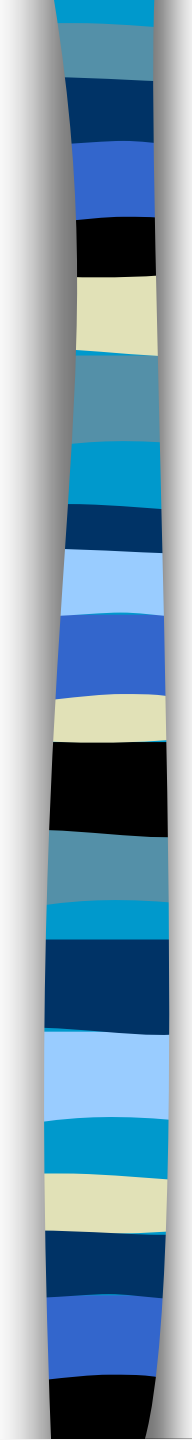


Il software Weka



Prof. Matteo Golfarelli

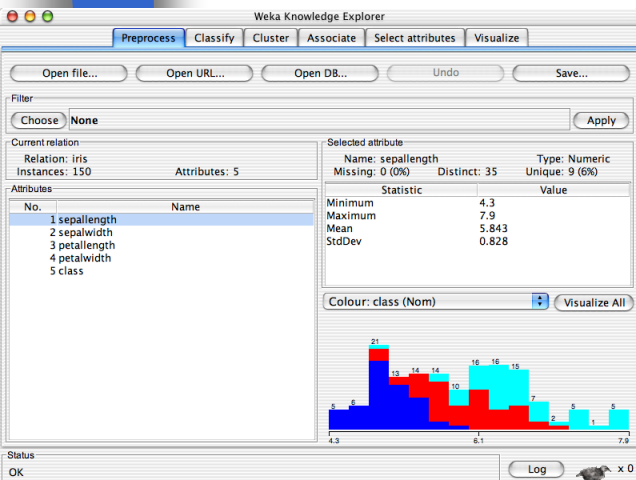
Alma Mater Studiorum - Università di Bologna



Weka

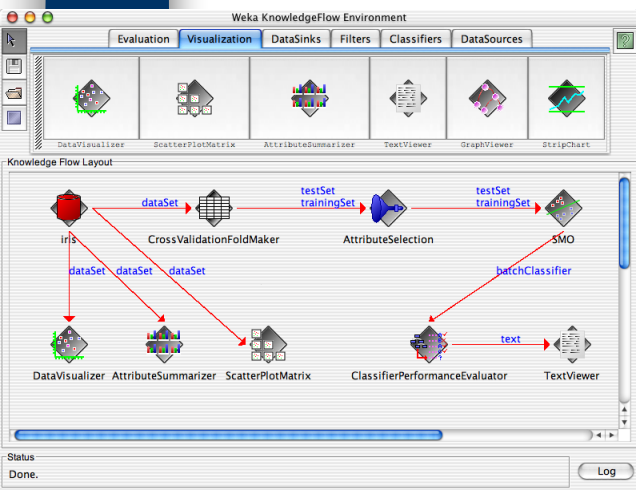
- Un software per il Data Mining/Machine learning scritto in Java e distribuito sotto la GNU Public License)
 - ✓ [Waikato Environment for Knowledge Analysis](#)
 - ✓ [Come scaricarlo](#)
- Utilizzato in ambito scientifico, didattico e applicativo
- Include:
 - ✓ Un insieme di tool per il pre-processing, algoritmi di apprendimento e metodi di valutazione
 - ✓ Interfaccia grafica
 - ✓ Un ambiente per comparare i risultati degli algoritmi di apprendimento
- Perché lo scegliamo
 - ✓ Ambiente low code
 - ✓ Organizzato sulla metodologia CRISP-DM

