

Intelligenza Artificiale

A.A. 2014-2015

**Introduzione a WEKA
Esercizio su Reti Neurali Artificiali**

Weka

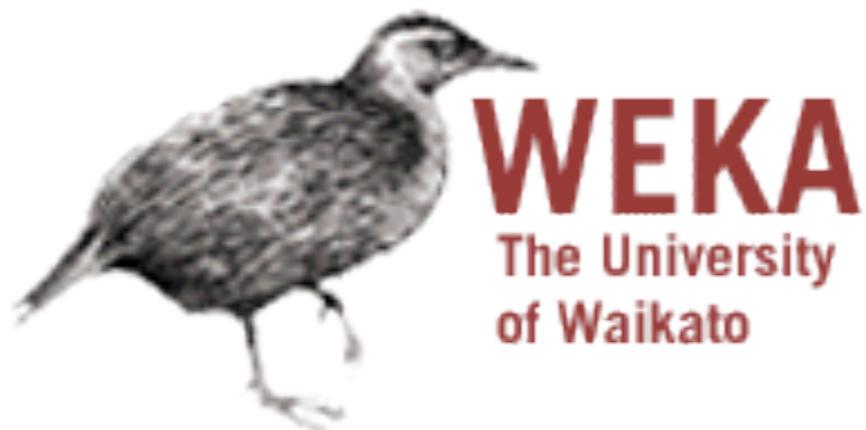
Analisi dei dati ed estrazione della conoscenza con WEKA

In theory, there is no difference between theory and practice. But, in practice, there is.

Jan L.A. van de Snepscheut (1953-1994), computer scientist and educator

WEKA

- Weka è l'acronimo di **Waikato Environment for Knowledge Analysis**, ma è anche il nome di un uccello della nuova zelanda...



Installazione

- Weka è scritto in Java, quindi si può utilizzare su qualunque sistema operativo dotato di un ambiente di esecuzione Java.
 - Windows
 - Se il JRE (Java Runtime Environment) è già installato, basta scaricare il solo programma di installazione Weka ed eseguirlo.
 - Altrimenti, la cosa più conveniente è scaricare il programma di installazione Weka + JRE che installa entrambi gli ambienti in una volta sola.

Gli ambienti operativi di Weka

- Una volta lanciato Weka possiamo scegliere tra 4 diversi ambienti operativi:
 - SimpleCLI
 - Explorer
 - Experimenter
 - KnowledgeFlow

SimpleCLI

- E' un ambiente a linea di comando, da usare per invocare direttamente le varie classi Java di cui Weka è composto.
- Tutto quello che si può fare dalla SimpleCLI è possibile farlo anche da un ambiente a linea di comando come il "prompt di DOS" di Windows o la shell di Unix.

Explorer

- E' l'ambiente che utilizzeremo più spesso. Con esso si possono caricare degli insiemi di dati, visualizzare in modo grafico la disposizione degli attributi, effettuare una serie di operazioni preliminari di preparazione, ed eseguire algoritmi di classificazione, clustering, selezione di attributi e determinazione di regole associative.

Explorer



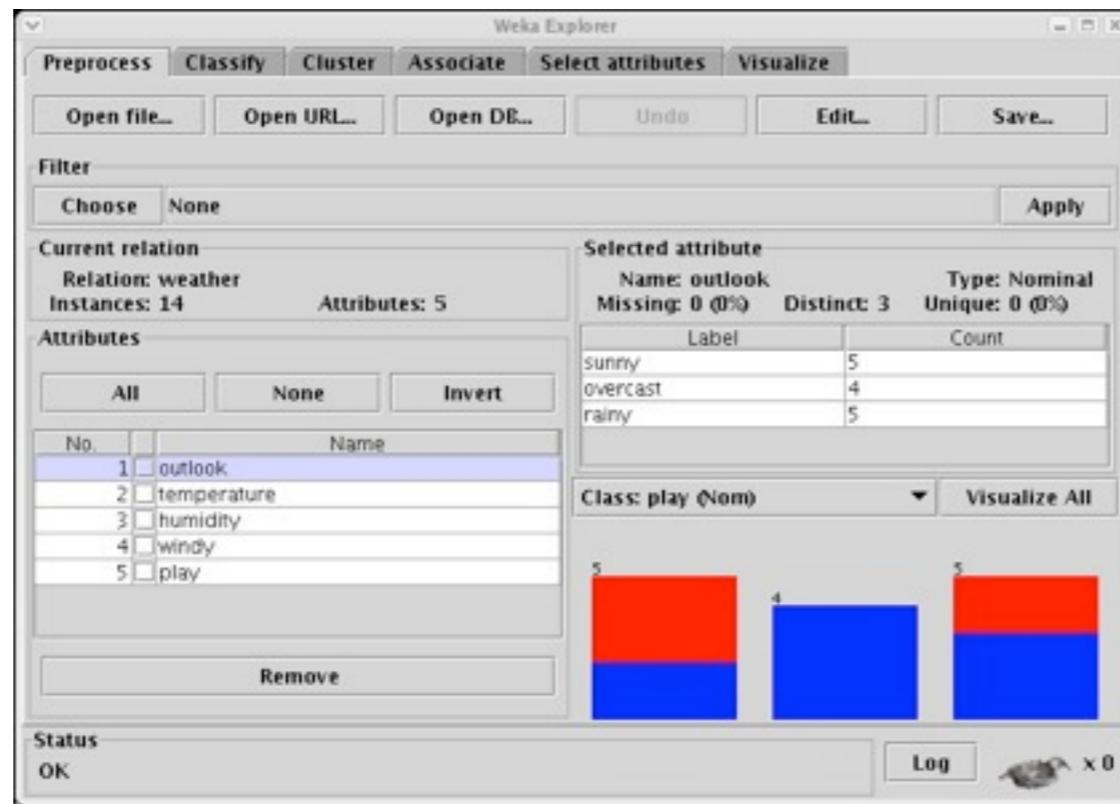
Experimenter

- E' una versione batch dell'Explorer. Consente di impostare una serie di analisi, su vari insiemi di dati e con vari algoritmi, ed eseguirle alla fine tutte insieme. E' possibile in questo modo confrontare vari tipi di algoritmi, e determinare qual è il più adatto a uno specifico insieme di dati.

Knowledge Flow

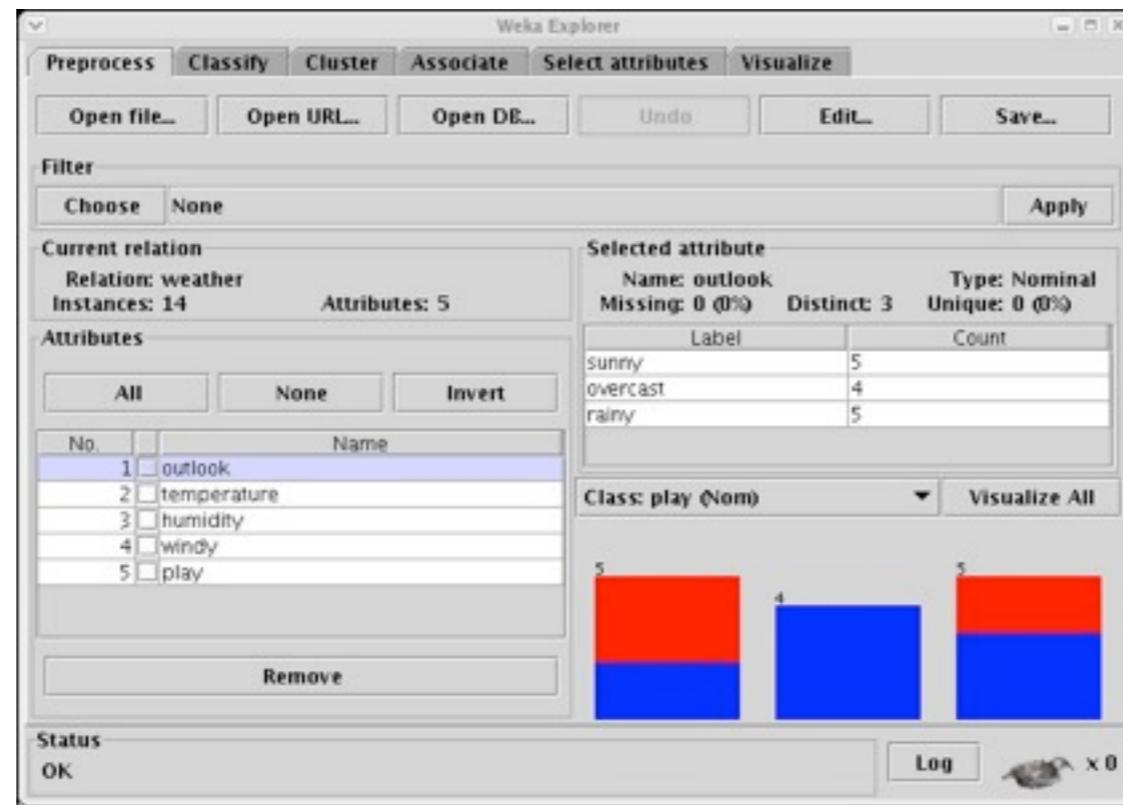
- Una variante dell'explorer, in cui le operazioni da eseguire esprimono in un ambiente grafico, disegnando un diagramma che esprime il “flusso della conoscenza”. E’ possibile selezionare da una tavolozza varie componenti come sorgenti di dati, filtri, algoritmi di classificazione e collegarli tra loro in un diagramma tipicamente detto “data-flow”.

