



Università
degli Studi
di Ferrara

Advanced School in Artificial Intelligence

Corso di Perfezionamento
Dipartimento di Ingegneria

Esercitazione pratica in Weka

Elisabetta Gentili, Damiano Azzolini

elisabetta.gentili1@unife.it, damiano.azzolini@unife.it

Anno Accademico 2022-23

WEKA

Waikato Environment for Knowledge Analysis



WEKA

The workbench for machine learning

Weka is tried and tested open source machine learning software that can be accessed through a graphical user interface, standard terminal applications, or a Java API. It is widely used for teaching, research, and industrial applications, contains a plethora of built-in tools for standard machine learning tasks, and additionally gives transparent access to well-known toolboxes such as [scikit-learn](#), [R](#), and [Deeplearning4j](#).

<https://www.cs.waikato.ac.nz/ml/weka/>

WEKA

Waikato Environment for Knowledge Analysis

Fornisce

- strumento di preprocessingamento di dataset
- strumento di trasformazione di dataset
- algoritmi di machine learning e data mining da applicare direttamente ad un proprio dataset: regressione, classificazione, clustering, scoperta di regole associative, selezione di attributi
- metriche di misurazione delle performance sul test set

Installazione

- https://waikato.github.io/weka-wiki/downloading_weka/
- Nella sezione **Developer version** scaricare il file d'installazione per il proprio sistema operativo. nei PC di laboratorio è già installata
- Per aprire Weka eseguire le istruzioni indicate per il proprio sistema operativo

Come fare Machine Learning/Data Mining



Come fare Machine Learning/Data Mining

1. Caricare i dati nell'Explorer (diversi formati possibili)
2. Selezionare l'algoritmo di apprendimento
 - pannello *Classify* per Decision trees e Random Forest
 - pannello *Cluster* per clustering
 - pannello *Associate* per regole associative
3. Valutare le performance (possibile per modelli predittivi)

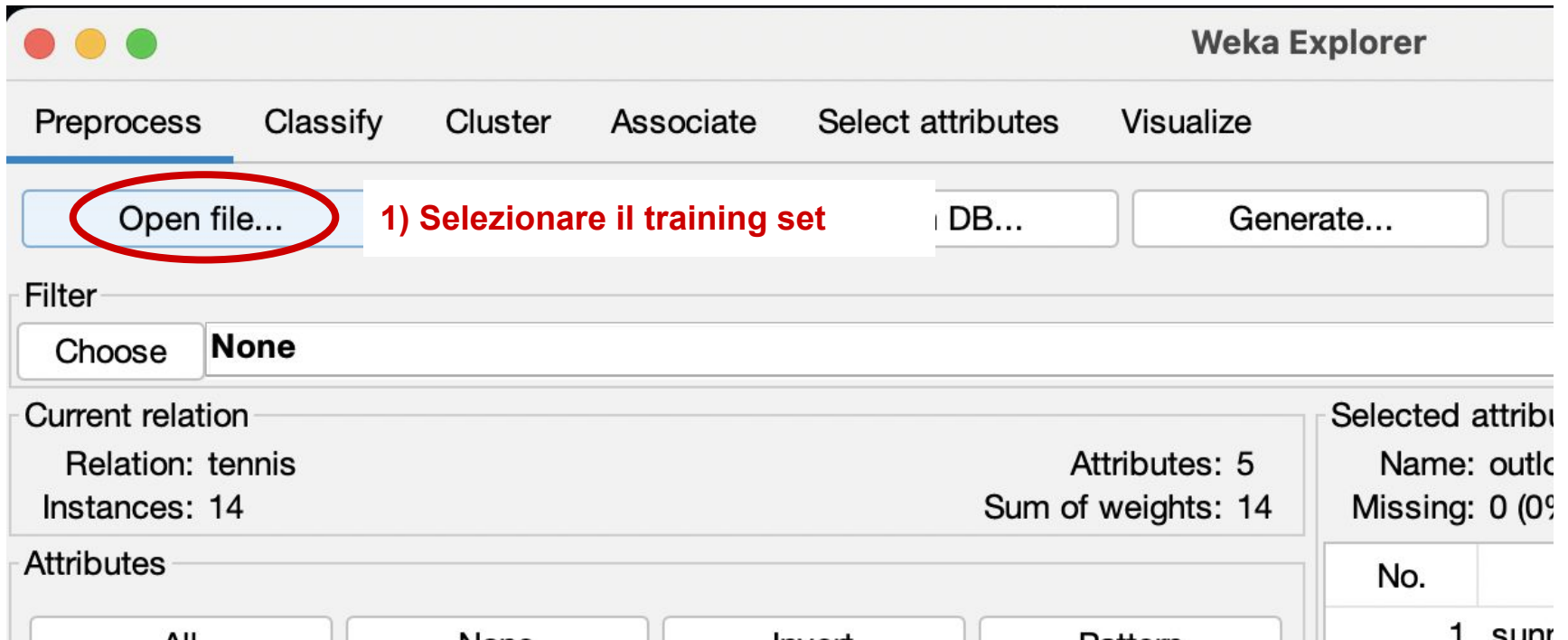
Formati di input

- Weka riconosce diversi formati per i dati in input, tra cui:
 - CSV (attenzione a non avere righe vuote a fine file)
 - ARFF, un formato di file di testo utilizzato specificatamente da Weka:

```
@relation weather
@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
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
```

