Università degli Studi di Cagliari
Corso di Laurea DSBAI

Web Analytics e Analisi Testuale

http://agile-group.org

A.A. 2019/2020

Ing. Marco Ortu
Via Porcell 4, primo piano
mail: marco.ortu@unica.it

Introduzione a WEKA

WEKA: Introduction

WEKA: Introduction

Weka è una raccolta di algoritmi di apprendimento statistico per attività di data mining, gli algoritmi possono essere applicati direttamente a set di dati o chiamati dal proprio codice Java.

Weka contiene strumenti per la pre-elaborazione dei dati, classificazione, regressione, clustering, regole di associazione e visualizzazione.

Weka:

http://www.cs.waikato.ac.nz/ml/weka/

Manuale:

http://transact.dl.sourceforge.net/sourceforge/weka/WekaManual-3.6.0.pdf

Download e installazione:

https://www.cs.waikato.ac.nz/ml/weka/downloading.html

WEKA: Introduction

Viene utilizzato principalmente in tre diversi modi.

- ➔ Utilizzando l'interfaccia grafica.
- ➔ Da riga di comando utilizzando il supporto CLI (Command Line Interface).
- ➔ Direttamente nel codice Java utilizzando le varie librerie.

Il supporto WEKA per la categorizzazione dei testi è impressionante.

Una caratteristica importante è che questo pacchetto supporta la tokenizzazione di un testo in termini di indicizzazione (words stems, collocations) e l'assegnazione di un peso ai termini di indicizzazione (vettorizzazione), un passaggio obbligatorio in quasi tutte le attività di classificazione del testo.