Explorer (1/2)



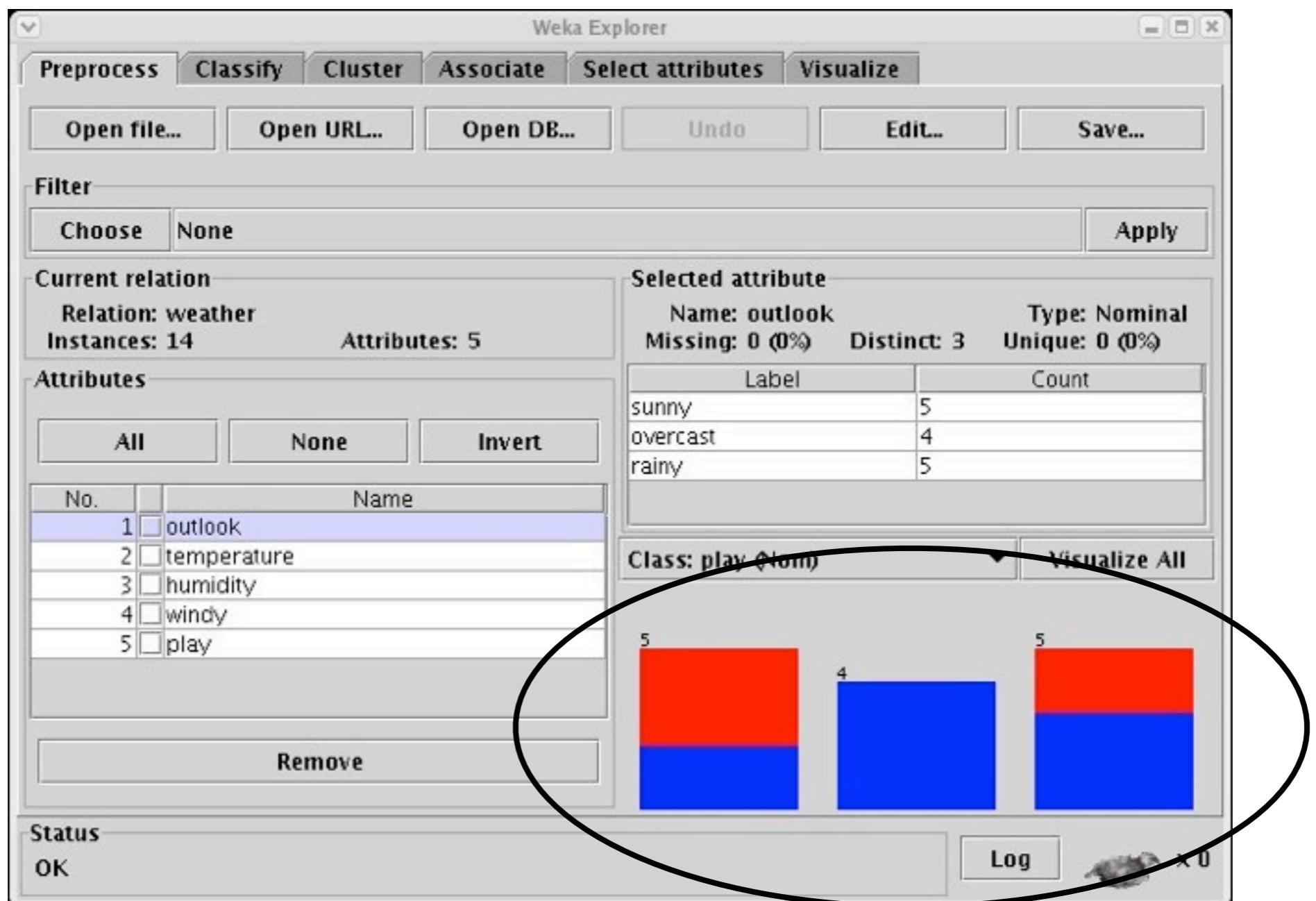
- Dall'Explorer, per gli attributi nominali abbiamo l'elenco dei possibili valori e, per ognuno di essi, il numero di istanze con quel valore. Interessante anche il conteggio del numero di istanze in cui l'attributo manca e del numero di valori che appaiono una sola volta.

Explorer (2/2)



- Dall'Explorer, per gli attributi numerici, abbiamo le informazioni sul valore massimo, minimo, media e deviazioni standard, oltre alle solite informazioni su numero di valori diverse, numero di valori unici e numero di istanze col valore mancante.

Iistogramma riassuntivo



Il formato dati ARFF

- Weka può prelevare i dati da
 - Un file di testo sul computer locale, in formato ARFF
 - Un file su Web in formato ARFF
 - Un database, tramite il driver JDBC

Il formato dati ARFF

- Un file ARFF è composto da una intestazione e il corpo dei dati vero e proprio.

```
@relation weather
```

```
% Relazione weather-data
```

```
@attribute outlook {sunny, overcast, rainy}
```

```
@attribute temperature real
```

```
@attribute humidity real
```

```
@attribute windy {TRUE, FALSE}
```

```
@attribute play {yes, no}
```

```
@data
```

```
sunny, 85, 85, FALSE, no
```

```
sunny, 80, 90, TRUE, no
```

```
overcast, 83, 86, FALSE, yes
```

Il formato dati ARFF

- La riga `@relation weather` specifica un nome per la relazione.
- La riga `"% Relazione weather-data"` è un commento.
- La riga `@attribute outlook {sunny, overcast, rainy}` specifica che il primo attributo è di tipo categoriale e può assumere i valori sunny, overcast e rainy. Il nome dell'attributo è "outlook"
- La riga `@attribute temperature real` specifica che il secondo attributo è di tipo numerico ed ha nome "temperature"
- La riga `@data` indica l'inizio dei dati veri e propri
- La riga `sunny, 85, 85, FALSE, no` indica che la prima istanza ha valori outlook=sunny, temperature=85, humidity=85, windy=FALSE e play=no.

ARFF per classificare...

- In generale, ogni volta che serve individuare un attributo particolare come la “classe” dell’istanza (ad esempio per problemi di classificazione), l’ultimo attributo gioca questo ruolo.
- Si può utilizzare il valore **?** come dato mancante.

ESERCIZIO

- Modificare i dati del seguente file, in modo da aggiungere un'istanza con attributo outlook mancante e un nuovo valore di outlook che occorre unicamente in una istanza.

```
@relation weather  
% Relazione weather-data  
  
@attribute outlook {sunny, overcast, rainy}  
@attribute temperature real  
@attribute humidity real  
@attribute windy {TRUE, FALSE}  
@attribute play {yes, no}  
  
@data  
sunny, 85, 85, FALSE, no  
sunny, 80, 90, TRUE, no  
overcast, 83, 86, FALSE, yes
```

Pre-elaborazione dei dati

- Dall'Explorer di Weka, tutte le operazioni di pre-elaborazioni si possono eseguire dalla scheda “Pre-process”.
- Filtri:
 - Supervisionati:
 - Esiste un attributo speciale, l'attributo classe, che viene usato per guidare le operazioni di filtraggio.
 - Non Supervisionati:
 - Tratta gli attributi allo stesso modo

Filtri supervisionati

- **AddCluster**: aggiunge un nuovo attributo che rappresenta la classe assegnata ad ogni istanza da un algoritmo di clustering (raggruppamento).
 - Esempio: set di dati iris.arff, sugli attributi petalwidth e petallength. Controllare la precisione ottenuta tramite il classificatore J48 dopo la rimozione degli attributi originari, rispetto al set di dati iniziale.
 - Esercizio: cosa succede se si esegue il raggruppamento solo su uno dei due attributi?
- **Discretize**: discretizza un attributo con il metodo dell'equi-width o equi-depth binning
 - Esempio: set di dati iris.arff, provare i risultati delle due varianti sull'attributo sepallength con 4 intervalli.
- **Normalize**: normalizza col metodo min-max, restringendo tutti gli attributi numerici all'intervallo 0-1.
 - Esempio: set di dati iris.arff.
- **Numeric Transform**: applica una generica funzione matematica a determinati attributi.
- **Replace Missing Values**: rimpiazza tutti i valori mancanti con la moda dell'attributo (se si tratta di attributi nominali) o la media (per attributi numerici)
- **Standardize**: normalizza col metodo z-score tutti gli attributi numerici
- **Resample**: campionamento semplice con rimpiazzamento dei dati

Filtri non supervisionati

- **Discretize**: discretizza gli attributi usando il metodo MDL di Fayyad & Irani's
 - esperimento: confrontare i risultati di precisione (col classificatore J48) dei dati originari, dei dati discretizzati con binning e con quest'ultimo metodo del set di dati "segmentation-challenge.arff"
- **Resample**: campionamento con rimpiazzamento dei dati
 - può funzionare come il metodo non supervisionato oppure è in grado di effettuare il campionamento in modo che l'attributo classe abbia una distribuzione uniforme. Ciò avviene settando a 1.0 il valore dell'attributo biasToUniformClass.
 - Esempio: provare sul set di dati soybean.arff

Selezione di attributi rilevanti

- Esistono vari modi per la selezione degli attributi. Alcuni considerano un attributo alla volta e determinano la misura della sua significatività in base alla capacità di discriminare una classe da un'altra. Altri considerano invece una collezione di attributi e ne valutano l'efficacia complessiva.

