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 preprocessamento 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

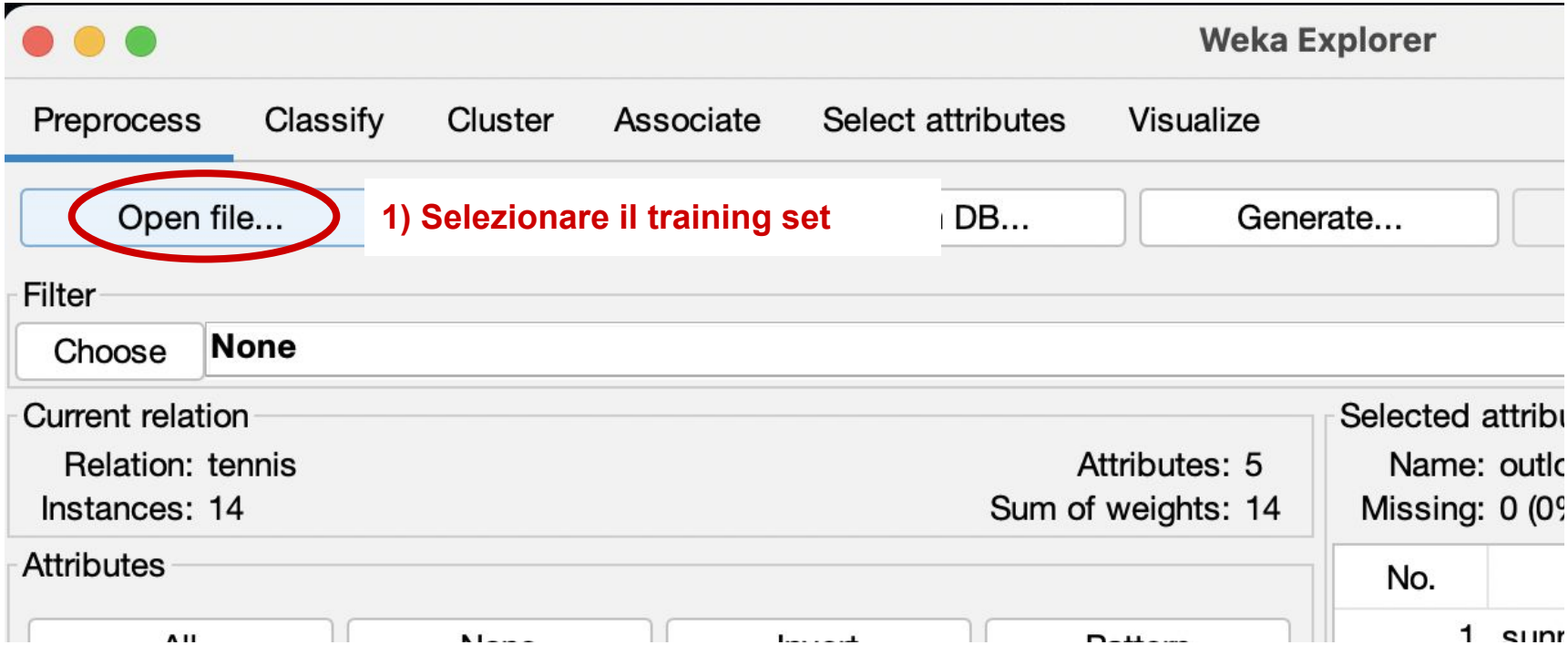
1. Caricare i dati nell'Explorer (diversi formati possibili)
2. Selezionare l'algoritmo di apprendimento
   - pannello *Classify* per Decision trees e Random Forest
   - pannello *Cluster* per clustering
   - pannello *Associate* per regole associative
3. Valutare le performance (possibile per modelli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