WEKA: ARFF files

Weka utilizza un formato particolare per rappresentare i dati che utilizza per il machine learning, chiamato ***ARFF,*** Attribute-Relation File Format.
Un file ARFF e' costituito da due parti un ***header*** contenente le informazioni sugli attributi (nome e tipo) .

```
% 1. Title: Iris Plants Database
%
% 2. Sources:
%      (a) Creator: R.A. Fisher
%      (b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
%      (c) Date: July, 1988
%
@RELATION iris

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

WEKA: ARFF files

Una parte ***data*** che contiene i dati nel formato specificato nell'header.

@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa

WEKA: ARFF files

➔ Il nome della relazione viene specificato nel formato:

@relation <relation-name>

➔ Gli attributi, ovvero le features su cui vengono addestrati i classificatori sono specificati nel formato:

@attribute <attribute-name> <datatype>

dove <datatype> può assumere solo i seguenti valori:

- numeric
- <nominal-specification>
- string
- date [<date-format>]

WEKA: ARFF files

➔ I dati sono specificati nel formato:

@data
4.4,?,1.5,?,Iris-setosa

Gli attributi mancanti vengono indicati utilizzando un punto interrogativo ***?***.

I valori di stringa e attributi nominali sono case sensitive e quelli che contengono spazi devono essere quotati, come segue:

@relation LCCvsLCSH

@attribute LCC string
@attribute LCSH string

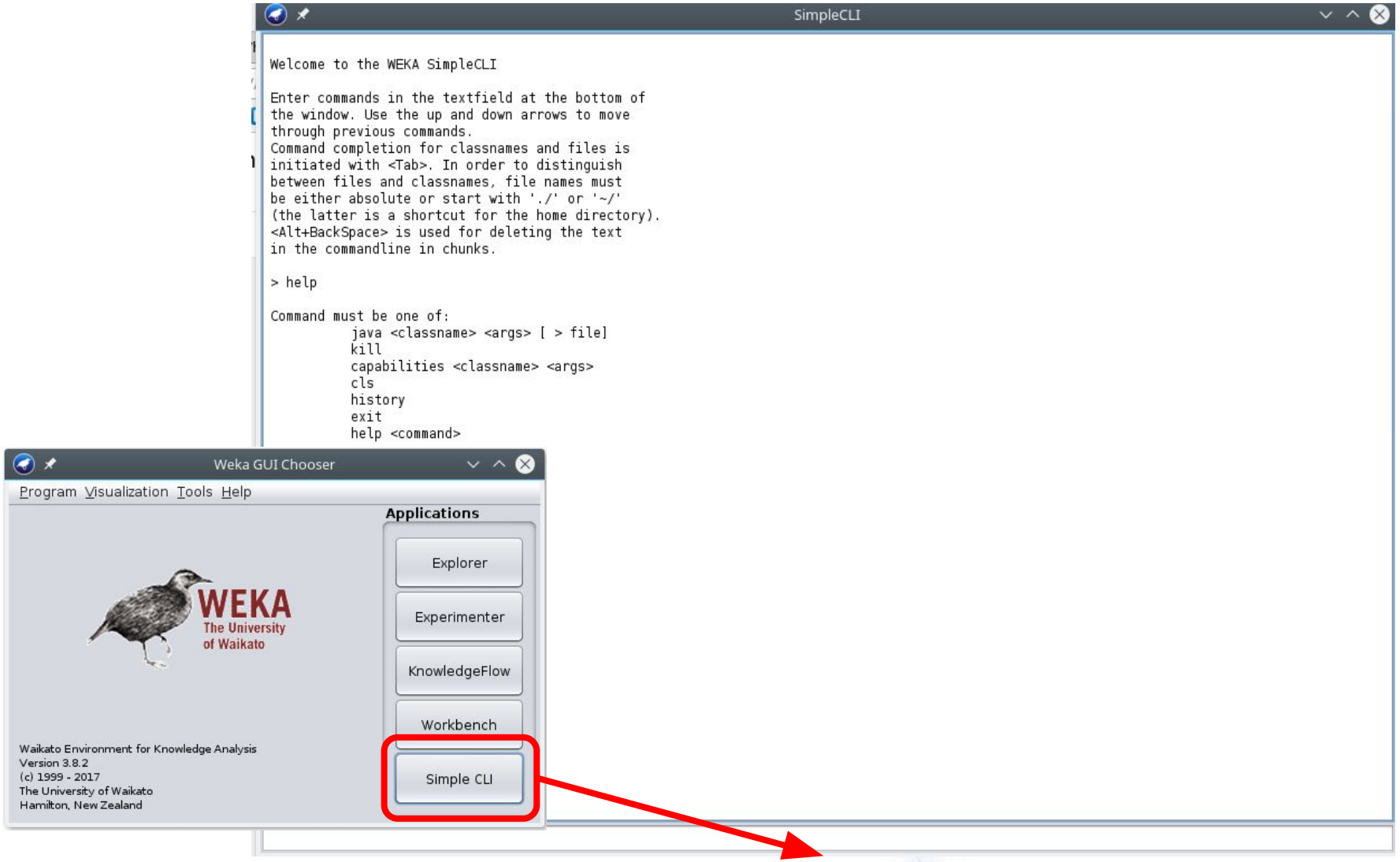
@data
AG5, 'Encyclopedias and dictionaries.;Twentieth century.'
AS262, 'Science -- Soviet Union -- History.'

WEKA: ARFF files

Weka permette di trasformare direttamente un cartella contenente file di testo in formato ARFF. Ad esempio data la seguente struttura di cartelle possiamo utilizzare la CLI per convertirla in un file ARFF.