Selezione di attributi rilevanti

- Possiamo prendere come set di dati **zoo.arff** e scegliere come valutatore per gli attributi il metodo InfoGainAttributeEval che calcola, per ogni attributo, il guadagno di informazione...
- Ricordiamo che
 - $\text{InfoGain}(\text{Attr}) = H(\text{Class}) - H(\text{Class}|\text{Attr})$
 - $\text{GainRatio}(\text{Attr}) = \text{InfoGain}(\text{Attr}) / H(\text{Attr})$

Output

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

Search Method:

Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 18 type):
Information Gain Ranking Filter

Ranked attributes:

2.3906	1	animal
1.3108	14	legs
0.9743	5	milk
0.8657	9	toothed
0.8301	4	eggs
0.7907	2	hair
0.7179	3	feathers
0.6762	10	backbone
0.6145	11	breathes
0.5005	15	tail
0.4697	6	airborne
0.4666	13	fins
0.3895	7	aquatic
0.3085	17	catsize
0.1331	12	venomous
0.0934	8	predator
0.0507	16	domestic

Selected attributes: 1,14,5,9,4,2,3,10,11,15,6,13,7,17,12,8,16 : 17

Feature Selection

- Per ogni attributo viene visualizzato un punteggio, in questo caso il guadagno di informazione, a partire dall'attributo che ha il guadagno maggiore fino a quello che ha il guadagno minore. Normalmente tutti gli attributi vengono selezionati, indipendentemente dal valore del punteggio. Tuttavia, modificando i parametri del metodo di ricerca **Ranker**, è possibile cambiare il comportamento
- **numToSelect**: se viene impostato al valore **n** positivo, indica che si vogliono selezionare solo i primi **n** attributi in ordine di rilevanza. Se impostato a un valore negativo, la selezione avviene in base al valore di soglia di cui si parla qui sotto.
- **threshold**: se **numToSelect** è negativo, vengono selezionati gli attributi con ranking superiore a questa soglia. Il valore di default di -1.7976931348623157E308 causa la selezione di tutti gli attributi.

Applicare la riduzione...

- Supponiamo di voler determinare un albero di classificazione usando l'algoritmo **ID3**. Dopo aver discretizzato i dati col filtro supervisionato **Discretize** (l'algoritmo ID3 si applica solo a dati discreti) si ottiene un grafo ad un solo livello in cui viene selezionato l'attributo animal. Questo perché ID3 utilizza il guadagno di informazione per selezionare gli attributi sui vari nodi. L'accuratezza del metodo, valutata con la tecnica della "**cross validation**", è vicino allo 0!!
- L'algoritmo **J48** (che è un clone di **C4.5**) si basa invece sul **Gain Ratio**, ed è immune a questo fenomeno, e genera un buon classificatore con accuratezza del 92% circa. Notare che se si elimina l'attributo animal manualmente, l'algoritmo ID3 produce un buon albero di classificazione, migliore di quello di C4.5 (accuratezza del 97%).

Metodi di classificazione

- Analizziamo un insieme di dati (zoo.arff) con i vari metodi di classificazione conosciuti.
- Prima di effettuare le varie analisi, discretizziamo il set di dati usando il filtro supervisionato Discretize. In questo modo possiamo utilizzare anche algoritmi che funzionano solo su dati discreti.

Classificare...

■ **ZeroR**

- Viene selezionata come predizione la classe più frequente nel set di dati. Equivale a un albero di classificazione di altezza 0.
 - Accuratezza = 41%
- Notare che per ZeroR (e per tutti gli algoritmi di classificazione) è possibile ottenere uno scatter plot, ma con in più l'indicazione se l'istanza è classificata correttamente oppure no.
 - Istanza classificate correttamente sono marcate con '+', mentre le istanze classificate non correttamente sono classificate con un quadratino.

Classificare...

- J48 e ID3
 - Genera un albero di classificazione.
 - Accuratezza: 89%
- Naive Bayes
 - Utilizza il metodo bayesiano con incremento dei contatori di 1, in modo da evitare i problemi che sorgono con probabilità nulle.
 - Accuratezza: 93%

Classificare...

■ BayesNet

- È un metodo per apprendere reti bayesiane. La struttura della rete può essere fissata o può essere appresa dall'algoritmo, e si possono modificare sia l'euristica usata per la ricerca della struttura della rete, sia il metodo usato per stimare le probabilità condizionate.
- Accuratezza: 95%

Classificare...

- IDk
 - Implementa il metodo “k-NN”. E’ possibile selezionare il valore di k e se si vuole che le istanze siano pesate a seconda della distanza.
 - Accuratezza:
 - K=1 → 96%
 - K=5 → 93%

Analisi di raggruppamento

- Weka dispone di alcuni (pochi) tipi di analisi di raggruppamento.
- K-Means
 - Input:
 - numero di partizioni
 - “seme” per la generazione di numeri casuali.

Cluster Mode

- Nel pannello di sinistra, cluster mode, è possibile selezionare come effettuare il raggruppamento. Si può scegliere tra:
 - Use Training Set.
 - Viene applicato l'algoritmo di raggruppamento su tutti gli attributi. Come metodo per valutare la bontà del risultato, viene calcolata la somma delle distanze al quadrato dal centro del cluster.
 - Supplied Test Set.
 - Simile al metodo precedente, ma la somma delle distanze al quadrato, viene calcolata su un insieme di test differente
 - Percentage Split.
 - Il set di dati viene diviso in una parte di addestramento ed una di test secondo al percentuale indicata.
 - Classes to cluster evaluation.
 - Si sfrutta l'esistenza di un attributo di classe. L'addestramento avviene su tutto l'insieme di dati, e successivamente viene mostrato, per ogni cluster, come si distribuiscono le varie classi al suo interno.

ANN in weka

- Scelta dell'algoritmo di classificazione:
 - Classifiers/Functions/MultilayerPerceptron
 - Percettrone multilivello
 - Funzione di attivazione sigmoide
 - Apprendimento tramite backpropagation
 - Impostare i parametri per l'apprendimento della rete
 - Avviare il processo di apprendimento e modificare eventualmente i parametri dall'interfaccia grafica

ANN in weka

■ Aprire il file bank-data.arff

- 12 Attributi
- 600 Istanze

Attributo	Descrizione
id	Identificativo univoco
age	Età del cliente in anni (numeric)
sex	MALE/FEMALE
region	Inner_city/town/rural/suburban
income	Reddito del cliente (numeric)
married	Sposato? YES/NO
children	Numero figli
car	Possiede un automobile? YES/NO
save_act	Ha un conto di risparmio? YES/NO
current_act	Ha un conto corrente? YES/NO
mortgage	Ha un mutuo? YES/NO
pep	Ha acquistato un PEP dopo l'ultimo invio postale? YES/NO

ANN in weka

- Prima di iniziare ad eseguire prove sui parametri del multi-layer perceptron
 - Eliminare dal dataset l'attributo id
 - Scegliere Percentage split 66%

Impostazione dei parametri

- GUI: se TRUE visualizza un'interfaccia per modificare la rete neurale (es. aggiungere, eliminare nodi) durante la fase di training
 - **VERDE** = nodi di input
 - **ROSSO** = nodi nascosti
 - **ARANCIONE** = nodi di output

Impostazione dei parametri

- Hidden Layer: permette di definire il numero di livelli e nodi nascosti
 - #nodi 1° livello nascosto, #nodi 2° livello nascosto
 - HiddenLayer = 4,2
 - 4 nodi nel primo livello, 2 nodi nel secondo livello
- Configurazioni predefinite (alcuni esempi):
 - a: 1 solo livello nascosto che contiene un numero di nodi nascosti pari a (#attributi + #classi)/2
 - i: 1 solo livello nascosto che contiene un numero di nodi nascosti pari a #attributi
- Per gli attributi nominali viene creato un nodo per ogni valore assumibile dall'attributo stesso