Formati di input

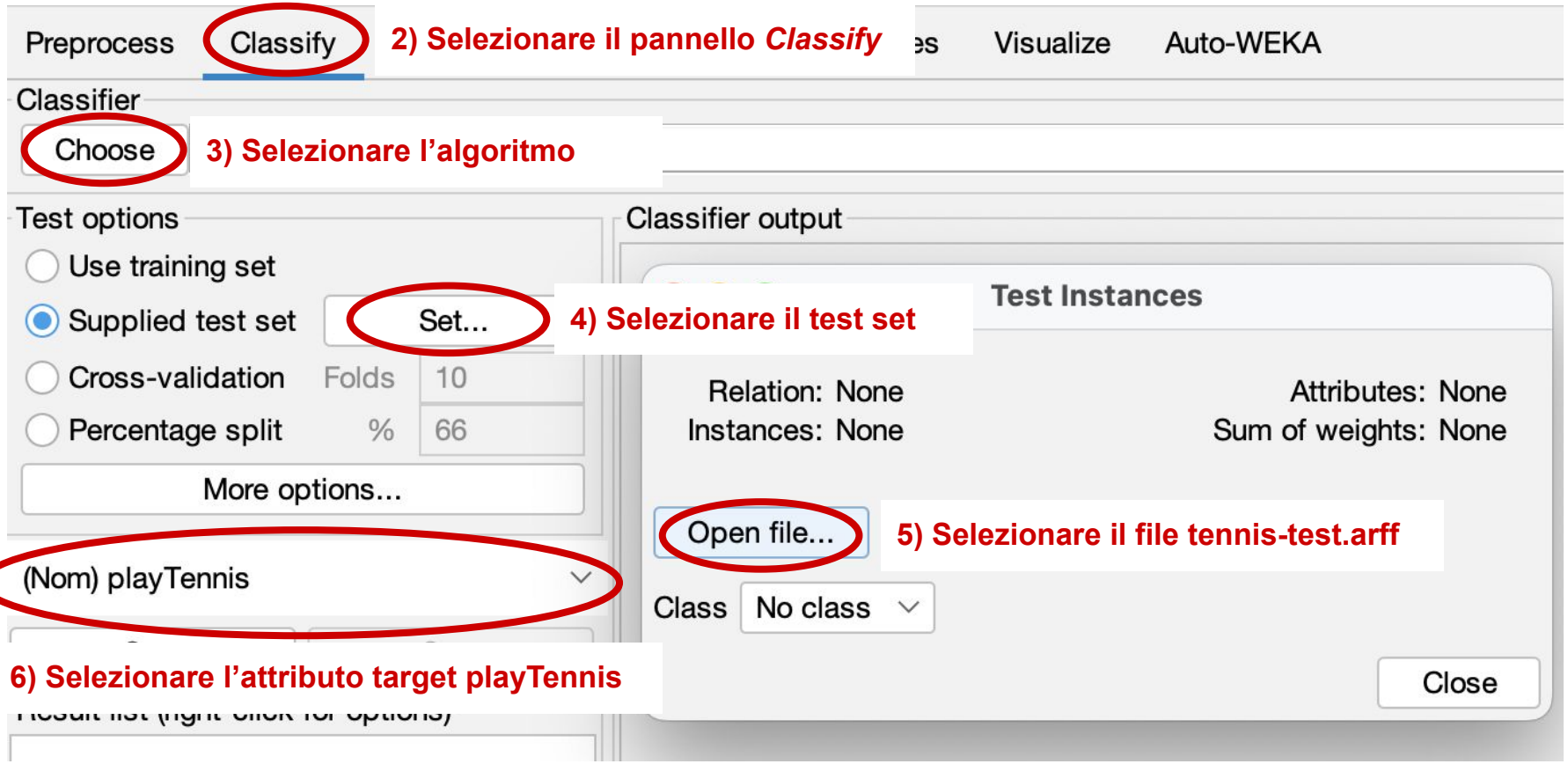
- Weka riconosce diversi formati per i dati in input, tra cui:
  - CSV (attenzione a non avere righe vuote a fine file)
  - ARFF, un formato di file di testo utilizzato specificatamente da Weka:

```
@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
```