```
+- text_example
   |
   +- class1
   |  |
   |  + file1.txt
   |  |
   |  + file2.txt
   |  |
   |  ...
   |
   +- class2
   |  |
   |  + another_file1.txt
   |  |
   |  + another_file2.txt
```

WEKA: ARFF files



WEKA: ARFF files

Utilizzando il seguente comando nella CLI di Weka viene creato un file ARFF a partire dalla struttura delle cartelle, ogni sottocartella diventa una classe e il testo contenuto un ogni file diventa un attributo.

java weka.core.converters.TextDirectoryLoader -dir text_example > text_example.arff

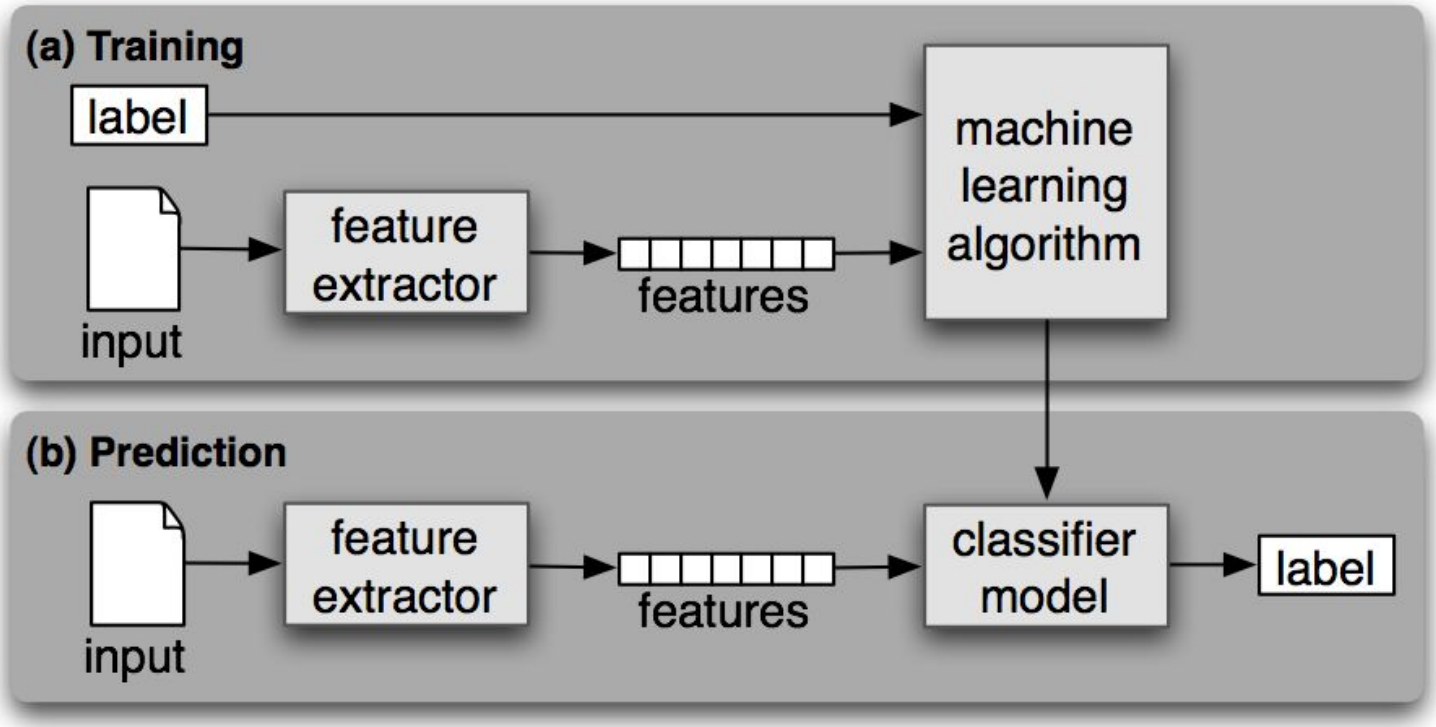