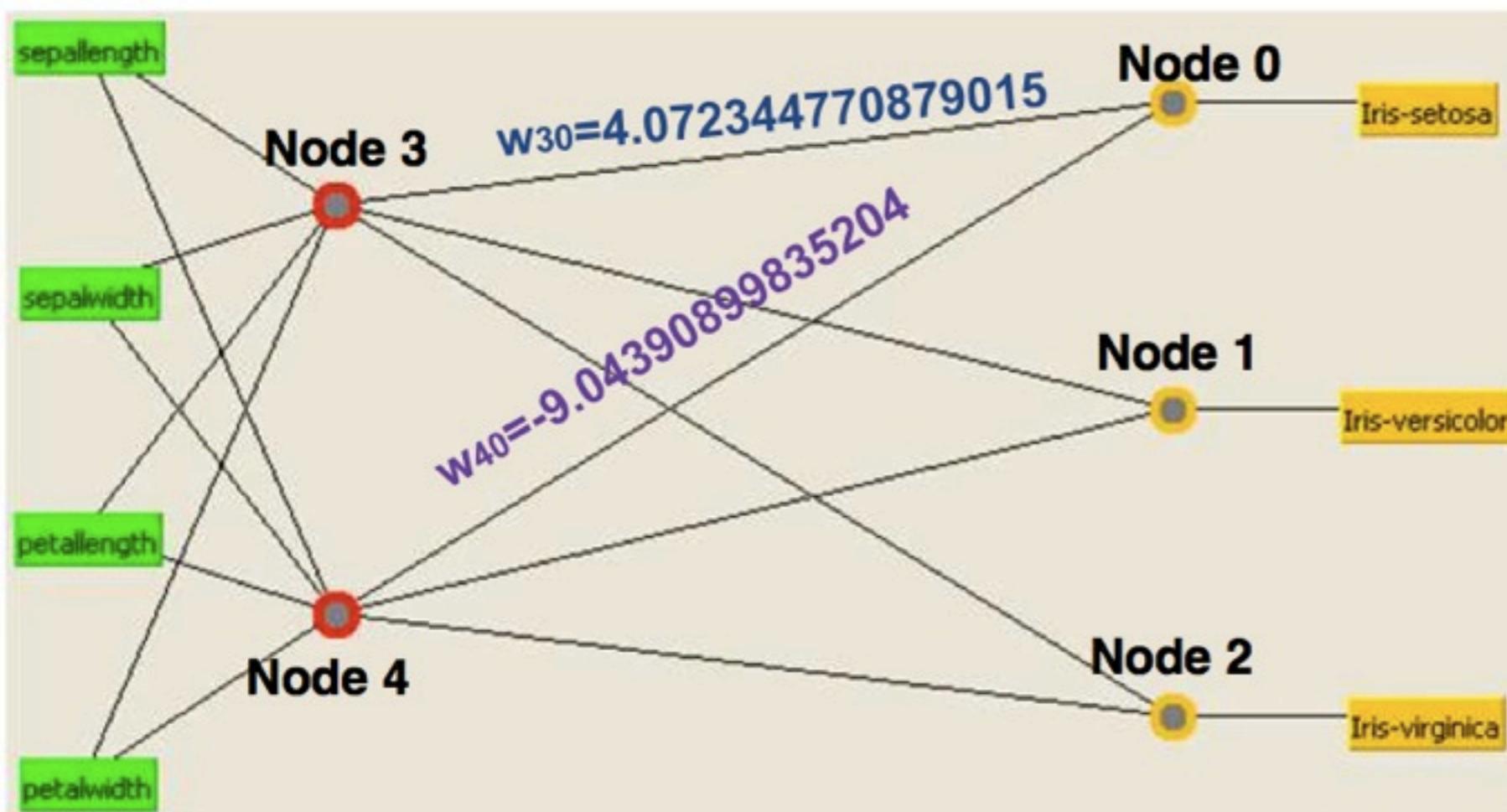
Impostazione dei parametri

- **decay**: se true il learning rate viene decrementato (il learning rate viene diviso per il numero corrispondente all'attuale epoca di apprendimento)
- **trainingTime**: numero di epoche di addestramento
- **validationSetSize**: dimensione del validation set (% del dataset su cui si sta eseguendo training)
- **validationThreshold**: usato per terminare anticipatamente l'apprendimento. Numero di volte in cui l'errore sul validation set può peggiorare prima di terminare l'apprendimento (la soglia è utilizzata solo se la dimensione del validation size>0)
 - Quando è utile?

Addestramento

- Utilizzando l'interfaccia GUI
 - Start: avvio dell'addestramento (possibilità di modifica dei parametri)
 - Accept: viene accettata la configurazione della rete; tale rete verrà usata per eseguire la classificazione
- Opzioni di test (alcuni esempi)
 - Use training set:
 - Start: esegue l'addestramento sul training set
 - Accept: visualizza le statistiche di classificazione del training set
 - Percentage Split:
 - 1° start: esegue addestramento sul training set; 1°accept: accetta la configurazione della rete per il training set
 - 2° start: esegue validazione sul test set; 2°accept: visualizza le statistiche di classificazione sul test set

Configurazioni della rete



Class Iris-setosa
Input
Node 0
Class Iris-versicolor
Input
Node 1
Class Iris-virginica
Input
Node 2

Sigmoid Node 0	
Inputs	Weights
Threshold	0.4113564397167993
Node 3	4.072344770879015
Node 4	-9.04390899835204

Sigmoid Node 3	
Inputs	Weights
Threshold	8.060243821406793
Attrib sepallength	1.47666427492804
Attrib sepalwidth	3.9061728372847355
Attrib petallength	-9.762099445240409
Attrib petalwidth	-10.823154076553381

Topologia della rete

- Costruire un perceptron
 - Parametri di default
 - 0 hidden layer
- Costruire un perceptron multilivello
 - Parametri di default
 - hidden layer: a
- Per ogni configurazione modificare learning rate e numero delle epoche

Procedura di apprendimento



- 3 sottoinsiemi: Training, Validation e Test Set
- # nodi in ingresso = # features
- # nodi in uscita = # di classi
- # hidden layers e # nodi per livello: k-fold cross validation sul training set
- Alleno la struttura scelta con tutto il training set, limitando l'overfitting col validation set
- Valuto l'accuratezza finale sul test set

Buone abitudini e regole euristiche



- 1 hidden layer è sufficiente per la stragrande maggioranza dei problemi (e l'allenamento è più rapido)
- Se devo scegliere il numero di nodi interni, parto con pochi e cresco (esponenzialmente) finché vedo un miglioramento:
 - 5 10 20 50 100 ..

Esercizio

- Analizzare labor_training_set.arff: contiene informazioni sulle diverse tipologie contrattuali lavorative stipulate in Canada.
 - 37 istanze
 - Dati missing

Attributo	Descrizione	Attributo	Descrizione
duration	Durata contratto	education-allowance	Sono previsti contributi per la formazione del personale? (YES/NO)
wage-increase-first-year	Incremento del salario per il 1° anno	holidays	Giorni di ferie obbligatori
wage-increase-second-year	Incremento del salario per il 2° anno	vacation	Giorni di assenza aggiuntivi
wage-increase-third-year	Incremento del salario per il 3° anno	Longterm-disability-assistance	Sono previsti contributi per invalidità? (YES/NO)
cost-of-living-adjusting	Fattore di aggiustamento rispetto al costo della vita	Contribution-to-dental-plan	Tipo di contributo previsto per spese dentistiche
working-hours	Ore lavorative	Bereavement-assistance	Sono previsti contributi in caso di lutto? (YES/NO)
pension	Tipo di contributi versati ai fini pensionistici	class	“Good” se il contratto è buono, “Bad” altrimenti
standby_pay	Pagamento aggiuntivo per reperibilità		
shift_diff	Pagamento aggiuntivo per ore lavorative extra (es. notturno)		

Classificazione labor data J48

- Caricare il file labor_training_set.arff
- Effettuare un'analisi manuale dei dati mediante visualizzazione
 - Distribuzioni attributo - classe
 - Distribuzioni attributo - attributo - classe
 - Individuare gli attributo che meglio discriminano le classi
 - Analisi dei dati mancanti
- Eseguire l'algoritmo di classificazione J48. Analizzare la matrice di confusione ottenuta e salvare i risultati per ognuno dei seguenti test:
 - Use training set
 - Supplied Test Set (labor_test_set.arff)
 - Cross Validation

Analisi dei dati

- Alcuni attributi discriminanti
 - Wage-increase
 - Cost of living adjustment
 - Longterm disability assistance
 - Statutory-holidays
 - Contribution to dental plan
- Forte presenza di dati missing

Classificazione J48

Missing values

J48 - Training test 89,19 %

```
a b    <-- classified as  
9 3 | a = bad  
1 24 | b = good
```

J48 – Supplied test 75 %

```
a b    <-- classified as  
4 4 | a = bad  
1 11 | b = good
```

J48 – Cross validation 81.08 %

```
a b    <-- classified as  
9 3 | a = bad  
4 21 | b = good
```

No Missing values

J48 - Training test 94,59 %

```
a b    <-- classified as  
11 1 | a = bad  
1 24 | b = good
```

J48 – Supplied test 90 %

```
a b    <-- classified as  
7 1 | a = bad  
1 11 | b = good
```

J48 – Cross validation 86,49 %

```
a b    <-- classified as  
8 4 | a = bad  
1 24 | b = good
```

Classificazione con MLP

- Eseguire l'algoritmo con MLP su `labor_training_set.arff`
 - utilizzare i parametri di default
 - sperimentare MLP con le stesse opzioni di test utilizzate per J48
 - Salvare le matrici di confusione ottenute
 - Comparare i risultati di J48 con quelli del percettrone multilivello
 - Quale algoritmo produce risultati migliori?
 - Quale possibile motivazione?

Classificazione MLP

Missing values

MLP - Training test 100 %

```
a b    <-- classified as  
12 0 | a = bad  
0 25 | b = good
```

MLP – Supplied test 95 %

```
a b    <-- classified as  
7 1 | a = bad  
0 12 | b = good
```

MLP – Cross validation 91.89 %

```
a b    <-- classified as  
10 2 | a = bad  
1 24 | b = good
```

No Missing values

MLP - Training test 100 %

```
a b    <-- classified as  
12 0 | a = bad  
0 25 | b = good
```

MLP – Supplied test 95 %

```
a b    <-- classified as  
7 1 | a = bad  
0 12 | b = good
```

MLP – Cross validation 96,6 %

```
a b    <-- classified as  
11 1 | a = bad  
1 24 | b = good
```

- MLP ottiene migliori risultati anche in presenza di dati missing
- MLP impiega più tempo per la costruzione del modello
- Con MLP non si ha conoscenza del modello di classificazione, possiamo solo analizzare la topologia della rete utilizzando GUI.

