

WEKA



BIOINFORMATICS AND BIG DATA ANALYTICS

Ing. Antonio Brunetti
Prof. Vitoantonio Bevilacqua

Indice

Cosa è weka

- Tecnologie

Hands On Weka

- Weka Explorer
 - KnowledgeFlow /Simple CLI
- Caricare il dataset
- Il file arff
- Integrazione e valutazione dei dati
- Classificatori
 - Multilayer Perceptron
 - Random Forest
 - J48
- Matrice di Confusione e interpretazione dei risultati
- Esportazione delle reti addestrate
- Esempi pratici

Cosa è WEKA



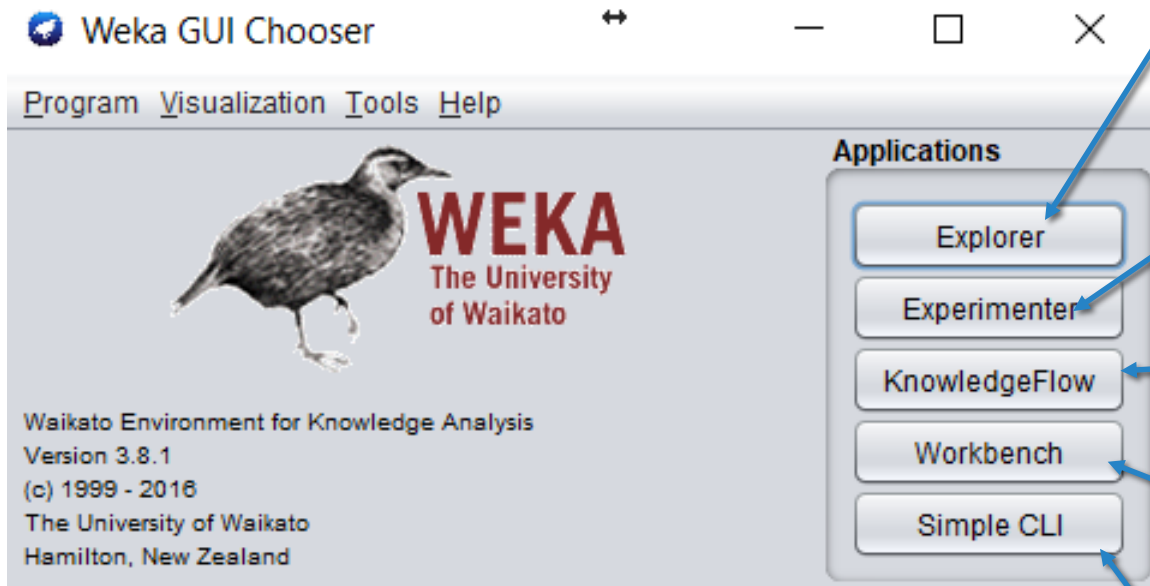
- Sviluppato dall'Università Waikato in Nuova Zelanda.
- Acronimo per **W**aikato **E**nvironment for **K**nowledge **A**nalysis.
- È una collezione di algoritmi e tecniche di apprendimento automatico (**machine learning**).
- È stato disegnato per testare velocemente metodologie esistenti su insiemi di dati diversi, in modo flessibile.
- Set completo di strumenti per il pre-processing, algoritmi di apprendimento e metodi di valutazione.
- Disponibile gratuitamente all'indirizzo <http://www.cs.waikato.ac.nz/ml/weka/>

Tecnologie

- Weka è realizzato in Java
- Implementa un vastissimo numero di algoritmi di classificazione e clustering
 - Reti Bayesiane
 - Funzioni di regressione e EBP
 - Classificatori basati su regole
 - Classificatori basati su alberi
- È possibile utilizzare in applicativi JAVA i motori di Weka ed importare direttamente i classificatori addestrati