Utilizzando il seguente comando nella CLI di Weka invece viene trasformato un file csv in formato ARFF.

java weka.core.converters.CSVLoader file.csv > file.arff

WEKA: Text Mining

Weka permette di utilizzare delle ***pipelines*** per automatizzare l'estrazione delle features (attributi nella terminologia di Weka) dal testo puro.

Questo processo di tokenizzazione e indicizzazione viene ottenuto utilizzando un filtro molto flessibile denominato ***StringToWordVector***.



WEKA: Text Mining

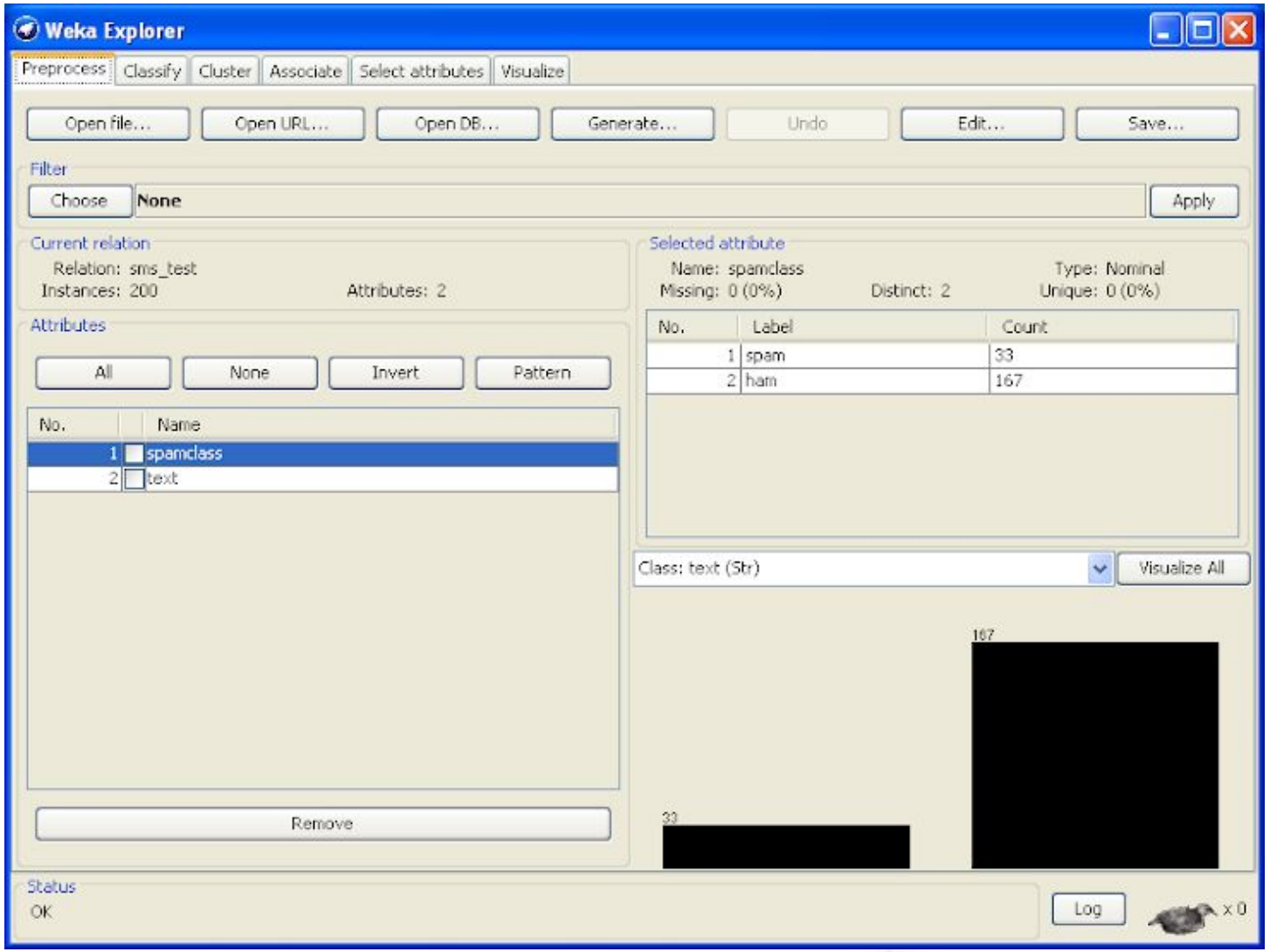