Classificazione MLP

MLP - Training test 100 % MLP – Supplied test 95 % MLP – Cross validation 91,89 %

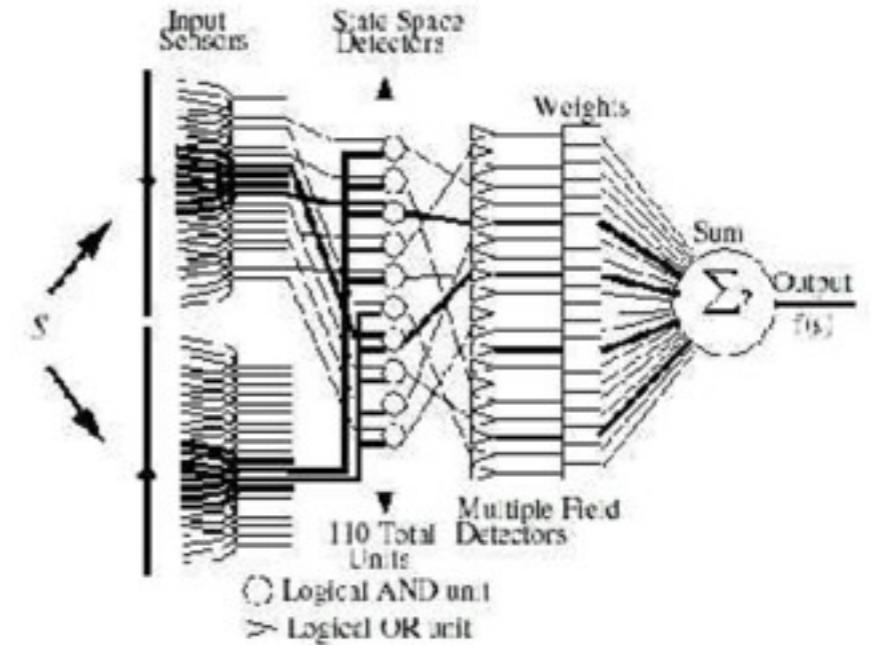
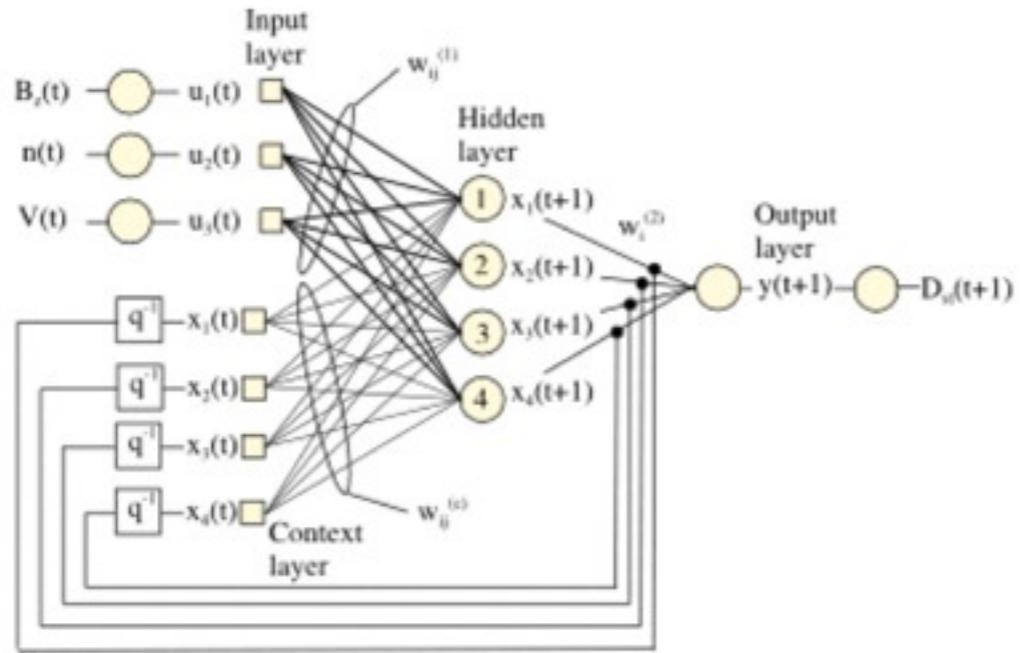
a b <-- classified as
12 0 | a = bad
0 25 | b = good

a b <-- classified as
7 1 | a = bad
0 12 | b = good

a b <-- classified as
10 2 | a = bad
1 24 | b = good

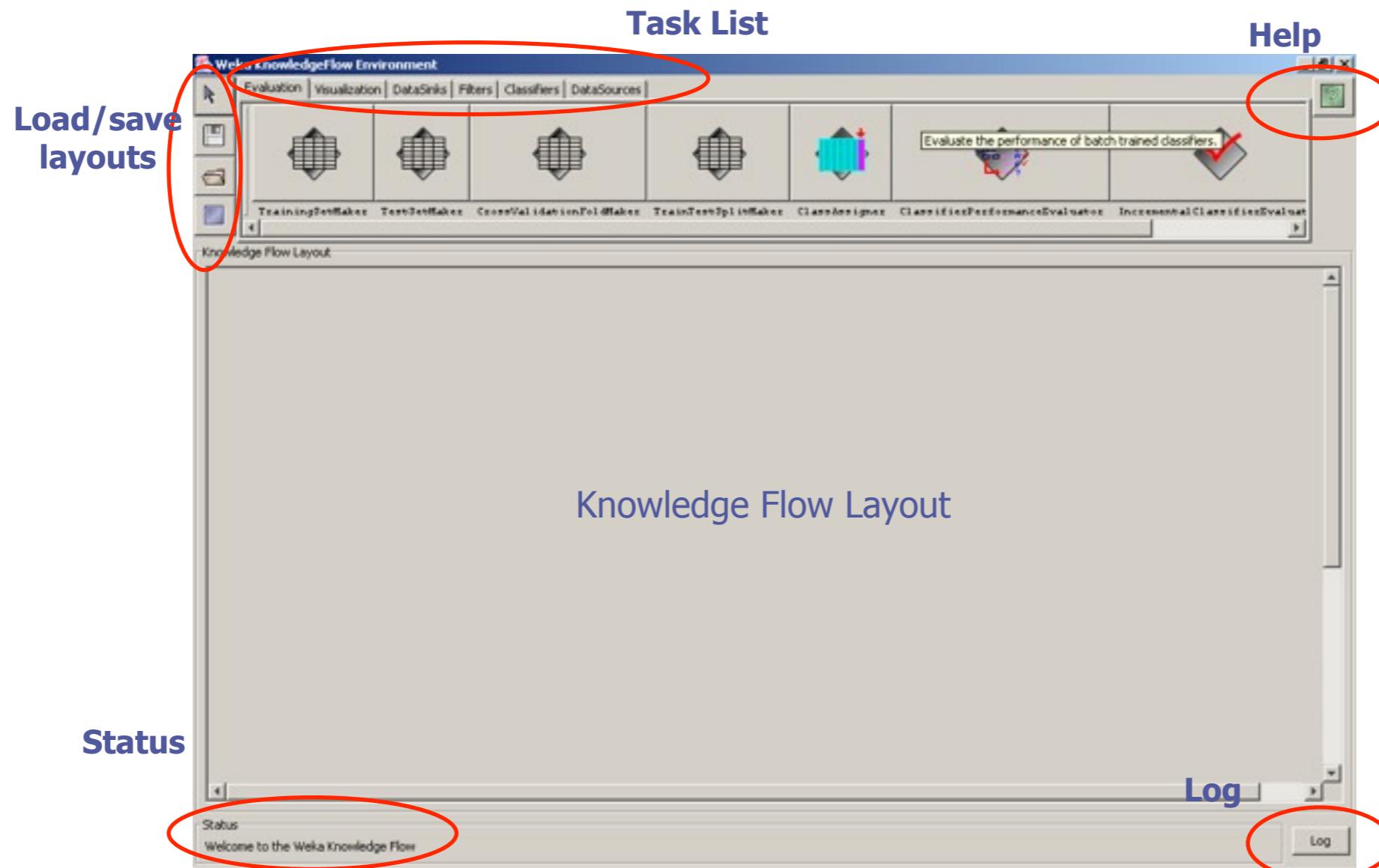
- HiddenLayers=t (aumento dei nodi nascosti)
- Anche aumentando i nodi nascosti (quindi aumentando la complessità della topologia della rete) in presenza di dati missing le prestazioni MLP non migliorano.

Altri tipi di reti neurali

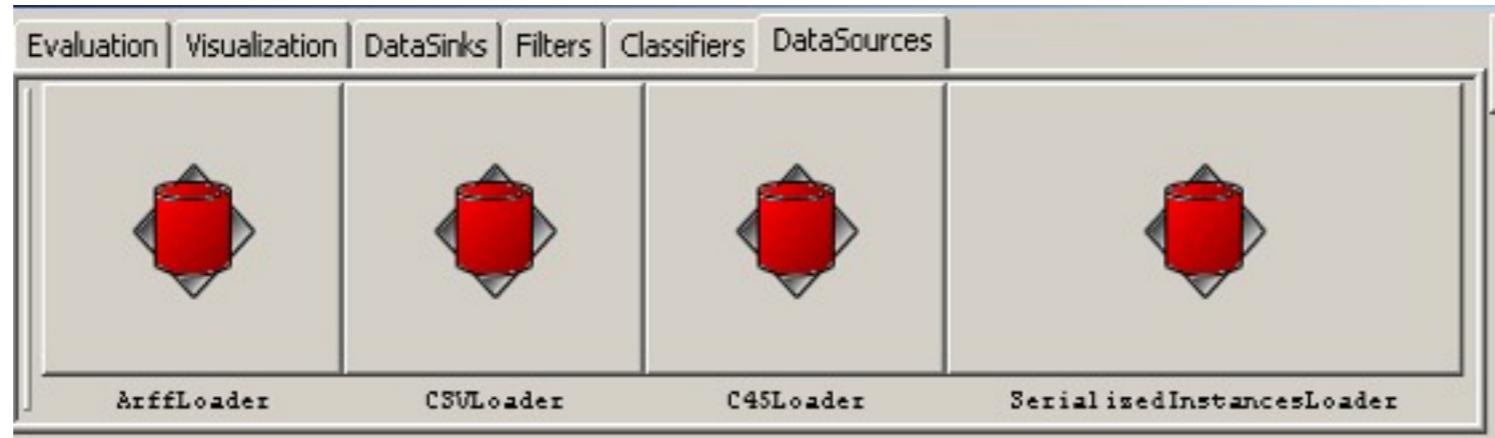


- Recurrent Neural Networks
- Associative Neural Networks
- Stochastic Neural Networks
- Spiking Neural Networks