Hands on Weka



EXPLORER

Funzionalità completa di clustering e classificazione

EXPERIMENTER

Permette un approccio di aggregazione dati per cercare l'algoritmo più corretto

KnowledgeFlow

Possibilità di modellare la classificazione in un ambiente simulink-like

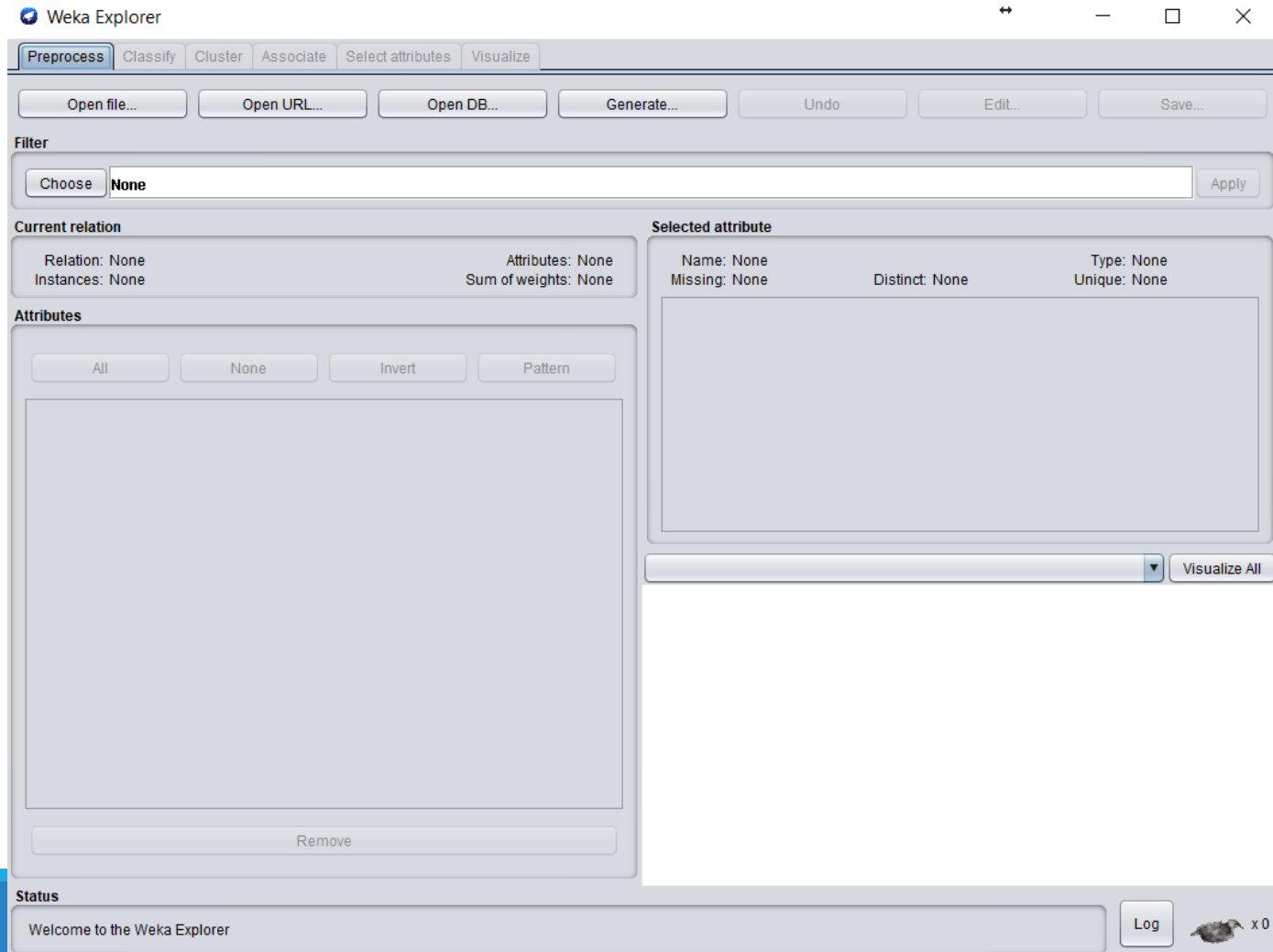
Wokbench

Integra tutti i controlli Explorer, KnowledgeFlow E CLI

Simple CLI

Controllo a riga di comando

Weka Explorer



Il file ARFF

È un file testuale composto da due sezioni

- Header
- Dataset

Nella dichiarazione vengono presentati e categorizzati i dati che saranno poi presenti nella sezione @DATA



iris.arff

```
@RELATION iris
@ATTRIBUTE sepallength    REAL
@ATTRIBUTE sepalwidth    REAL
@ATTRIBUTE petallength    REAL
@ATTRIBUTE petalwidth    REAL
@ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica}

@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
[...]
```

Weka Explorer

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

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

Filter
Choose **None** Apply

Current relation
Relation: iris | Instances: 150 | Attributes: 5 | Sum of weights: 150

Attributes
All | None | Invert | Pattern

No.	Name
1	<input type="checkbox"/> sepallength
2	<input checked="" type="checkbox"/> sepalwidth
3	<input type="checkbox"/> petallength
4	<input type="checkbox"/> petalwidth
5	<input type="checkbox"/> class

Remove

Selected attribute
Name: sepalwidth | Type: Numeric
Missing: 0 (0%) | Distinct: 23 | Unique: 5 (3%)

Statistic	Value
Minimum	2
Maximum	4.4
Mean	3.054
StdDev	0.434

Class: class (Nom) Visualize All

Bin Range	Count
2.0 - 2.2	8
2.2 - 2.4	16
2.4 - 2.6	33
2.6 - 2.8	51
2.8 - 3.0	24
3.0 - 3.2	12
3.2 - 3.4	4
3.4 - 3.6	2