Weka



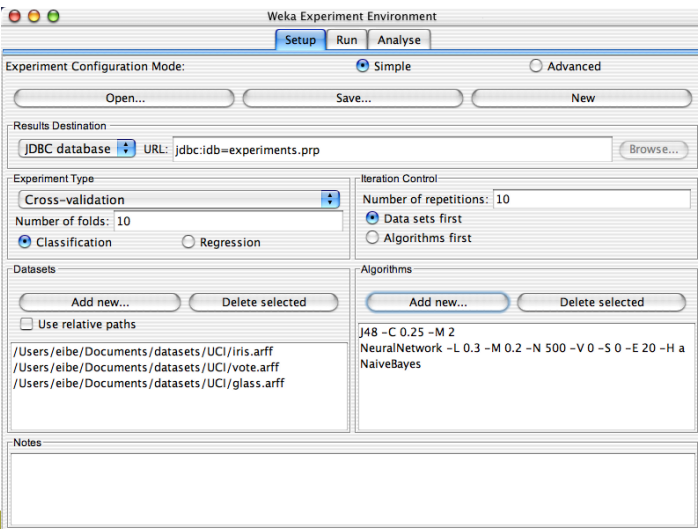
Explorer

Pre-processing
e creazione di modelli di learning



Experimenter

Ambiente per la comparazione
dei risultati su più algoritmi
di mining o su più data set

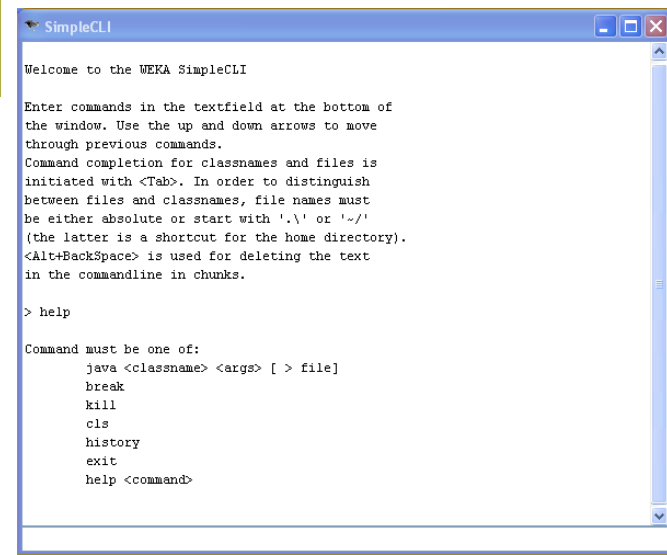


KnowledgeFlow

Creazione di flussi
complessi di funzioni Weka

SimpleCLI

Interfaccia testuale



Gestione dei dati

- Il principale tipo di dati con cui opera WEKA è l'Attribute – Relation file (ARFF file)
 - ✓ I file descrivono la relazione, gli attributi e i valori che questi possono contenere, i dati

@relation heart-disease-simplified

@attribute age numeric

@attribute sex { female, male}

@attribute chest_pain_type { typ_angina, asympt, non_anginal, atyp_angina}

@attribute cholesterol numeric

@attribute exercise_induced_angina { no, yes}

@attribute class { present, not_present}

@data

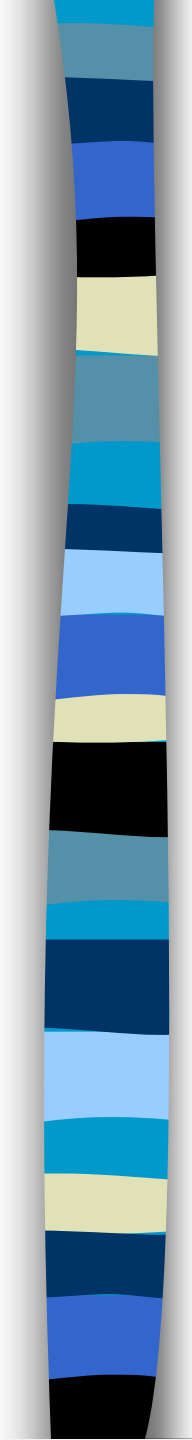
63,male,typ_angina,233,no,not_present

67,male,asympt,286,yes,present

67,male,asympt,229,yes,present

38,female,non_anginal,?,no,not_present

...