Decision Tree

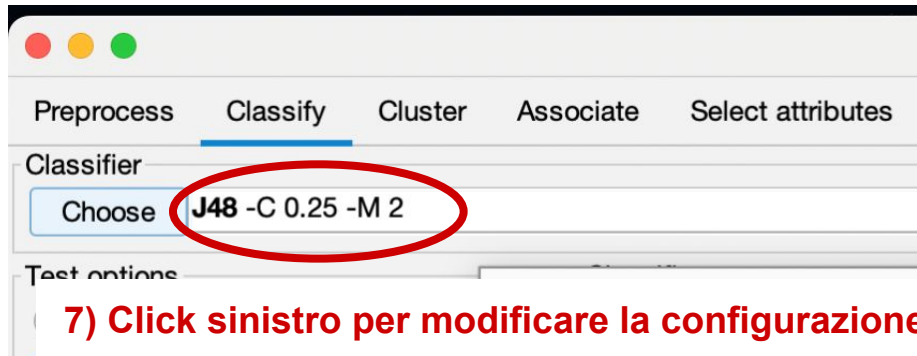


Decision Tree

The screenshot shows the WEKA software interface with several components and annotations:

- Top Bar:** Contains tabs for 'Preprocess', 'Classify' (circled in red), 'Visualize', and 'Auto-WEKA'. An annotation '2) Selezionare il pannello *Classify*' points to the 'Classify' tab.
- Classifier Section:** Includes a 'Classifier' dropdown menu with a 'Choose' button circled in red. An annotation '3) Selezionare l'algoritmo' points to this button.
- Test options Section:** Contains radio buttons for 'Use training set', 'Supplied test set' (selected), 'Cross-validation', and 'Percentage split'. Next to 'Supplied test set' is a 'Set...' button circled in red, with an annotation '4) Selezionare il test set' pointing to it. Below these are fields for 'Folds' (10) and 'Percentage split' (% 66), and a 'More options...' button.
- Test Instances Panel:** A panel on the right showing 'Relation: None', 'Instances: None', 'Attributes: None', and 'Sum of weights: None'. It has an 'Open file...' button circled in red, with an annotation '5) Selezionare il file tennis-test.arff' pointing to it. Below this is a 'Class' dropdown menu set to 'No class'.
- Target Attribute:** At the bottom left, a dropdown menu shows '(Nom) playTennis' circled in red, with an annotation '6) Selezionare l'attributo target playTennis' pointing to it.
- Buttons:** A 'Close' button is located at the bottom right of the 'Test Instances' panel.

Decision Tree

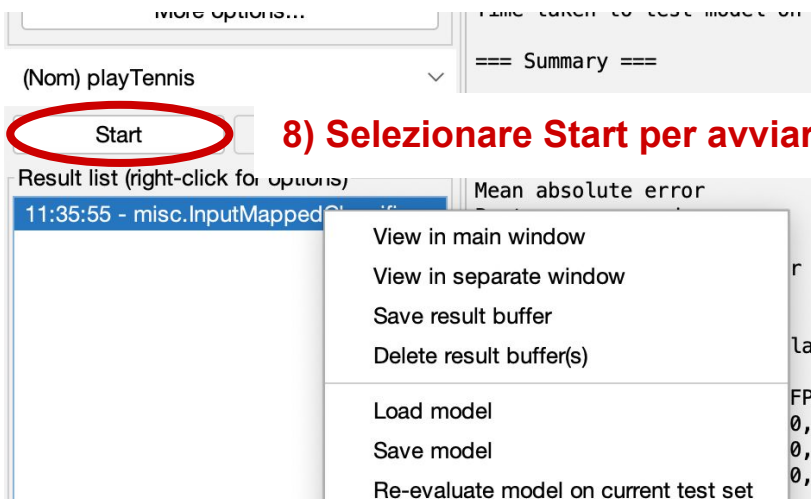


Capabilities

batchSize	100
binarySplits	False
collapseTree	True
confidenceFactor	0.25
debug	False
doNotCheckCapabilities	False
doNotMakeSplitPointActualValue	False
minNumObj	2
numDecimalPlaces	2
numFolds	3
reducedErrorPruning	False
saveInstanceData	False
seed	1
subtreeRaising	True
unpruned	False
useLaplace	False
useMDLcorrection	True