Status
OK | Log x 0

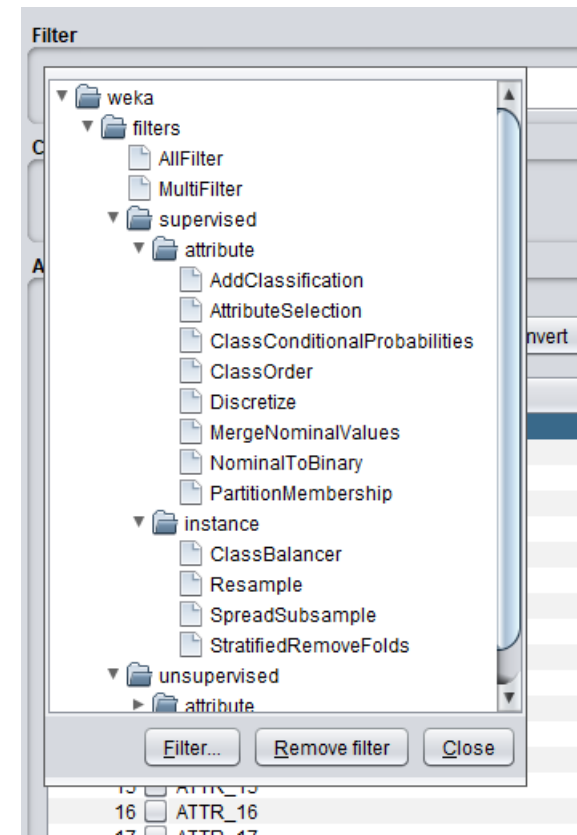
Integrazione e valutazione del dataset

Le funzioni di gestione dei dati (filtri) possono essere

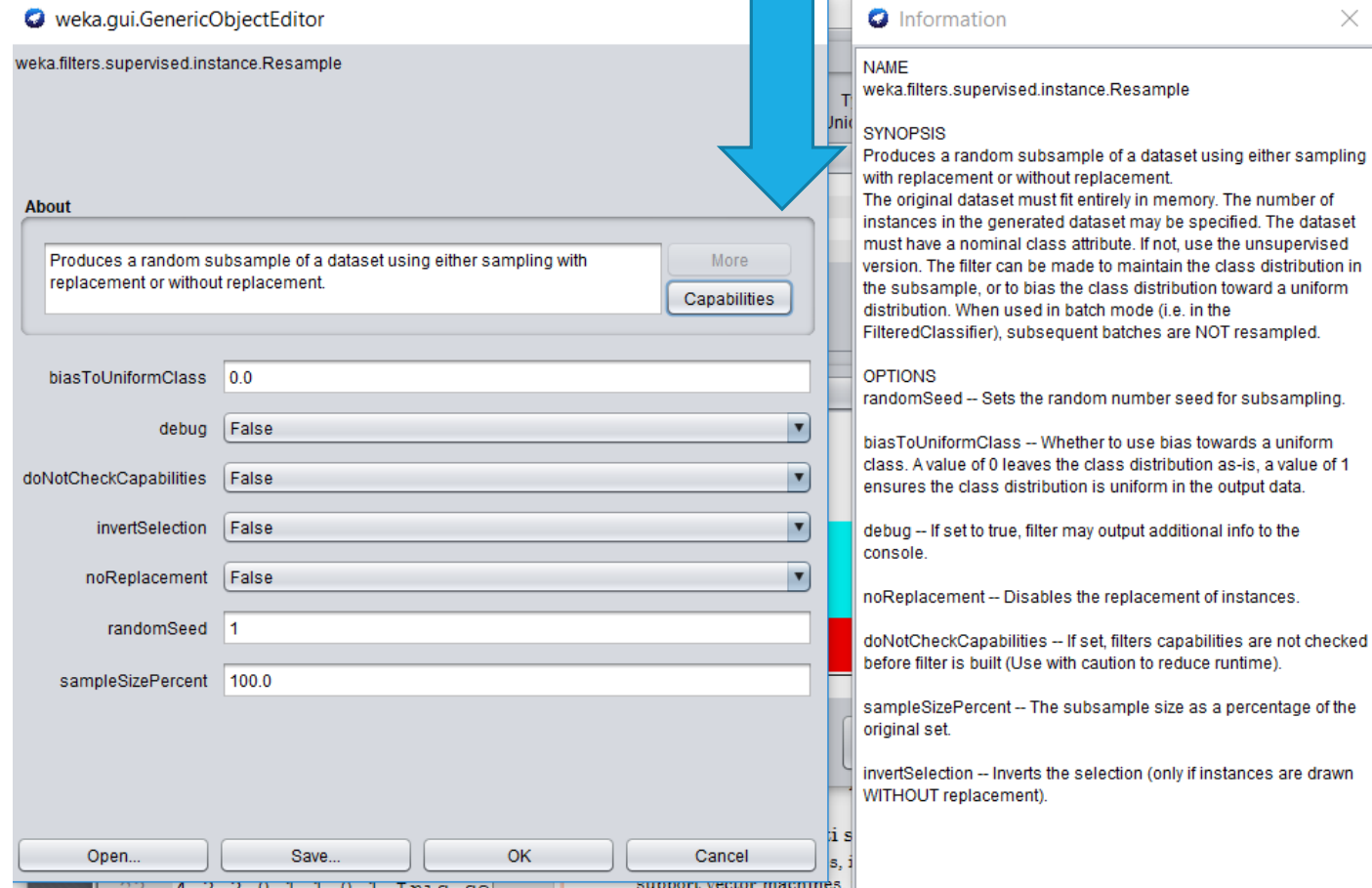
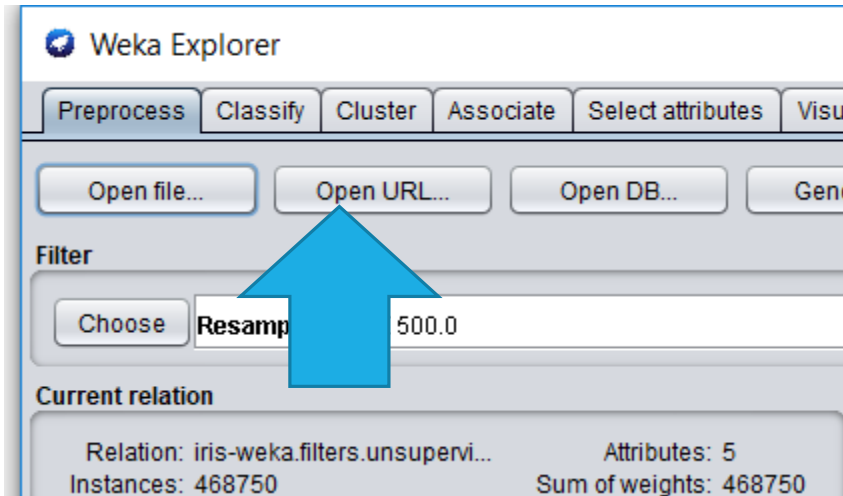
- Supervised: usano informazioni sulla classe per determinare gli attributi/istanze per effettuare l'operazione
- Unsupervised: non usano le informazioni sulla classe

Inoltre, si distingue tra:

- Attribute filters: effettua un'operazione di attributo
- Instance filters: effettua un'operazione sulle istanze di un determinato attributo



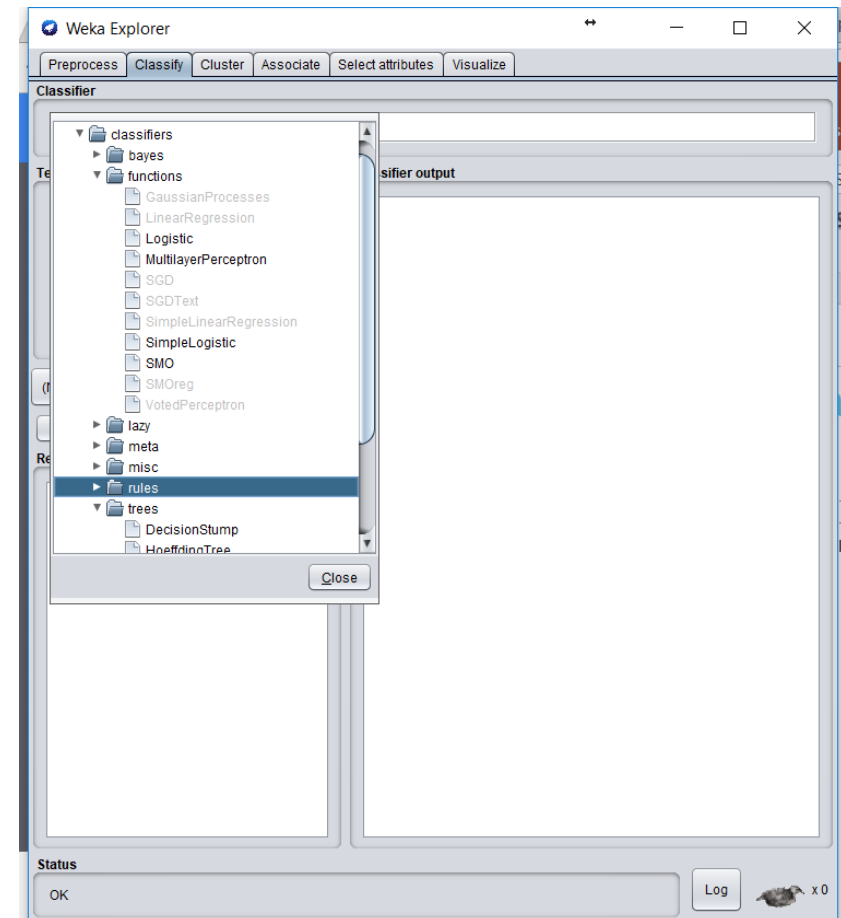
Informazioni dettagliate



Classify

Contiene i metodi per effettuare operazioni di classificazione

- Test Options:
 - Permette di specificare le opzioni per il training e il test del classificatore
 - Use training set
 - Supplied test set
 - Cross validation
 - Percentage Split
 - Specificare l'attributo da usare come classe



Output di Classificazione

More Options

Result List

- Show Tree
- Roc Curve
- Model Export

Classifier output

The screenshot displays the Weka Explorer interface. The 'Classifier' tab is active, showing a 'RandomForest' classifier with parameters: -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1. The 'Test options' section is set to 'Cross-validation' with 10 folds. A 'Classifier evaluation options' dialog box is open, showing the following settings:

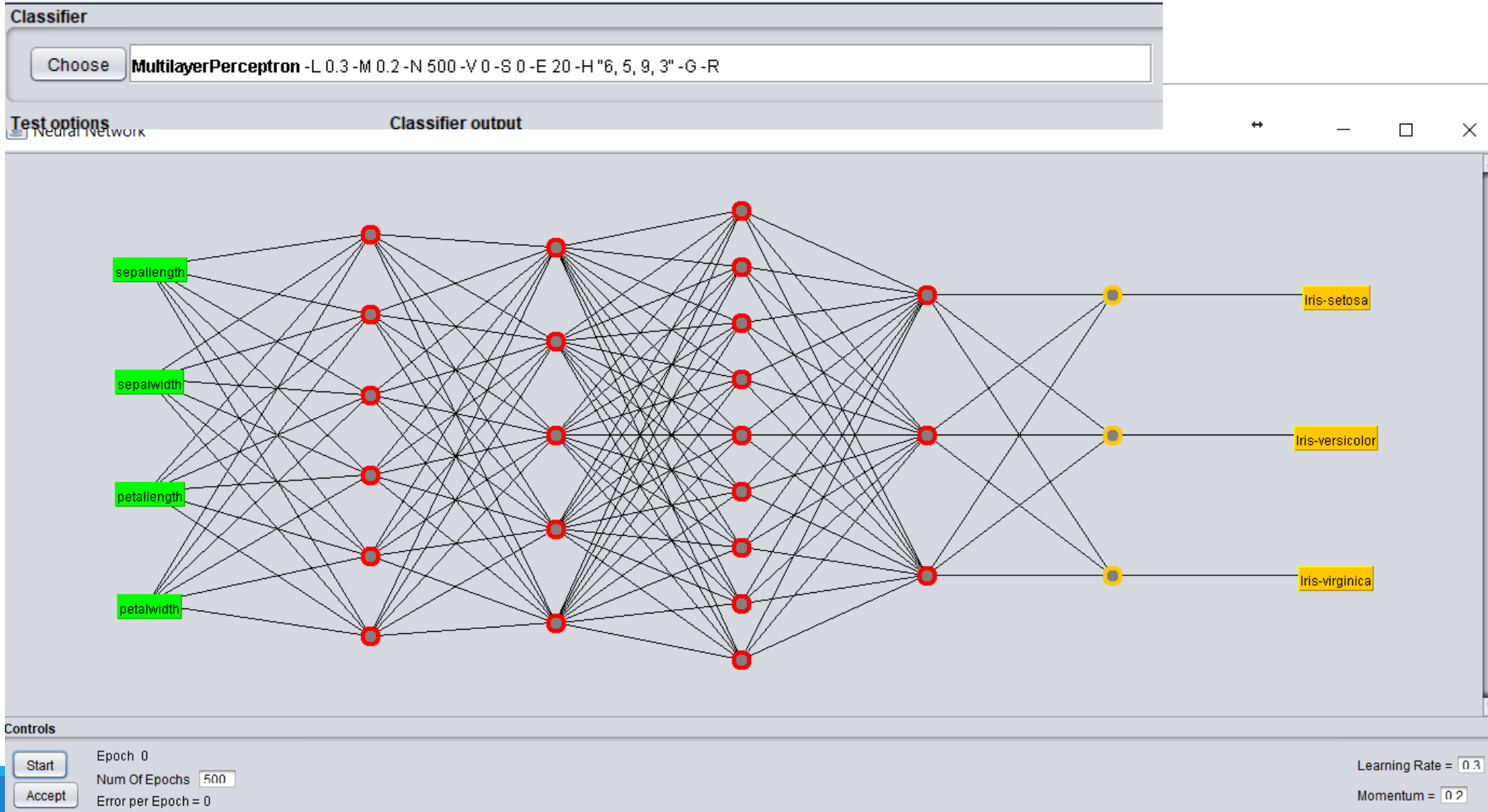
- Output model
- Output per-class stats
- Output entropy evaluation measures
- Output confusion matrix
- Store predictions for visualization
- Error plot point size proportional to margin
- Output predictions: Choose **Null**
- Cost-sensitive evaluation (Set...)
- Random seed for XVal / % Split: **1**
- Preserve order for % Split
- Output source code: WekaClassifier

The 'Result list (right-click for options)' shows a list of classifiers and their execution times. The 'RandomForest' classifier is highlighted:

- 16:51:32 - rules.ZeroR
- 16:51:40 - functions.MultilayerPerceptron
- 16:51:48 - functions.MultilayerPerceptron
- 16:52:07 - trees.RandomForest
- 17:13:30 - functions.MultilayerPerceptron
- 17:13:43 - functions.MultilayerPerceptron
- 17:14:00 - functions.MultilayerPerceptron
- 17:14:41 - functions.MultilayerPerceptron
- 17:21:14 - trees.J48
- 17:24:52 - trees.RandomForest

The 'Status' bar at the bottom shows 'OK' and a 'Log' button.

Esempio di Multilayer Perceptron



Impostazioni Principali

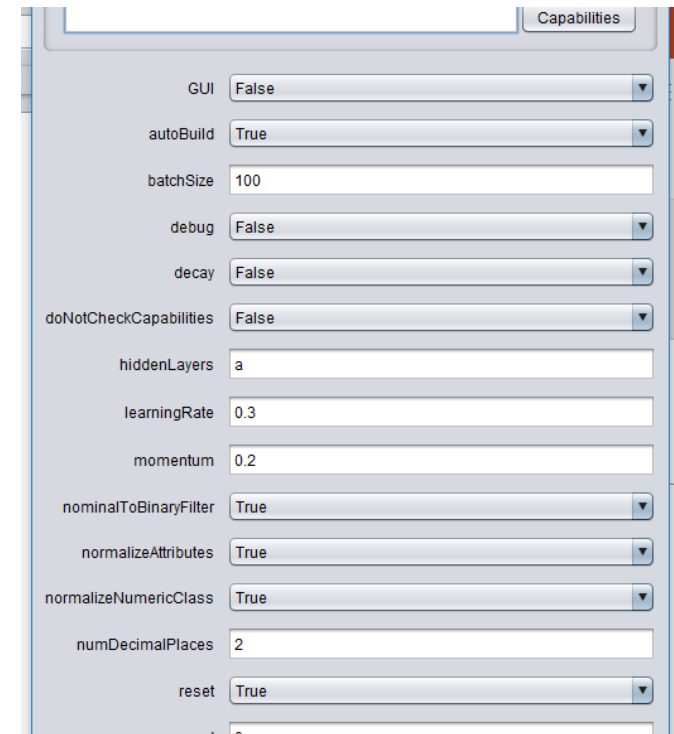
GUI: Mostra il layout della rete e permette di modificarlo

Learning Rate: di quanto vengono aggiornati i pesi, accelera il processo inizialmente ma rende più difficile il fine tuning

Hidden layers, numero di neuroni per layer, implementa la topologia della rete Es: 6,7,3,2

Seed: Configurazione iniziale dei pesi

Training Time: Numero di epoche di addestramento prima che l'algoritmo si fermi



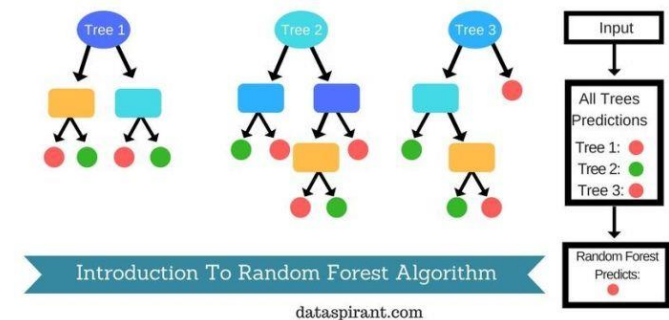
The image shows a configuration window titled "Capabilities" with the following settings:

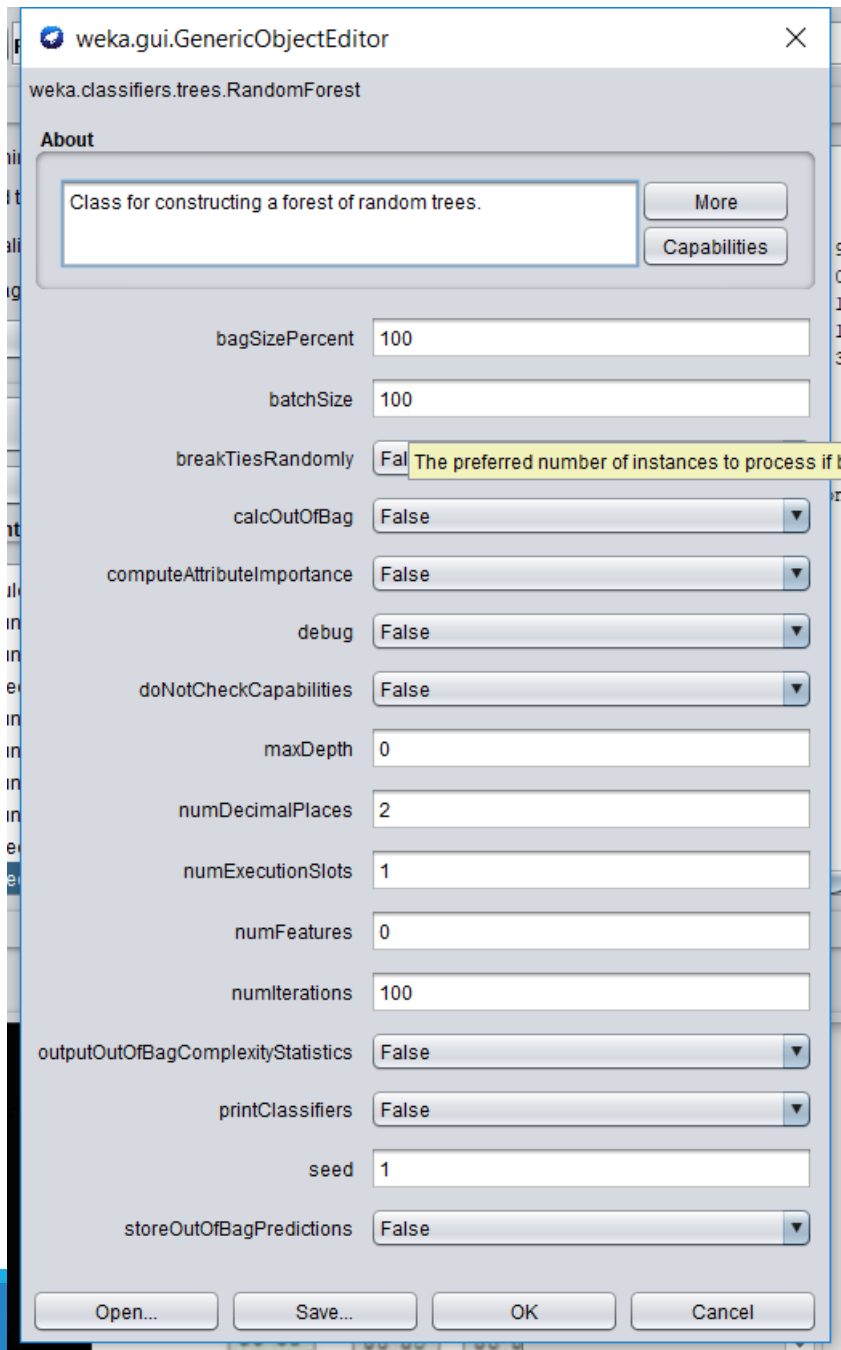
Property	Value
GUI	False
autoBuild	True
batchSize	100
debug	False
decay	False
doNotCheckCapabilities	False
hiddenLayers	a
learningRate	0.3
momentum	0.2
nominalToBinaryFilter	True
normalizeAttributes	True
normalizeNumericClass	True
numDecimalPlaces	2
reset	True
seed	0

Random Forest

Le random forest sono uno strumento di classificazione introdotto per la prima volta nel 2001 da Leo Breiman, nate da lavori precedenti dello stesso autore volti a migliorare le prestazioni ottenibili da singoli alberi di decisione, introducendo alcune componenti aleatorie nella costruzione degli stessi e un meccanismo di voto per la determinazione del risultato della classificazione.

Oltre alla migliorata accuratezza, le foreste di decisione sono interessanti per la loro efficienza, dovuta al parallelismo sia in fase di costruzione sia in quella di classificazione.





Impostazioni Principali RF

batchSize – Numero di alberi nella foresta

Print Classifiers - mostra ogni singolo albero

maxDepth - Massima dimensione dell'albero

numFeatures – Numero di features utilizzate

J48

J48 è l'implementazione Java dell'algoritmo C4.5.

C4.5 è un algoritmo utilizzato per generare un albero decisionale sviluppato da Ross Quinlan, è un'estensione del precedente algoritmo ID3 di Quinlan .

Gli alberi decisionali generati dal C4.5 possono essere utilizzati per la classificazione, e per questo motivo, C4.5 viene spesso definito classificatore statistico

PRO:

- Espressivi quanto un Decision Tree
- Facili da interpretare
- Facili da generare
- Rapidi nella classificazione di nuove istanze

CONTRO:

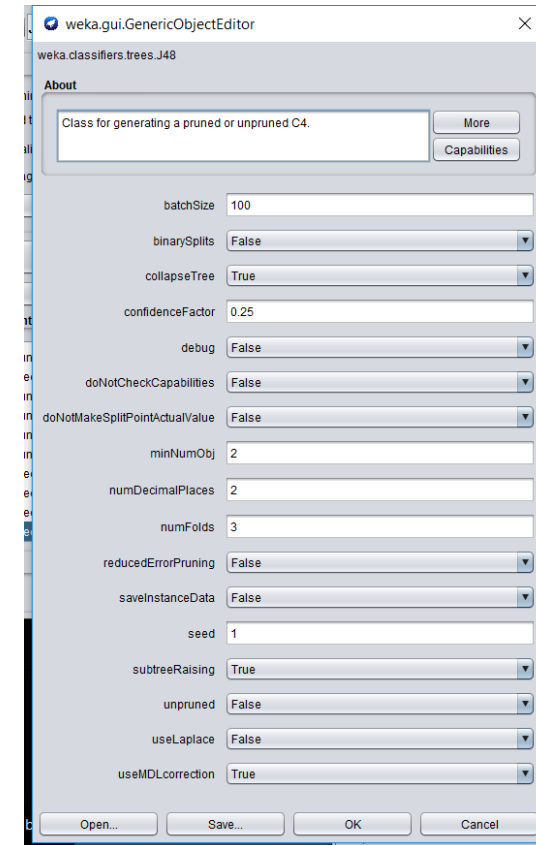
- Il costo di costruzione non scala al crescere del training set
- Risentono fortemente del rumore sui dati