Utilizziamo un dataset di esempio che contiene SMS classificati come legittimi (ham) oppure come spam. Il file ARFF ha la seguente struttura:

@relation sms_test

@attribute spamclass {spam,ham}
@attribute text String

@data
ham,'Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...'
ham,'Ok lar... Joking wif u oni...'
spam,'Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C\'s apply 08452810075over18\'s'

WEKA: Text Mining



WEKA: Text Mining

Il punto è che i messaggi sono presenti come attributi di stringa, quindi bisogna tokenizzarli in parole per consentire agli algoritmi di apprendimento di indurre i classificatori regole come:

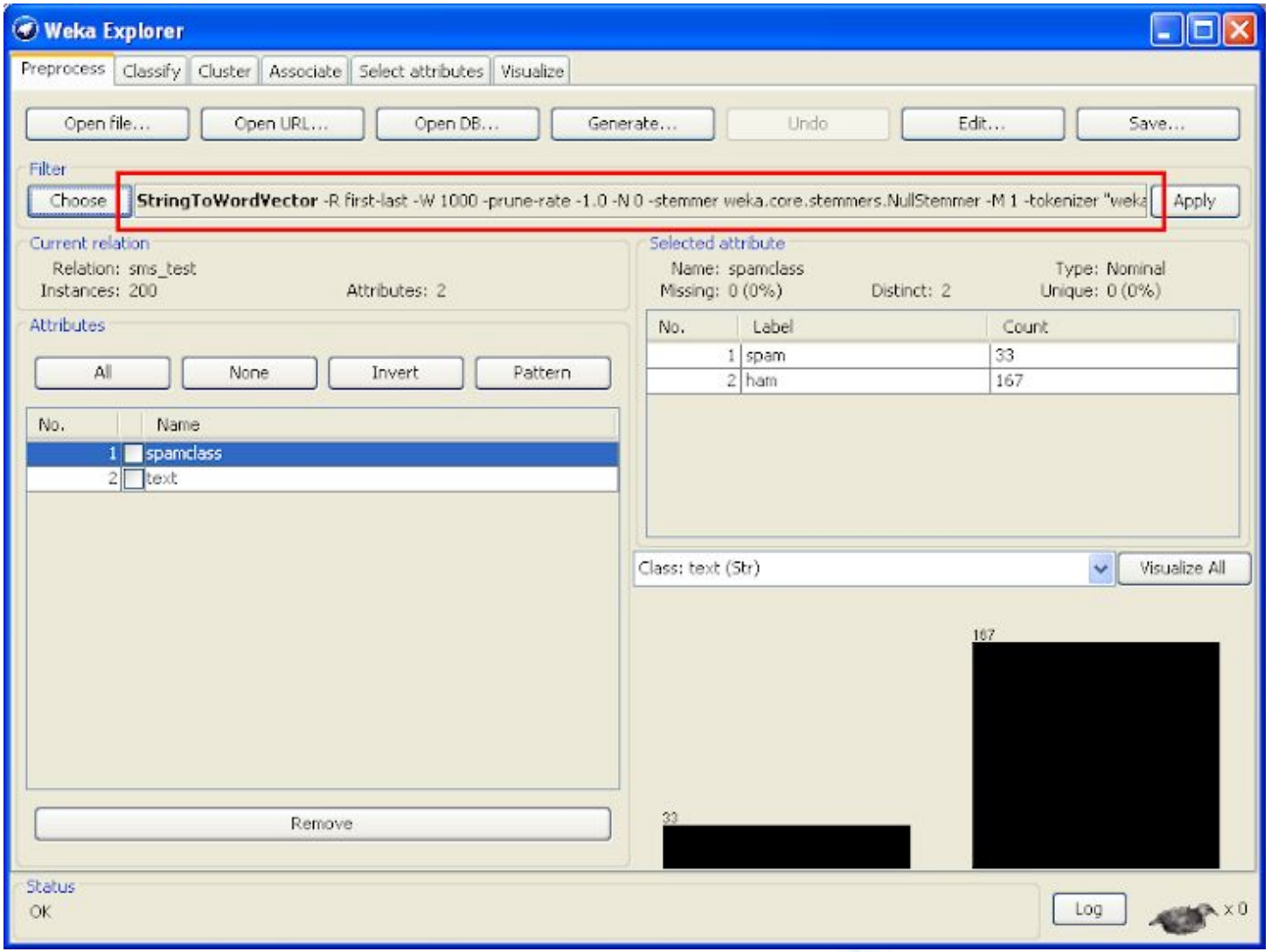
if ("urgent" **in** message) **then** class(message) == spam

Qui è dove il filtro ***StringToWordVector*** viene in aiuto.

Puoi selezionarlo semplicemente facendo clic sul pulsante "Choose" nell'area "Filtrer" e sfogliando le cartelle sull'attributo "weka> filters> unsupervised>".

Una volta selezionato, dovresti essere in grado di vedere qualcosa di simile a questo:

WEKA: Text Mining

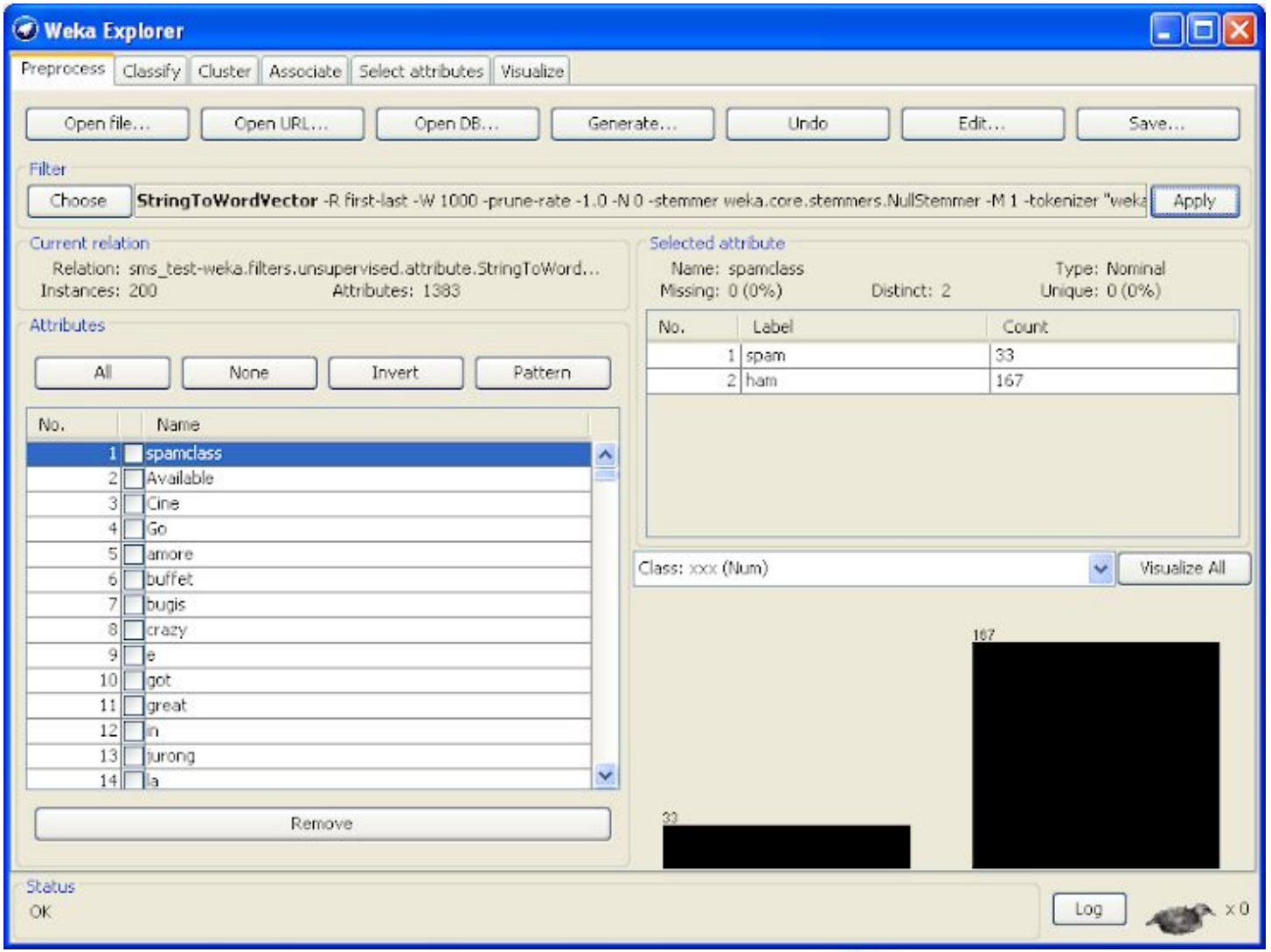