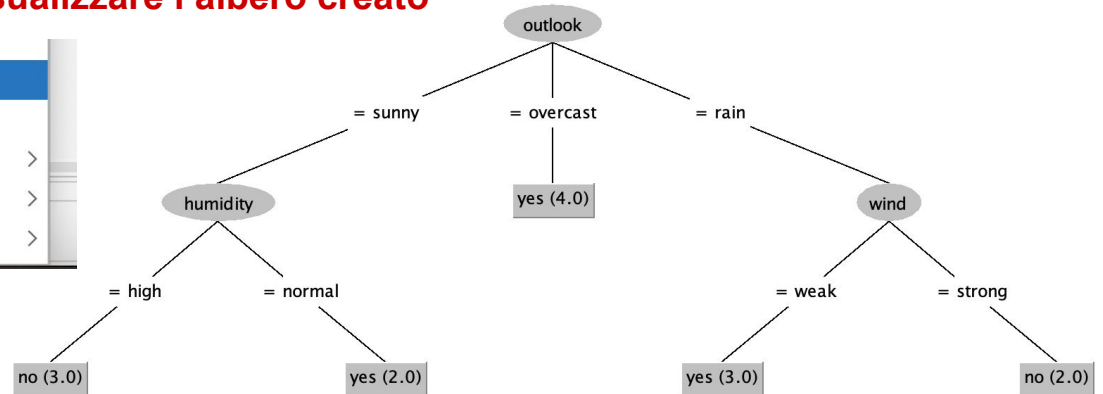
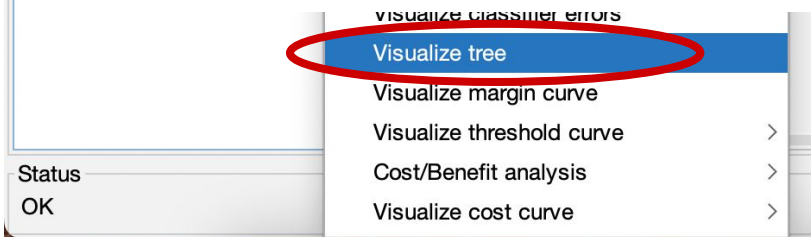
Open... Save... OK Cancel

Decision Tree - Training



8) Selezionare **Start** per avviare l'algoritmo sul training set

9) Selezionare **Visualize tree** per visualizzare l'albero creato



Decision Tree - Test

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	no
	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	yes
Weighted Avg.	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	

=== Confusion Matrix ===

```
a b  <-- classified as
2 0 | a = no
0 3 | b = yes
```

- $\text{Accuracy} = (\text{TP} + \text{TN}) / \text{TOT}$
- $\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$
- $\text{TPR (Recall)} = \text{TP} / (\text{TP} + \text{FN})$
- $\text{FPR} = \text{FP} / (\text{FP} + \text{TP})$
- $\text{F-Measure} = \text{Media armonica}(\text{Recall}, \text{Precision})$
- MCC → tasso statistico più affidabile che produce un punteggio elevato solo se la previsione ha ottenuto buoni risultati in tutte e quattro le categorie della matrice di confusione

Decision Tree - Cross validation

The screenshot shows the Orange3 software interface with the 'Classify' tab selected. The 'Classifier' section shows 'J48 -C 0.25 -M 2'. The 'Test options' section has 'Cross-validation' selected with 'Folds' set to 4. The target attribute is '(Nom) playTennis'. The 'Start' button is at the bottom. Red circles highlight these key elements, and red text annotations provide instructions in Italian.