Gestione dei dati

- I classificatori costruiti (addestrati) possono essere salvati su file denominati modelli
 - ✓ Il salvataggio e il caricamento del modello si avviano con un right click del mouse sulla result list
- E quindi possibile ricaricare un modello e rieseguirlo su un nuovo data set
 - ✓ Il data set deve essere caricato utilizzando la voce “Test options→Supplied test set”

Il pre-processing

- Il pre-processing si realizza mediante filtri:
 - ✓ Discretizzazione
 - ✓ Normalizzazione
 - ✓ Resampling
 - ✓ Selezione di attributi
 - ✓ Trasformazione di attributi
- Il tab pre-processing consente inoltre di attivare funzionalità di visualizzazione delle distribuzioni dei dati rispetto all'attributo di classificazione o altro attributo

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

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

Filter
Choose **None** Apply

Current relation
Relation: weather
Instances: 14
Attributes: 5

Attributes
All None Invert Pattern

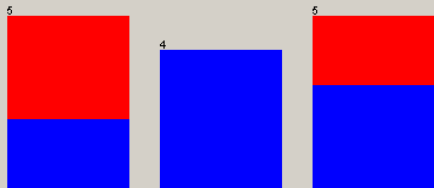
No.	Name
1	outlook
2	temperature
3	humidity
4	windy
5	play

Remove

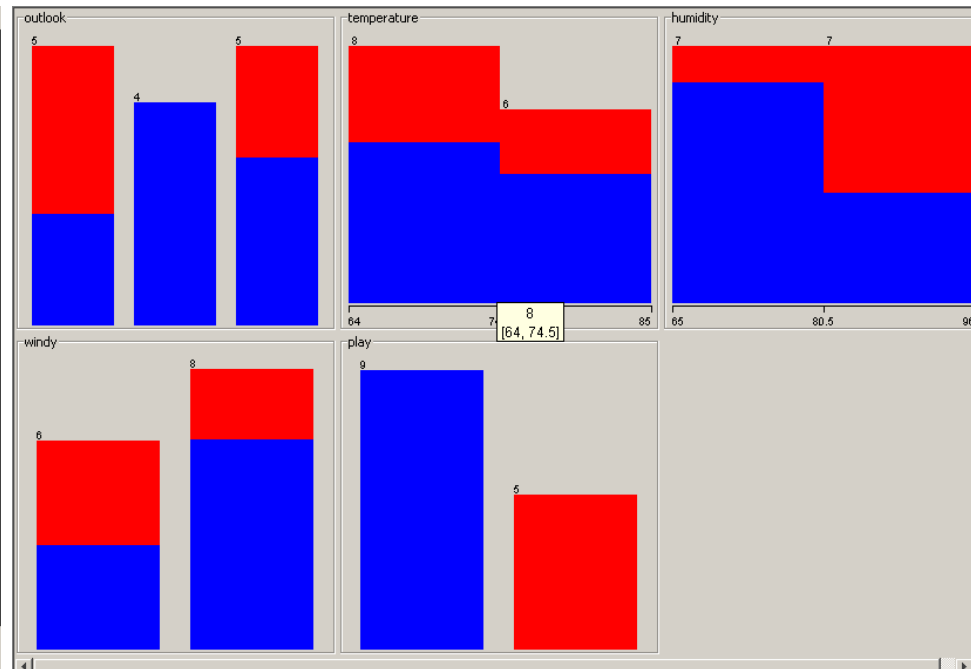
Selected attribute
Name: outlook
Missing: 0 (0%)
Distinct: 3
Type: Nominal
Unique: 0 (0%)

No.	Label	Count
1	sunny	5
2	overcast	4
3	rainy	5

Class: play (Nom) Visualize All



Status
OK Log x 0



II pre-processing

■ Il pre-processing si realizza mediante filtri:

✓ Discretizzazione

- **Discretize** (unsup.- attr.): An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.

✓ Normalizzazione

- **Normalize** (unsup.- attr.): normalizes all numeric values in the given dataset (apart from the class attribute, if set). The resulting values are by default in $[0,1]$ for the data used to compute the normalization intervals. But with the scale and translation parameters one can change that, e.g., with scale = 2.0 and translation = -1.0 you get values in the range $[-1,+1]$.
- **Standardize** (unsup.- attr.): standardizes all numeric attributes in the given dataset to have zero mean and unit variance (apart from the class attribute, if set).

II pre-processing

■ Il pre-processing si realizza mediante filtri:

✓ Resampling

- **Resample** (unsup.- inst.): produces a random subsample of a dataset using either sampling with replacement or without replacement.