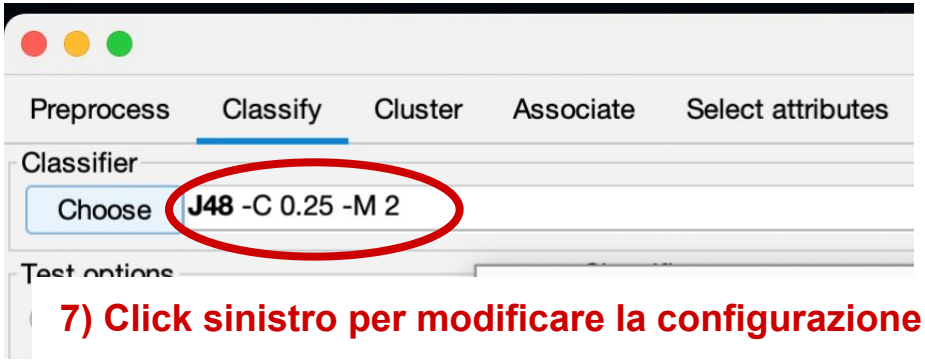
Decision Tree



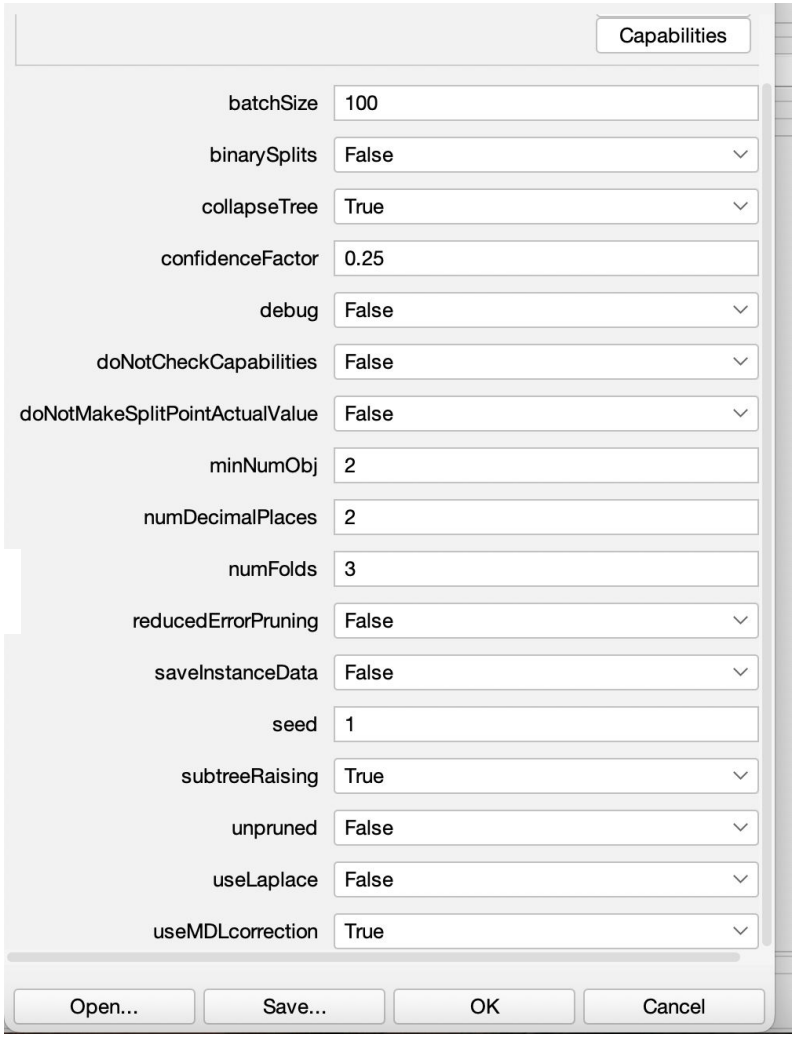
Decision Tree



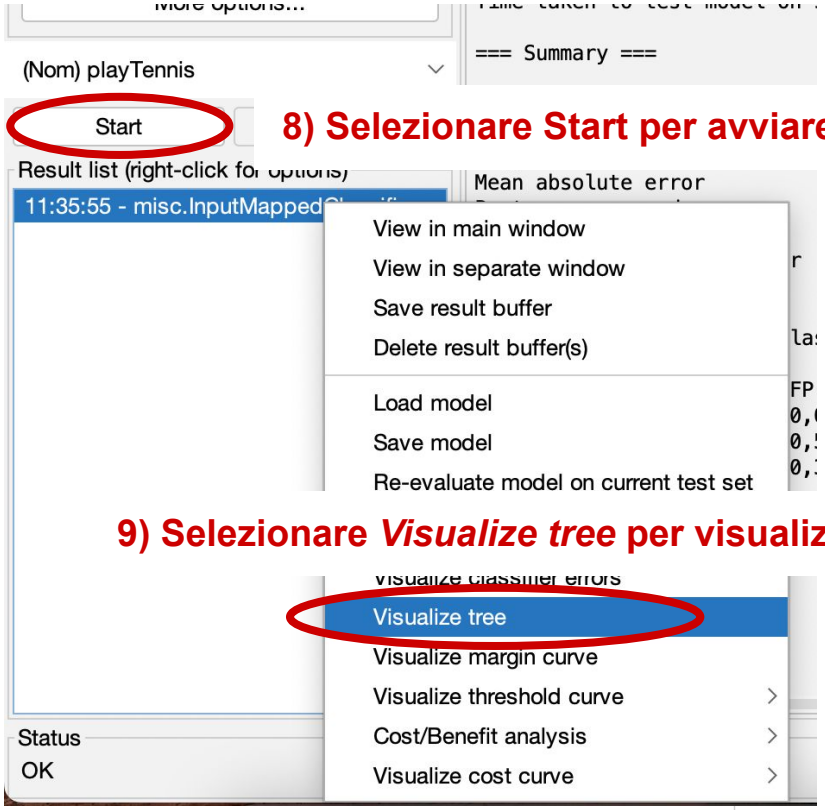
Decision Tree



7) Click sinistro per modificare la configurazione