Impostazioni Principali J48

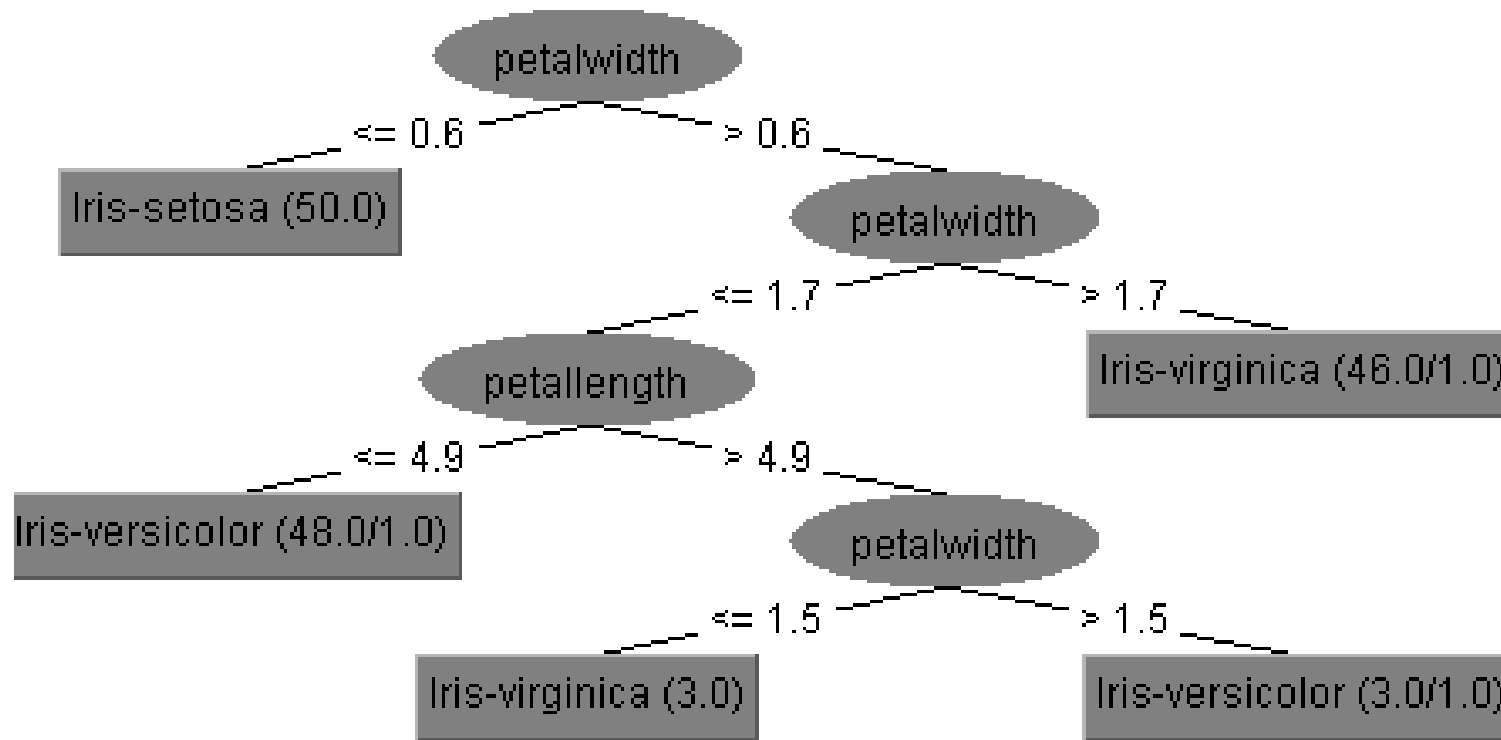
Il pruning è una attività fondamentale nella creazione dell'albero, molte impostazioni sono volte a migliorare questo aspetto

- Confidence Factor
- Num Folds
- reduceErrorPruning: utilizza un altro algoritmo rispetto a quello standard
- Unpruned

binarySplits – se false permette la creazione di split non binari



Visualizzazione dell'albero generato dal J48



Classifier Output

=== Run information ===

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

=== Stratified cross-validation ===

=== Summary ===

=== Detailed Accuracy By Class ===

=== Confusion Matrix ===

La schermata Classifier Output

=== Summary ===

Correctly Classified Instances	144	96 %
Incorrectly Classified Instances	6	4 %
Kappa statistic	0.94	
Mean absolute error	0.035	
Root mean squared error	0.1586	
Relative absolute error	7.8705 %	
Root relative squared error	33.6353 %	
Total Number of Instances	150	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,980	0,000	1,000	0,980	0,990	0,985	0,990	0,987	Iris-setosa
	0,940	0,030	0,940	0,940	0,940	0,910	0,952	0,880	Iris-versicolor
	0,960	0,030	0,941	0,960	0,950	0,925	0,961	0,905	Iris-virginica
Weighted Avg.	0,960	0,020	0,960	0,960	0,960	0,940	0,968	0,924	

=== Confusion Matrix ===

```
a b c <-- classified as
49 1 0 | a = Iris-setosa
0 47 3 | b = Iris-versicolor
0 2 48 | c = Iris-virginica
```

Confusion Matrix

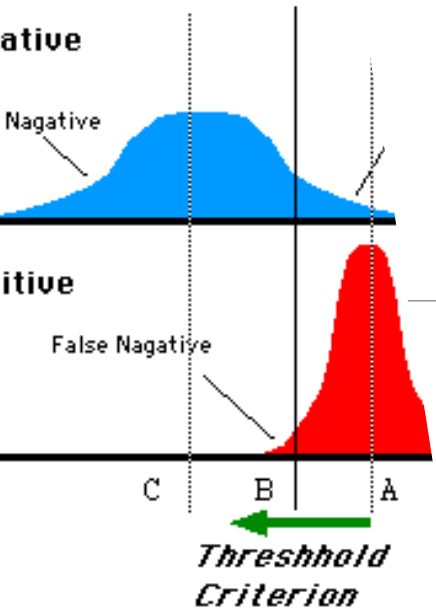
		Valori predetti		totale
		n'	p'	
Valori Reali	n	Veri negativi	Falsi positivi	N
	p	Falsi negativi	Veri positivi	
totale		N'	P'	P

a	b	c	<-- classified as
49	1	0	a = setosa
0	47	3	b = versicolor
0	2	48	c = virginica