✓ Trasformazione di attributi e valori

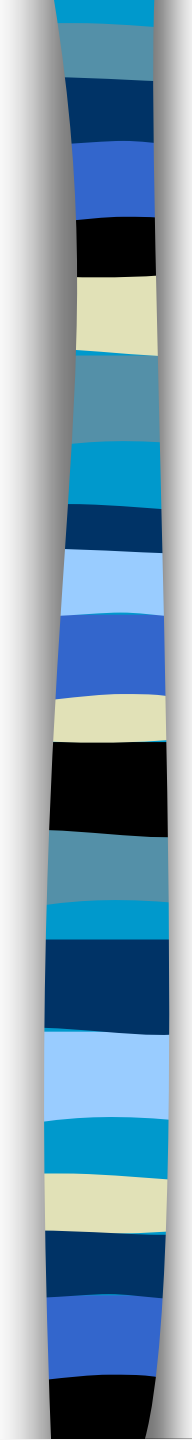
- **NominalToBinary** (unsup.- attr.): : converts all nominal attributes into binary numeric attributes.
- **AddNoise**: (unsup.- attr.): An instance filter that changes a percentage of a given attributes values. The attribute must be nominal. Missing value can be treated as value itself.

✓ Gestione valori mancanti

- **ReplaceMissingValues** (unsup.- attr.): replaces all missing values for nominal and numeric attributes in a dataset with the modes and means from the training data.

Il pre-processing

- **Selezione di attributi:** consente di identificare il subset di attributi che contenga la massima quantità di informazione
 - ✓ **CfsSubsetEval:** evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them
 - ✓ **ClassifierSubsetEval:** evaluates attribute subsets on training data (or a separate hold out testing set). Uses a classifier to estimate the 'merit' of a set of attributes.
- Lo spazio di ricerca (possibili subset degli attributi) può essere elevato è necessario definire un metodo di ricerca
 - ✓ Bestfirst
 - ✓ ExhaustiveSearch
 - ✓ GreedyStepWise
 - ✓ RandomSearch
 - ✓



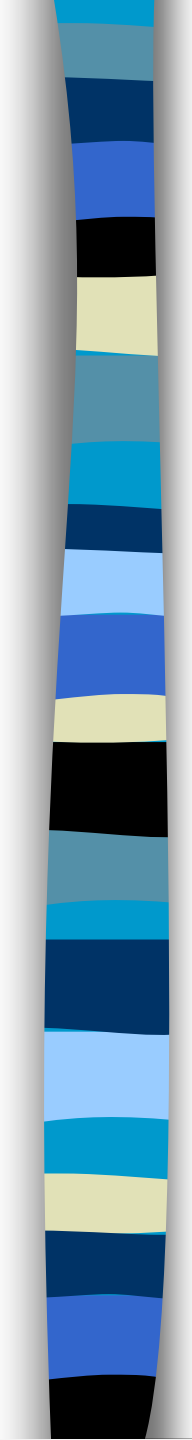
Visualizzazione

- Permette di visualizzare in un piano cartesiano le istanze del data set in funzione dei valori assunti da coppie di attributi
 - ✓ Il valore della classe è indicato tramite una diversa colorazione
- Dopo avere eseguito la classificazione è possibile utilizzare lo stesso tipo di visualizzazione per analizzare le istanze classificate in modo non corretto (rappresentate tramite rettangoli)
 - ✓ La visualizzazione si attiva con il tasto destro del mouse, selezionando la riga del log di esecuzione dell'algoritmo di learning
- Le tuple vere e proprie possono essere visualizzate
 - ✓ Dalla Weka GUI Choser
 - Tools→ArffViewer
 - ✓ Da Weka Explorer
 - Pulsante Edit