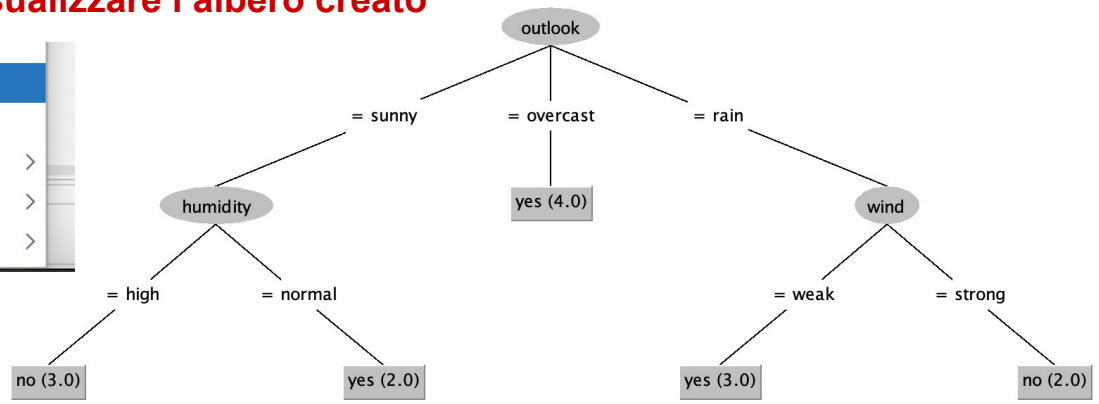


Decision Tree - Training



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

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

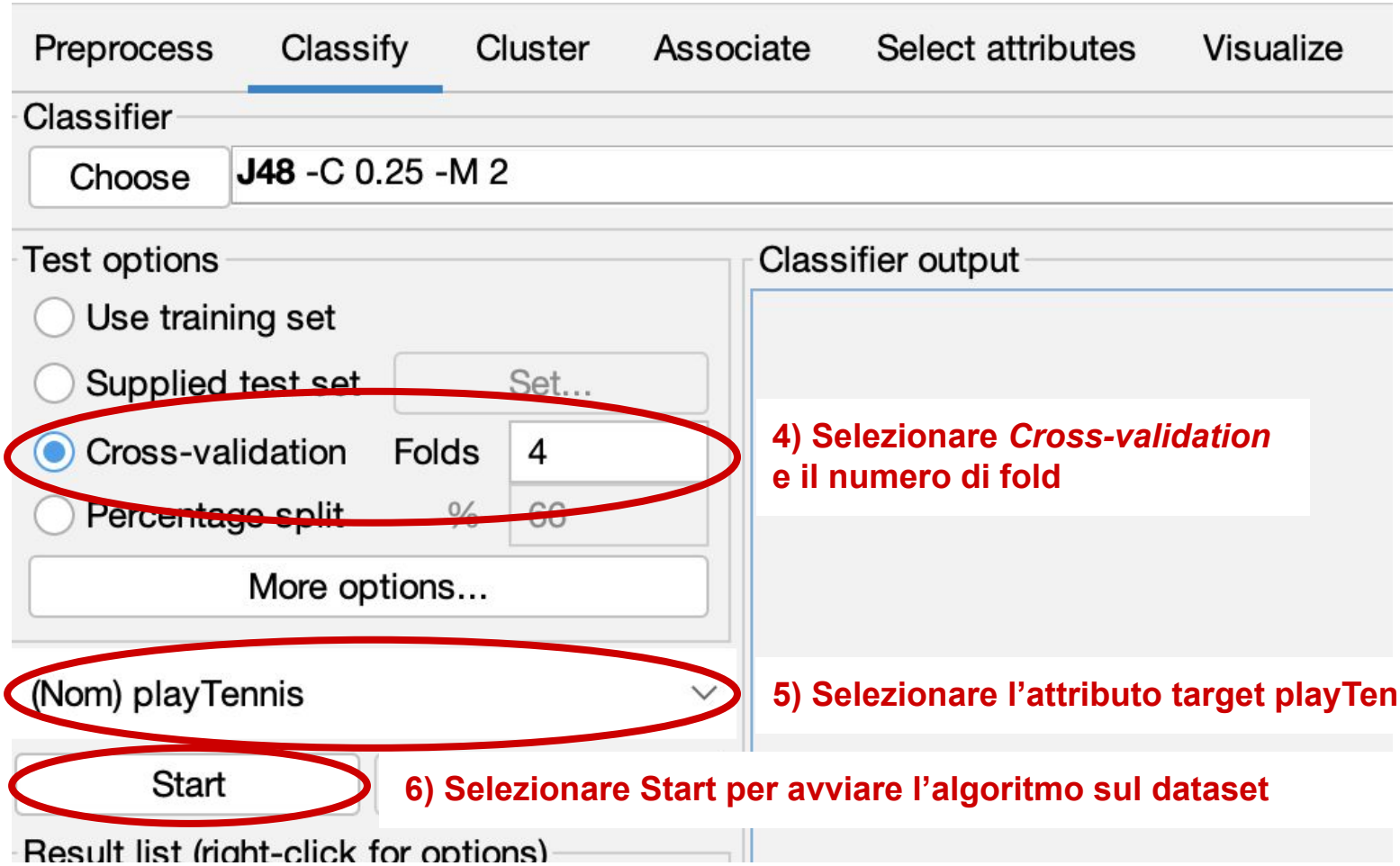


Decision Tree - Test

```
=== 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
```

- Accuracy = (TP+TN)/TOT
- Precision = TP/(TP+FP)
- TPR (Recall) = TP/(TP+FN)
- FPR = FP/(FP+TP)
- F-Measure = Media armonica(Recall, 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