Knowledge Flow: layouts

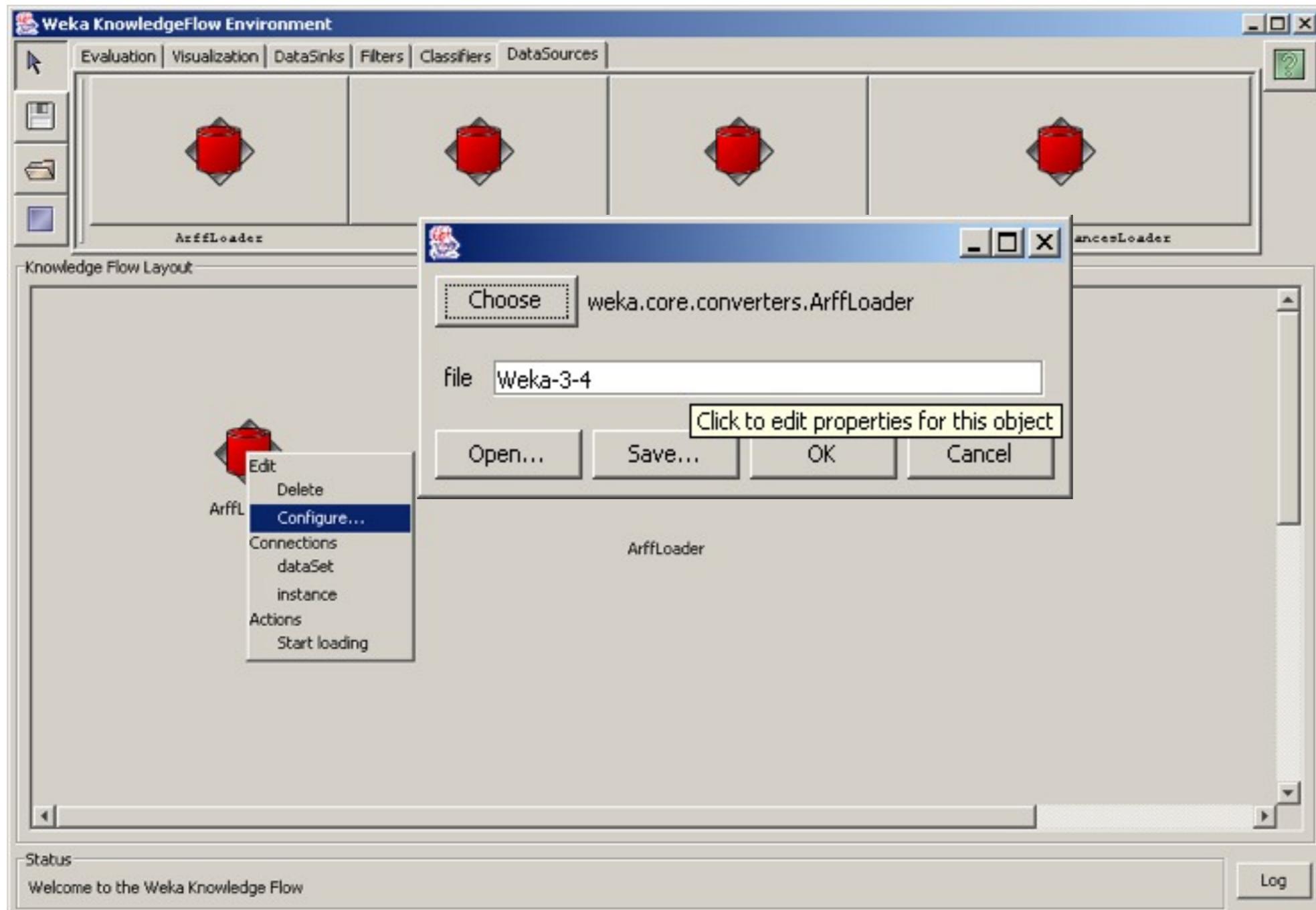


Sorgenti di dati



- ARFF - Attribute-Relational File Format
- CSV - Comma Separated Values
- C45 - Simile ad ARFF ma con metadati su file distinto (.data e .names)
- Serialize Instance - Oggetto “Tabella” serializzato su disco