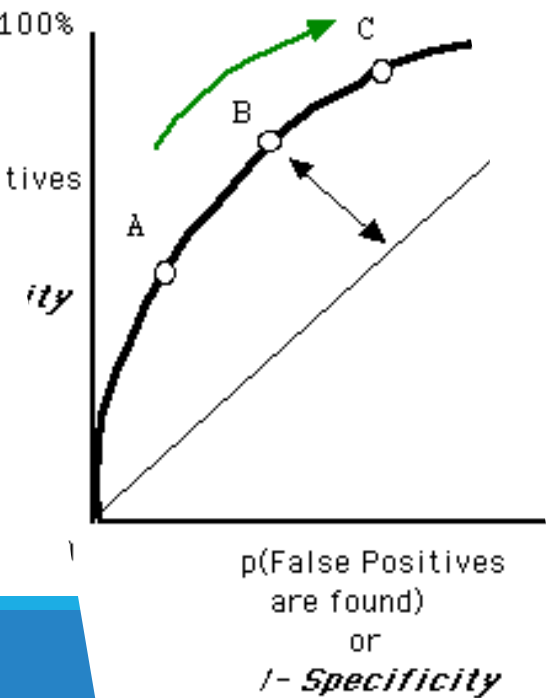
Indica che 3 campioni di Versicolor sono stati riconosciuti come Virginica

Indica che 2 campioni di virginica sono stati riconosciuti come versicolor

ons of the Observed si



Curve ROC



Test options

Use training set

Supplied test set Set...

Cross-validation Folds

Percentage split %

More options...

(Nom) class ▼

Start Stop

Result list (right-click for options)

- 18:49:02 - tree 149
- View in main window
- View in separate window
- Save result buffer
- Delete result buffer(s)
- Load model
- Save model
- Re-evaluate model on current test set
- Re-apply this model's configuration
- Visualize classifier errors
- Visualize tree
- Visualize margin curve
- Visualize threshold curve
- Cost/Benefit analysis
- Visualize cost curve

Classifier output

Size of the tree : 9

Time taken to build model: 0.0

=== Stratified cross-validated
=== Summary ===

Correctly Classified Instances

Incorrectly Classified Instance

Kappa statistic

Mean absolute error

Root mean squared error

Relative absolute error

Root relative squared error

Number of Instances

ailed Accuracy By Class

	TP Rate	FP F
	0,980	0,00
	0,940	0,03
	0,960	0,03
id Avg.	0,960	0,02

fusion Matrix ===

c <-- classified as

- setosa
- versicolor
- virginica

Weka Classifier Visualize: ThresholdCurve. (Class v...)

X: False Positive Rate (Num) Y: True Positive Rate (Num)

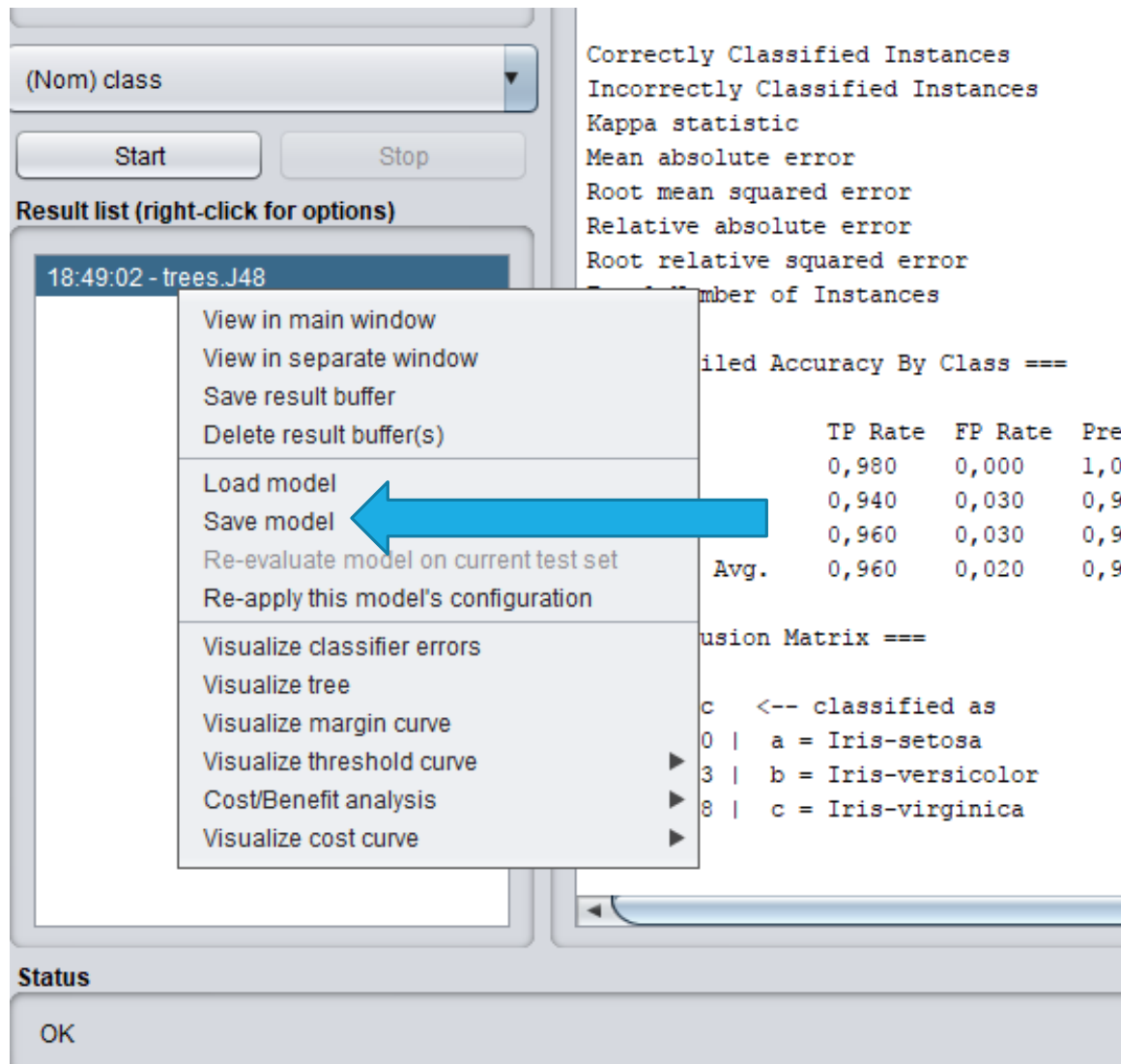
Colour: Threshold (Num) Select Instance

Reset Clear Open Save Jitter

Plot (Area under ROC = 0.9519)