4) Selezionare *Cross-validation* e il numero di fold

5) Selezionare l'attributo target playTennis

6) Selezionare Start per avviare l'algoritmo sul dataset

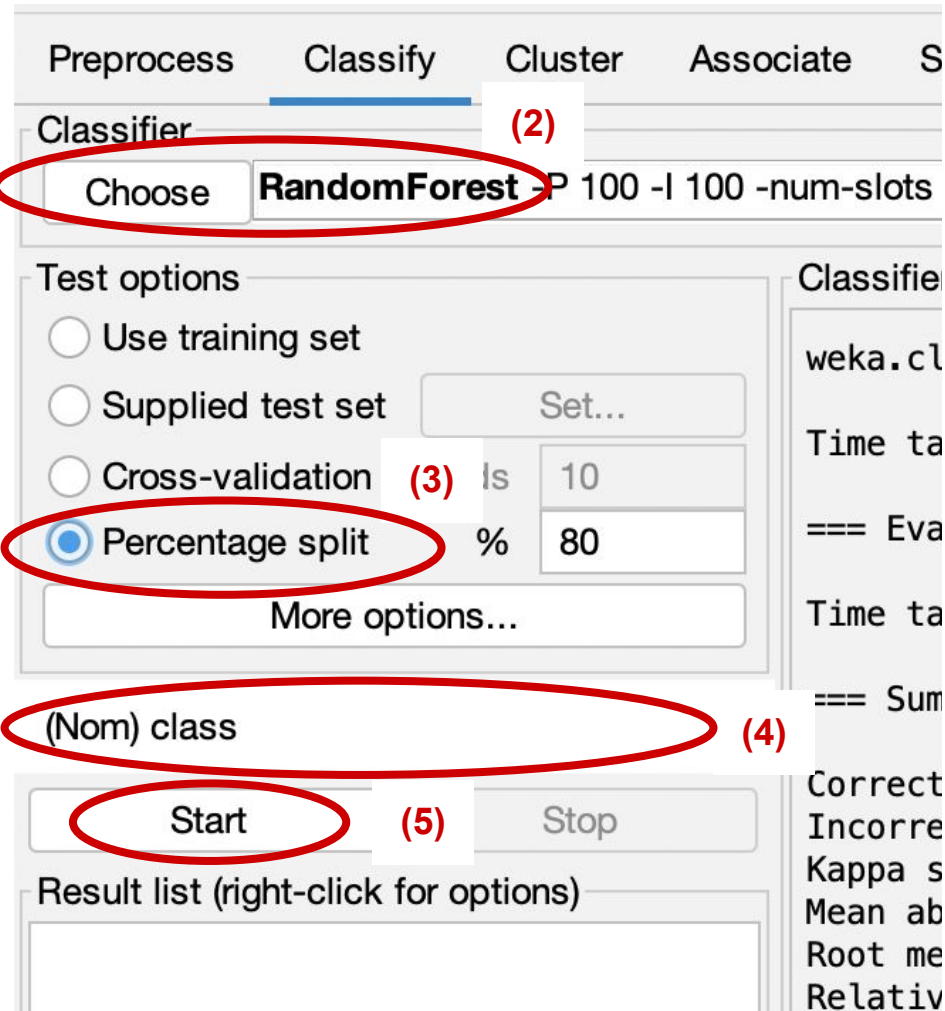
Esercizio su Decision Tree

1. Aprire in Weka il dataset tennis-training.arff, che contiene informazioni meteo sulle quali si basa la decisione di giocare a tennis o meno
2. Applicare l'algoritmo Decision Tree (J48)
3. Testare il modello sul file tennis-test.arff

ulteriori dataset reperibili sul sito:

<https://storm.cis.fordham.edu/~gweiss/data-mining/datasets.html>

Random Forest



1. Nel pannello *Preprocess* aprire il training set
2. Nel pannello *Classify* selezionare l'algoritmo
Choose>Tree>RandomForest
3. Nelle *test options* selezionare *Percentage split*
4. Selezionare l'attributo target *class*
5. Premere *Start*

Random Forest

Classifier

Choose **RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M**

Click sinistro per modificare la configurazione

Class for constructing a forest of random trees.

More
Capabilities

bagSizePercent 100

batchSize 100

breakTiesRandomly False

calcOutOfBag False

computeAttributeImportance False

debug False

doNotCheckCapabilities False

maxDepth 0

numDecimalPlaces 2

numExecutionSlots 1

numFeatures 0

numIterations 100

outputOutOfBagComplexityStatistics False

printClassifiers False

seed 1

storeOutOfBagPredictions False

Open... Save... OK Cancel

Random Forest

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,616	0,170	0,660	0,616	0,637	0,453	0,820	0,679	tested_positive
	0,830	0,384	0,801	0,830	0,815	0,453	0,820	0,886	tested_negative
Weighted Avg.	0,755	0,310	0,752	0,755	0,753	0,453	0,820	0,814	

=== Confusion Matrix ===

a	b	<-- classified as
165	103	a = tested_positive
85	415	b = tested_negative

- $\text{Accuracy} = (\text{TP} + \text{TN}) / \text{TOT}$
- $\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$
- $\text{TPR (Recall)} = \text{TP} / (\text{TP} + \text{FN})$
- $\text{FPR} = \text{FP} / (\text{FP} + \text{TP})$
- $\text{F-Measure} = \text{Media armonica}(\text{Recall}, \text{Precision})$
- MCC → tasso statistico più affidabile che produce un punteggio elevato solo se la previsione ha ottenuto buoni risultati in tutte e quattro le categorie della matrice di confusione

Esercizio su Random Forest

1. Aprire in Weka il dataset vote.arff, che contiene i voti di ciascun membro del Congresso della Camera dei Rappresentanti degli Stati Uniti
2. Numero di istanze: 435 (267 democratici, 168 repubblicani) ognuna delle quali con 16 attributi + la classe = 17 (tutti con valore Booleano)
3. Applicare l'algoritmo Random Forest