Caricare da una sorgente

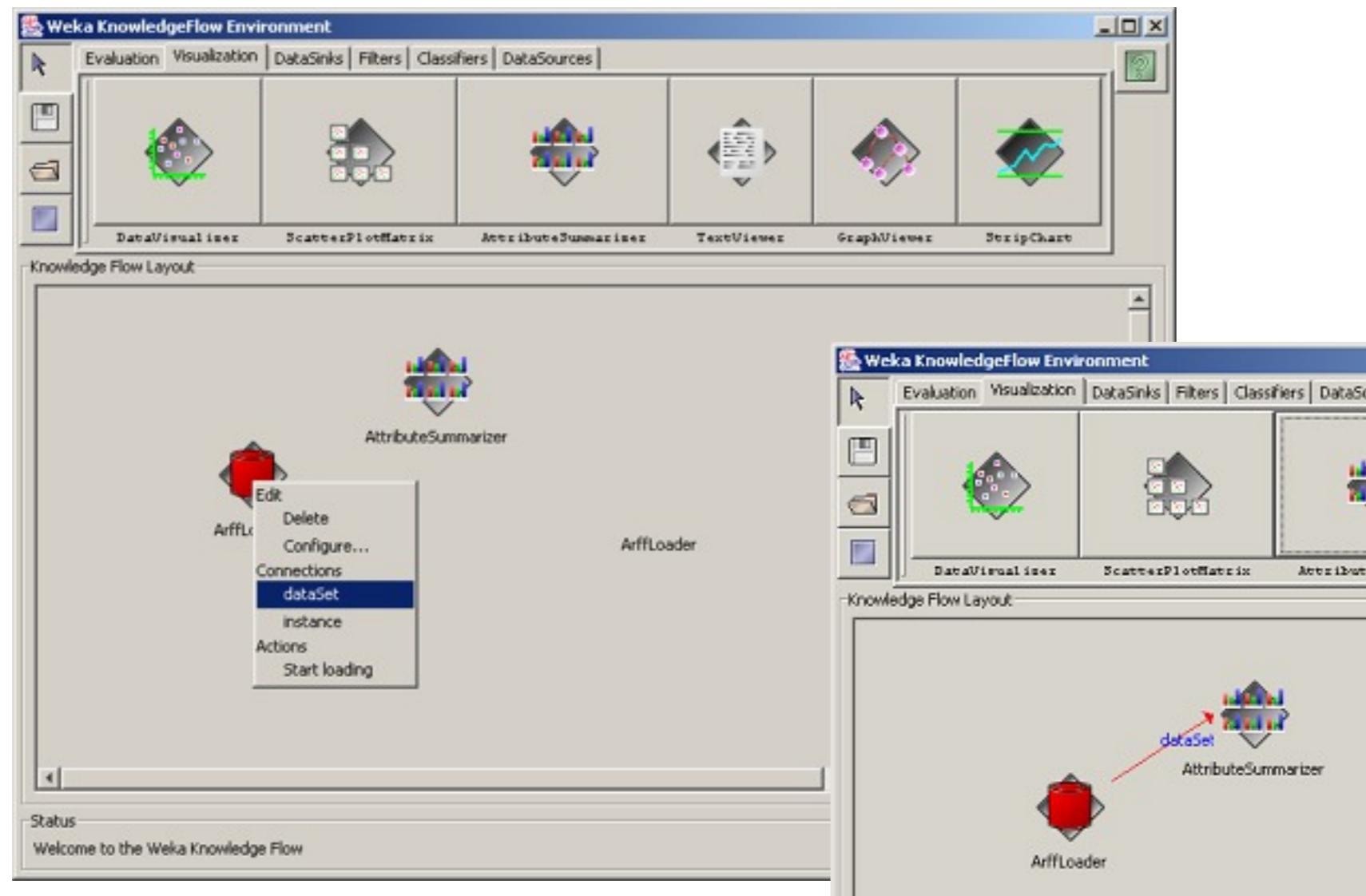


Visualizzatori

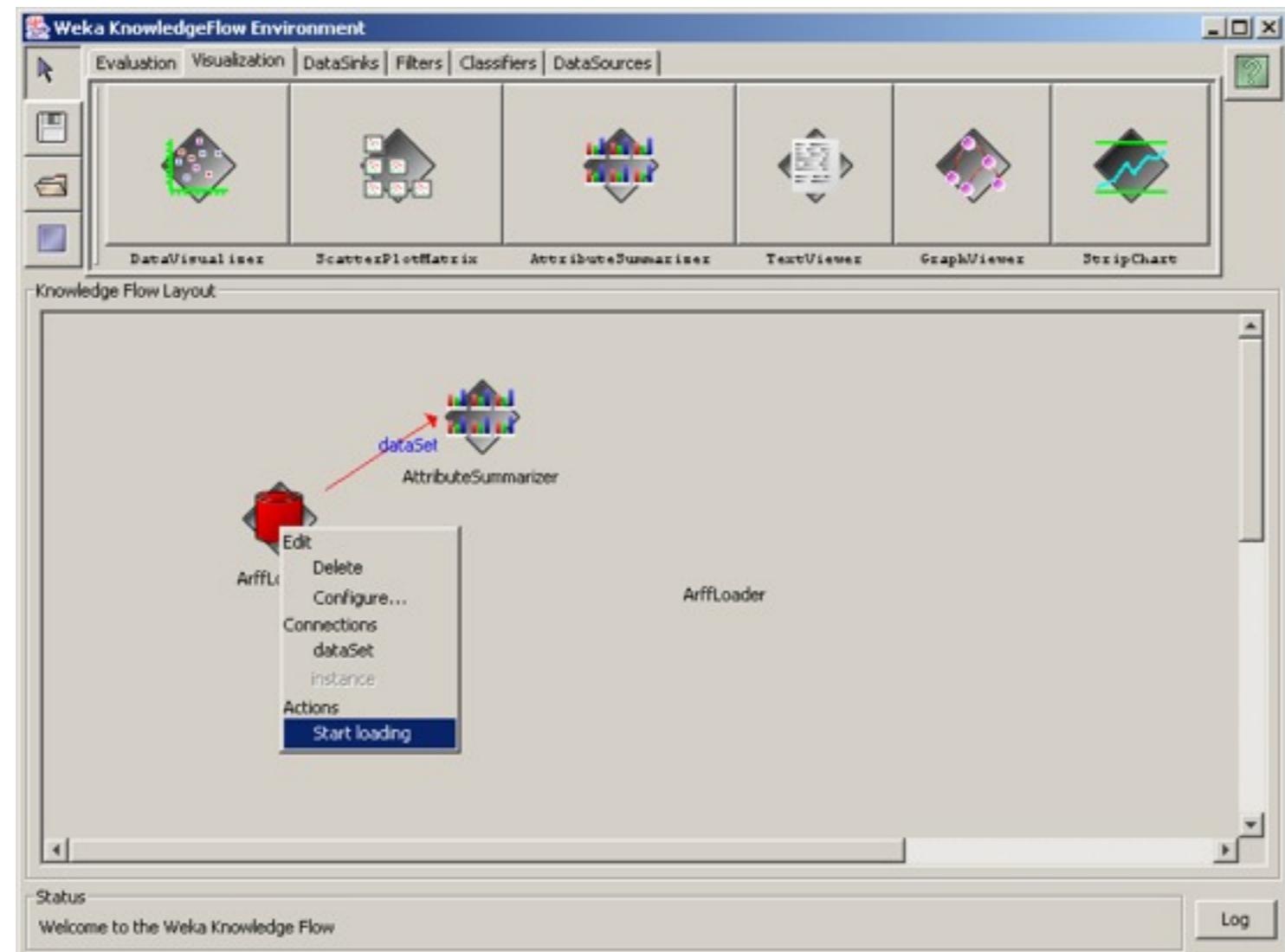


- Data Visualizers
 - 2D scatterplot
- ScatterPlotMatrix
 - Matrice 2D scatterplot
- AttributeSummarizer
 - Distribuzione dei valori negli attributi
- TextViewer
 - Visualizza dati/modelli in formato testuale

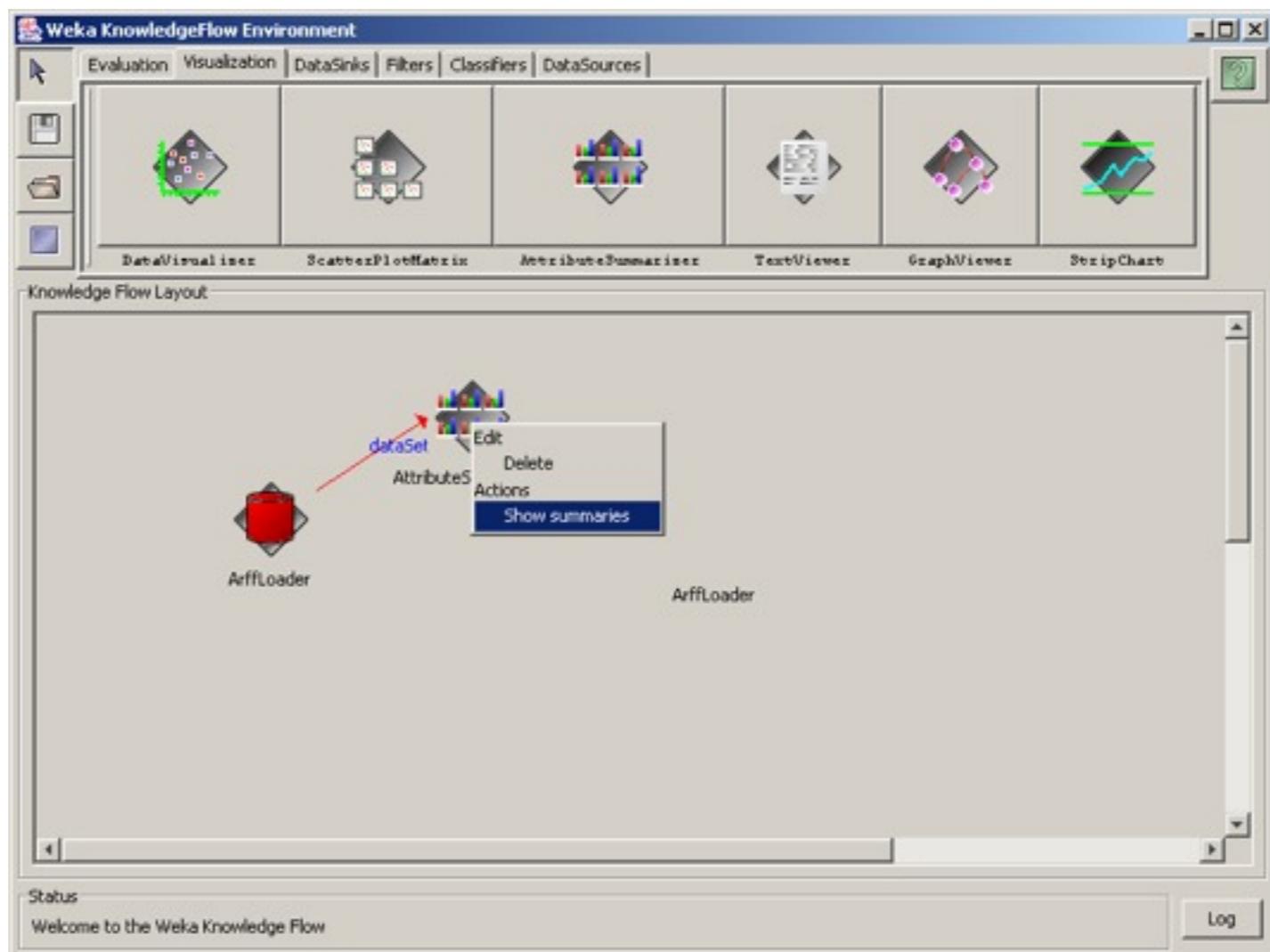
Connessione sorgente task



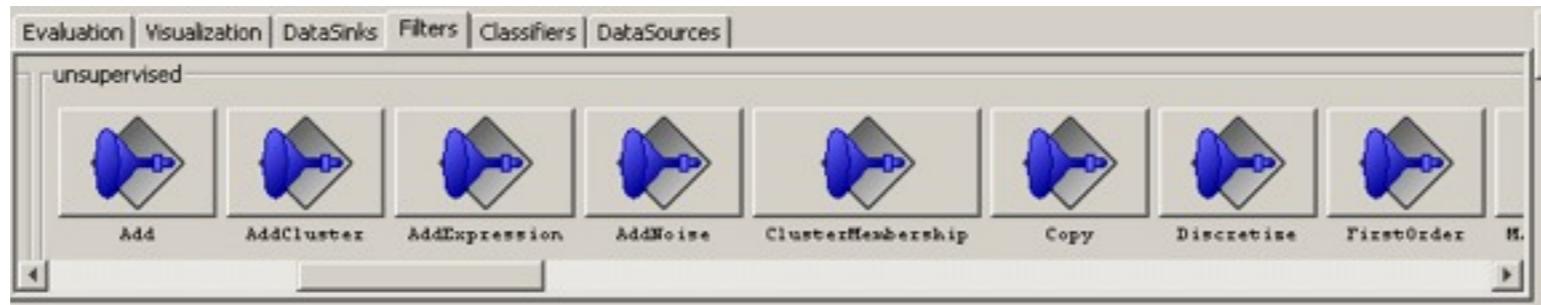
Avvio elaborazione flusso



Visualizzazione dei risultati



Filtri di preprocessamento



- Aggiunta e rimozione attributi
 - Add / AddExpression / Copy
- Remove / RemoveType / RemoveUseless
 - Rimuove un attributo / di un certo tipo / con valori sempre costanti o troppo variabili

Filtri di preprocessamento

- Trasformazioni attributi
 - NumericTransformation
 - Calcola una funzione matematica
 - ReplaceMissingValues
 - Rimpiazza NULL con moda (attributi discreti) e media (attributi continui)
 - AddNoise
 - Perturba una percentuale di valori di un attributo
 - MergeTwoValues
 - Fonde due valori di un attributo in uno solo