Class colour

0 0.5 1

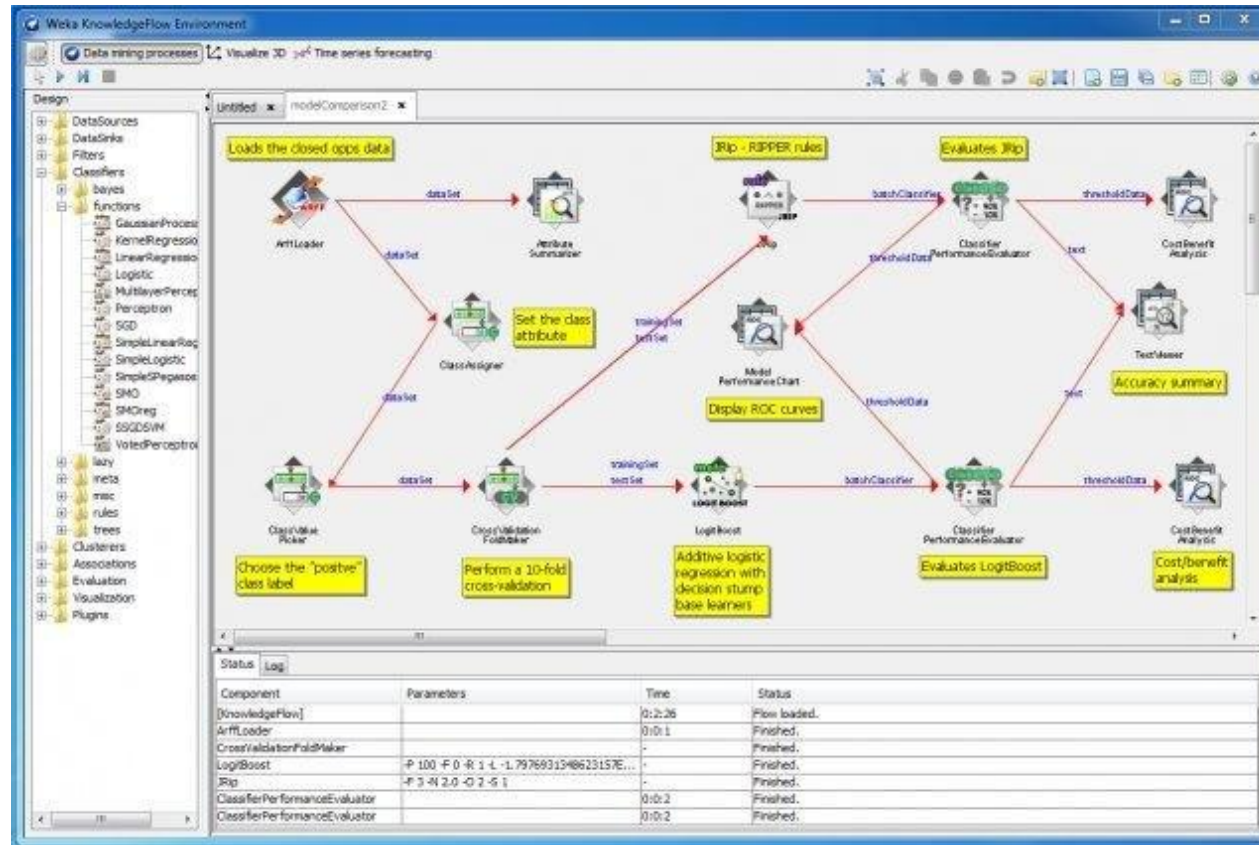


Esportazione dei motori addestrati

E' possibile salvare I motori addestrati da weka in file .model

Questi file possono essere richiamati per valutare nuovi dataset senza dover riaddestrare la rete, operazione che spesso può richiedere molto tempo

Knowledge Flow



Simple CLI

weka.classifiers.bayes.BayesNet

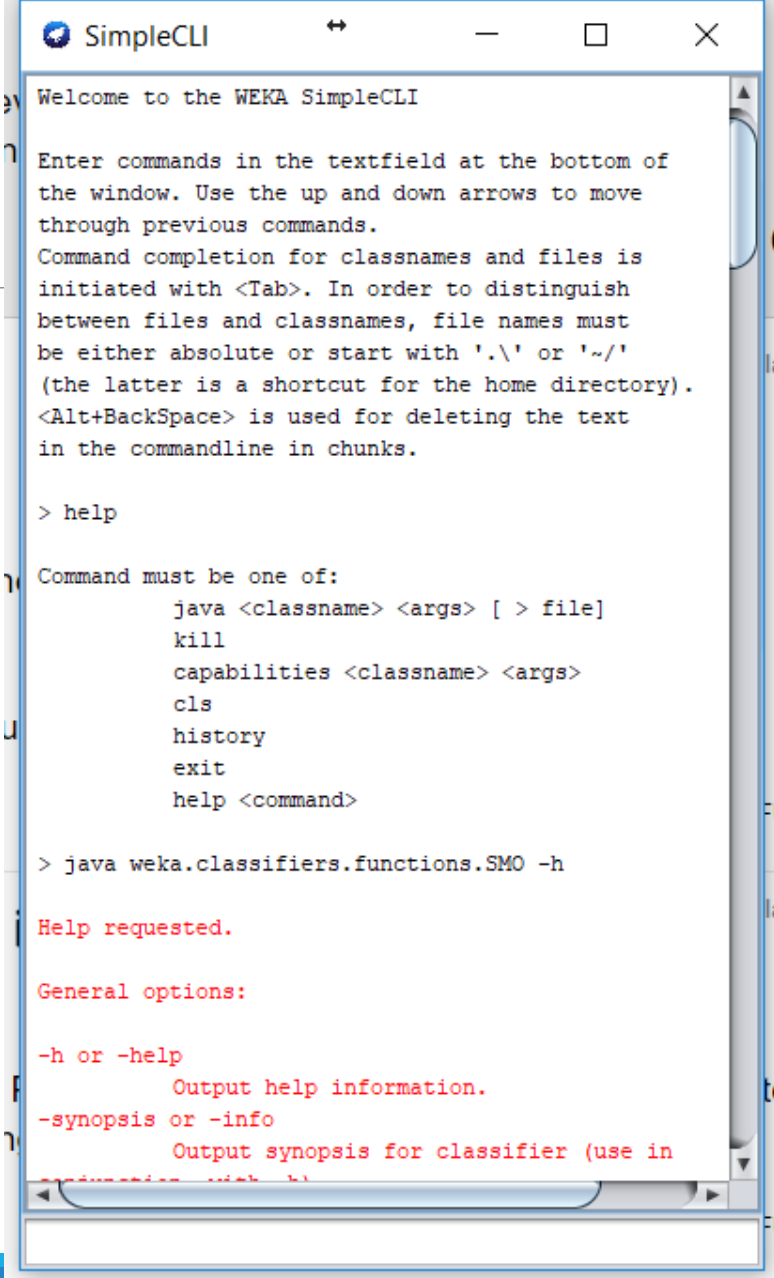
weka.classifiers.trees.J48

weka.classifiers.bayes.NaiveBayesMultinomial

weka.classifiers.RandomizableClassifier

weka.classifiers.functions.LinearRegression

weka.classifiers.trees.RandomTree



```
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>

> java weka.classifiers.functions.SMO -h

Help requested.

General options:

-h or -help
    Output help information.
-synopsis or -info
    Output synopsis for classifier (use in
    conjunction with -h)
```

Esempio pratico