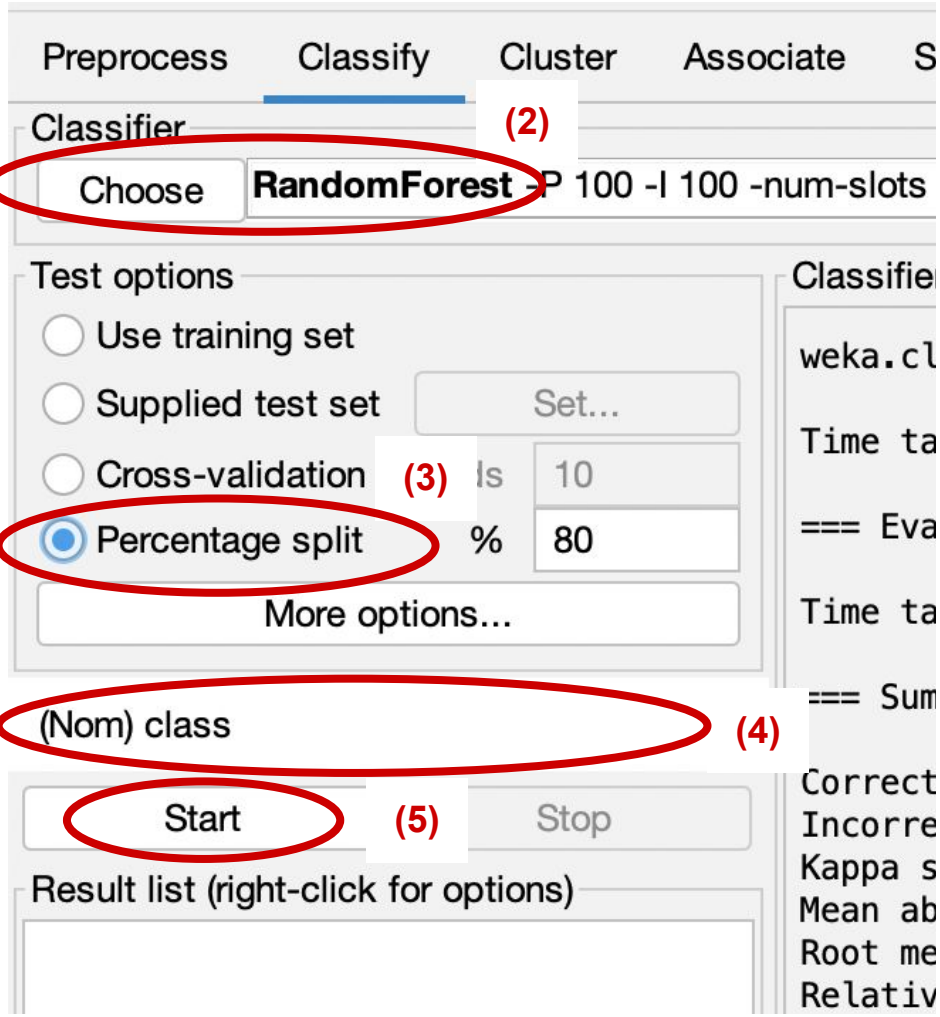


Esercizio su Decision Tree

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

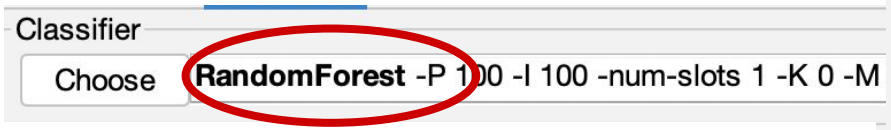
ulteriori dataset reperibili sul sito:
https://storm.cis.fordham.edu/~gweiss/data-mining/datasets.html

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



Click sinistro per modificare la configurazione

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
```

- Accuracy = (TP+TN)/TOT
- Precision = TP/(TP+FP)
- TPR (Recall) = TP/(TP+FN)
- FPR = FP/(FP+TP)
- F-Measure = Media armonica(Recall, 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