tools and techniques / Ian H. Witten,](#)  
Burlington, MA : Morgan Kaufmann Publishers, c2011  
Ubicación: Bca.Digital Complutense
- Manual español: info en detalle
  - <http://metaemotion.com/diego.garcia.morate/download/weka.pdf>
- Tutorial en español
  - <http://isa.umh.es/asignaturas/crss/tutorialWEKA.pdf>
- Medidas de resultados (matemáticas y estadística básicas)
  - <http://web.engr.oregonstate.edu/~tgd/classes/534/slides/part13.pdf>