Classificazione

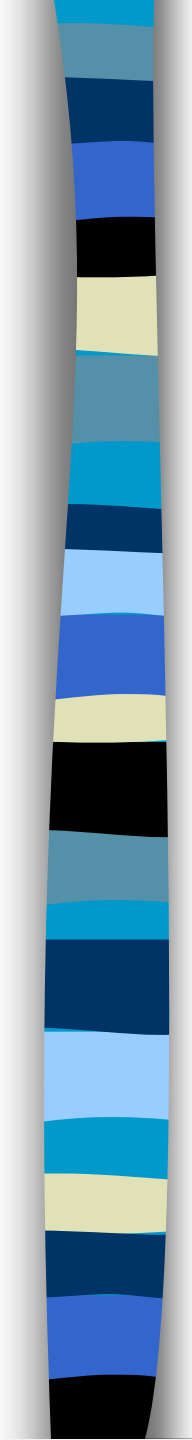
- Il tab di classificazione permette di:
 - ✓ Selezionare il tipo di algoritmo di classificazione e settarne i parametri
 - ✓ Definire le modalità per valutare la bontà del risultato
 - ✓ Visualizzare il risultato della classificazione

- Tra i molti classificatori a disposizione utilizzeremo:
 - ✓ Alberi decisionali (tree)
 - **J48**: implementazione dell'algoritmo C4.5
 - Unpruned (TRUE/FALSE) esegue o meno il post pruning
 - Confidence factor: valori piccoli accentuano l'effetto del post pruning
 - minNumObj : numero minimo di elementi in una foglia
 - **Decision Stump**: crea un albero decisionale a un livello
 - ✓ Classificatori basati su regole (rules)
 - **Jrip**: implementazione dell'algoritmo RIPPER
 - usePruning(TRUE/FALSE) esegue o meno il pruning
 - minNo: numero minimo di elementi coperti da una regola



Classificazione

- Il tab di classificazione permette di:
 - ✓ Selezionare il tipo di algoritmo di classificazione e settarne i parametri
 - ✓ Definire le modalità per valutare la bontà del risultato
 - ✓ Visualizzare il risultato della classificazione
- Tra i molti classificatori a disposizione utilizzeremo:
 - ✓ Classificatori instance based (lazy)
 - **IBK**: implementazione dell'algoritmo k-mediani
 - KNN: valore di k
 - nearestNeighbourSearchAlgorithm: tecnica utilizzata per la ricerca dell'NN
 - ✓ Classificatori Bayesiani (bayes)
 - **Naive Bayes**: implementazione dell'omonimo algoritmo



Classificazione

- Tra i molti classificatori a disposizione utilizzeremo:
 - ✓ Multi-classificatori (meta): utilizzano classificatori semplici per creare classificatori più complessi e potenti
 - Bagging
 - AdaBoost
 - RandomCommittee: il classificatore calcola la media dei risultati di più alberi decisionali ognuno dei quali utilizza un sottoinsieme random di attributi
 - CostSensitiveClassifier: rende cost sensitive il classificatore selezionato
- Test options: definiscono le modalità per verificare l'errore di classificazione:
 - ✓ Use training set
 - ✓ Supplied test set
 - ✓ Cross validation
 - ✓ Percentage split
- Attivando il flag More Options → Cost-Sensitive evaluation è inoltre possibile effettuare una **valutazione** dipendente dal peso degli errori

Analisi della classificazione

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

```
J48 pruned tree
```

```
-----  
wage-increase-first-year <= 2.5: bad (15.27/2.27)  
wage-increase-first-year > 2.5  
|   statutory-holidays <= 10: bad (10.77/4.77)  
|   statutory-holidays > 10: good (30.96/1.0)
```

Struttura dell'albero/regole con indicazione del numero di istanze del training set classificate correttamente e non. Le frazioni riguardano istanze con valori mancanti

```
Number of Leaves :      3
```

```
Size of the tree :      5
```

```
Time taken to build model: 0 seconds
```

```
=== Evaluation on training set ===
```

```
=== Summary ===
```

```
Correctly Classified Instances      50          87.7193 %  
Incorrectly Classified Instances    7           12.2807 %  
Kappa statistic                    0.745  
Mean absolute error                 0.195  
Root mean squared error             0.304  
Relative absolute error             42.6664 %  
Root relative squared error         63.6959 %  
Total Number of Instances          57
```

Statistiche riassuntive

```
=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.95	0.162	0.76	0.95	0.844	0.918	bad
	0.838	0.05	0.969	0.838	0.899	0.918	good
Weighted Avg.	0.877	0.089	0.896	0.877	0.88	0.918	

Statistiche dettagliate per classe

```
=== Confusion Matrix ===
```

```
 a  b  <-- classified as  
19  1 |  a = bad  
 6 31 |  b = good
```

Matrice di confusione