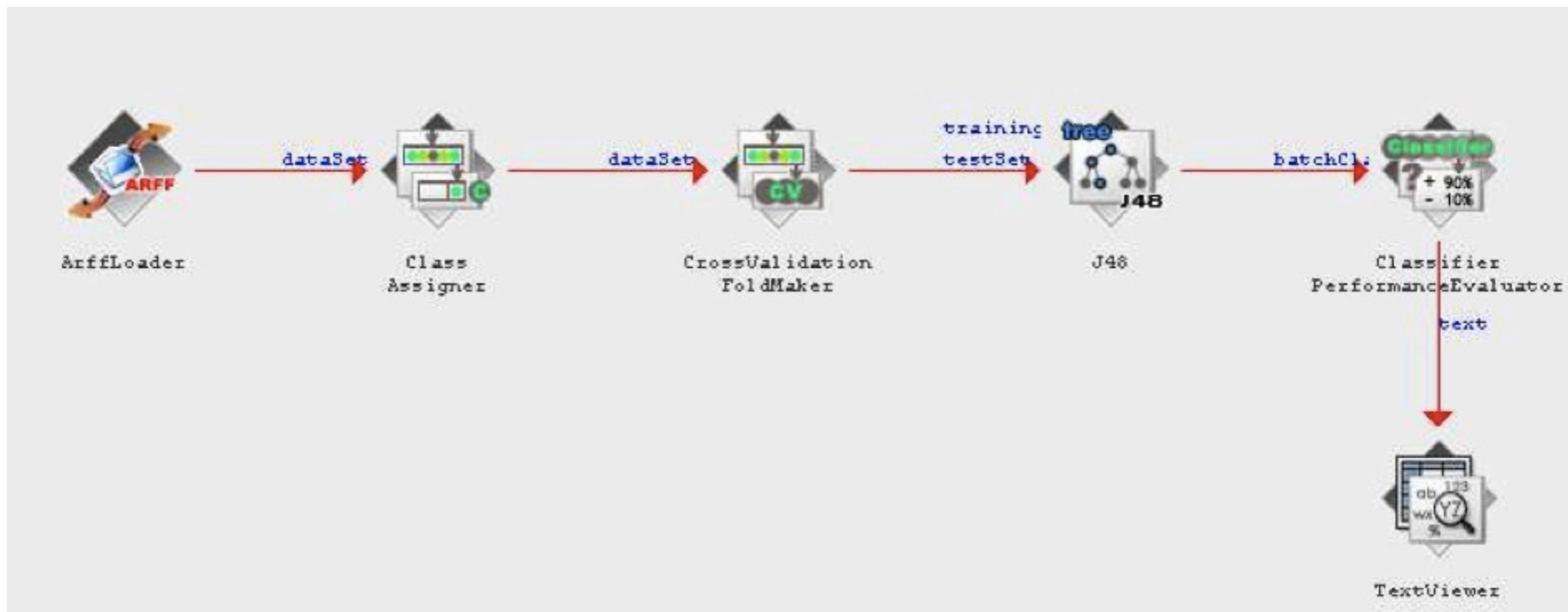
Filtri di preprocessamento

- Discretizzazione/Normalizzazione attributi
 - Normalize
 - Max-Min normalizzazione di un attributo numerico nell'intervallo [0,1)
 - Standardize
 - Z-score normalization
 - Discretize / PKDiscretize
 - Discretizzazione dei valori di un attributo
 - Equal width e Equal frequency

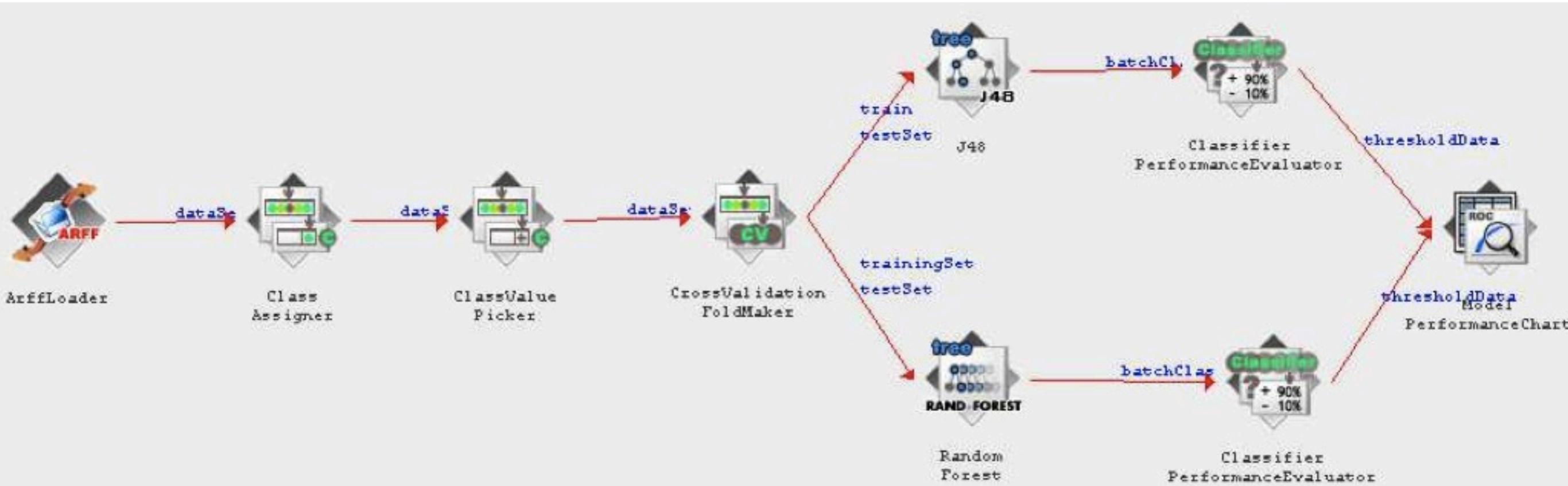
Filtri di preprocessamento

- Selezione e sampling
 - RemoveFold
 - Seleziona “1 out of n” record
 - RemovePercentage
 - Seleziona una percentuale dei record
 - Randomize
 - Mescola record in modo casuale
 - Resample
 - Seleziona una percentuale dei record in modo casuale
 - RemoveRange
 - Seleziona un intervallo di record

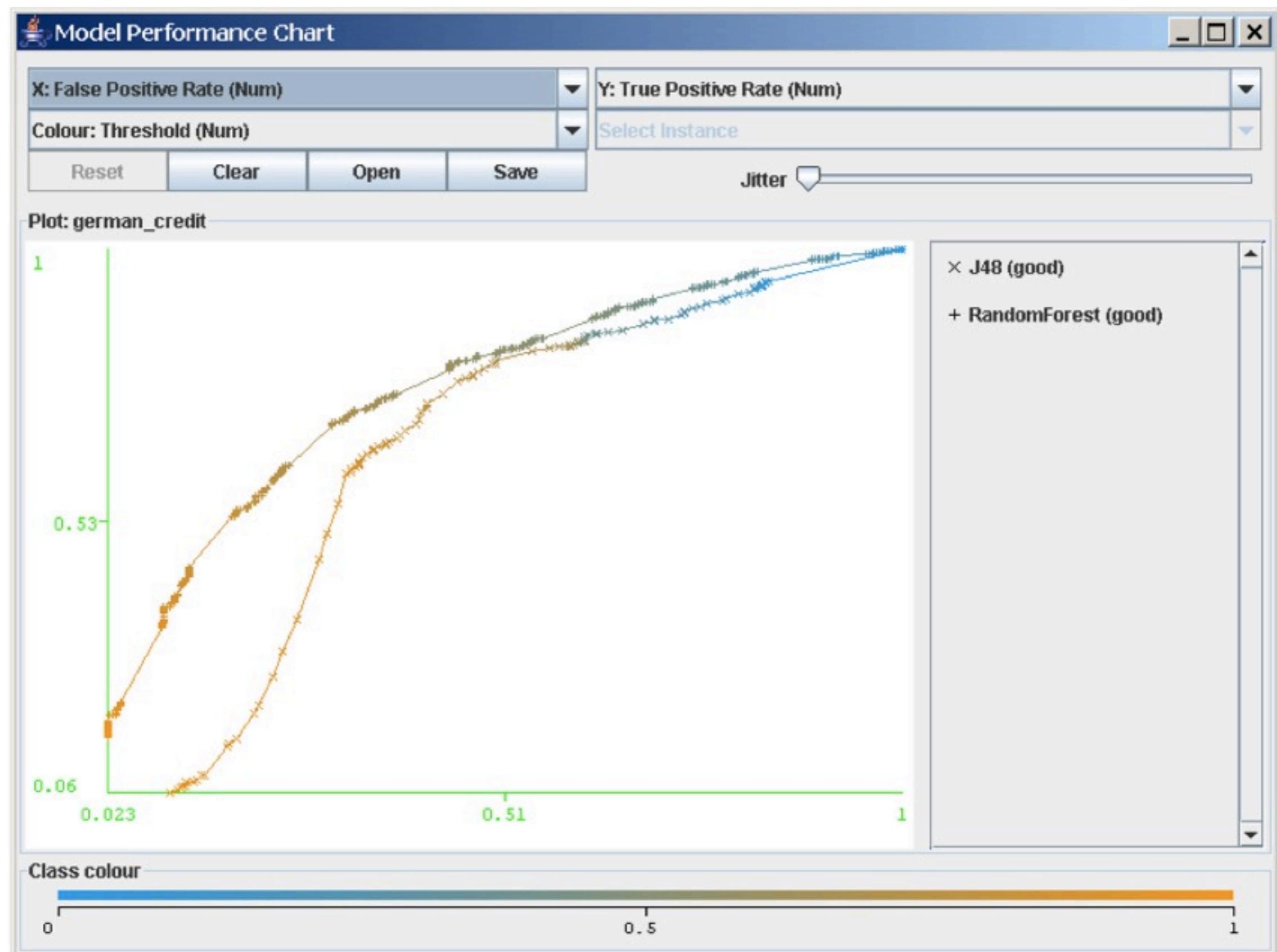
Esempio Cross-validated J48



Confrontare classificatori con curve ROC



Curve ROC



Apprendere incrementalmente