WEKA: Text Mining

Facendo clic sul nome del filtro, si ottengono molte opzioni.

Si può semplicemente applicare il filtro con le opzioni predefinite per ottenere una raccolta indicizzata di 850 messaggi ham e 152 messaggi spam e 2319 token di indicizzazione (oltre all'attributo class).

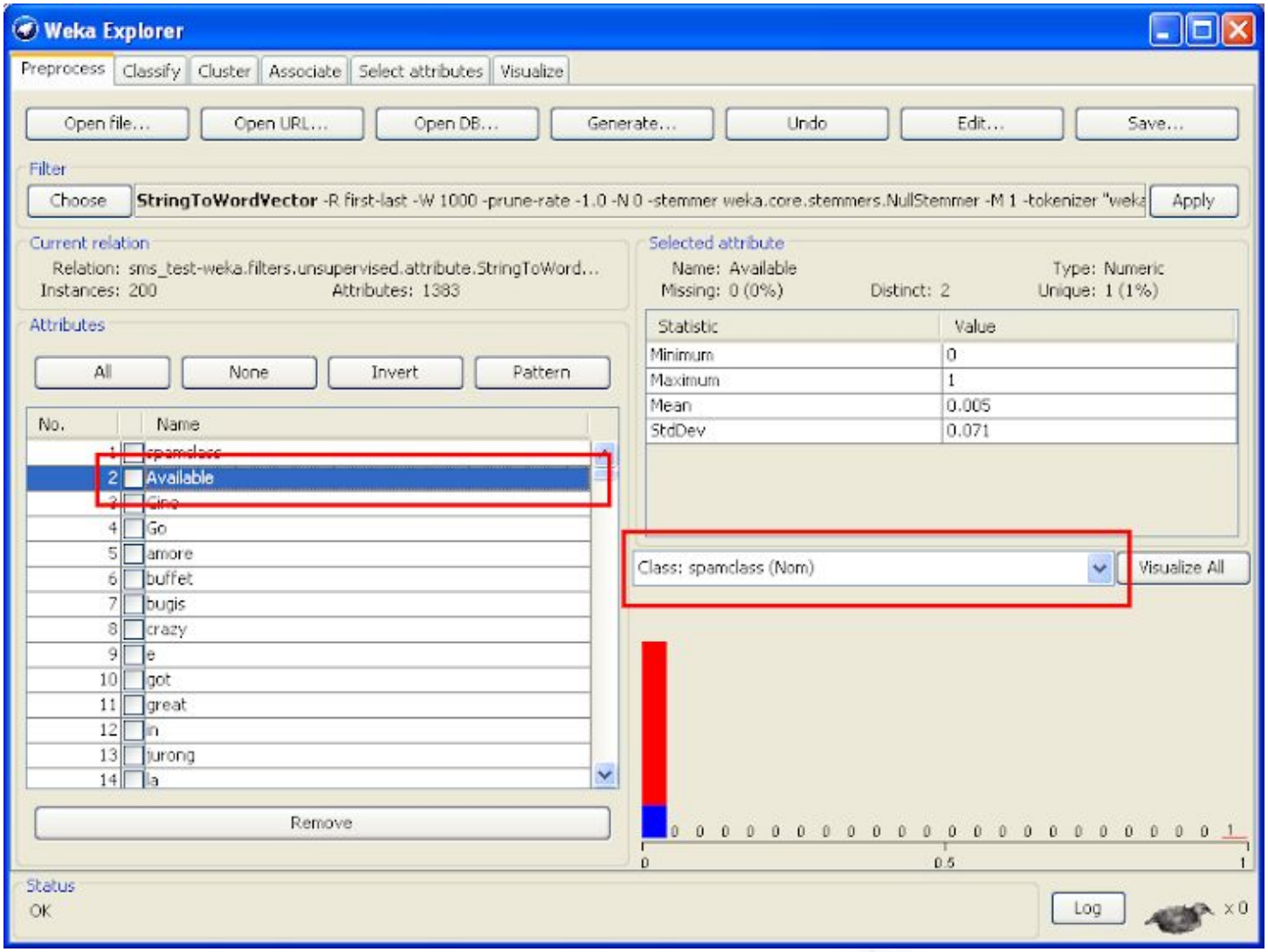
WEKA: Text Mining



WEKA: Text Mining

Per visualizzare i colori che mostrano la distribuzione di attributi (token) in base alla classe, selezionare l'attributo "class" come attributo di classe nell'area in basso a sinistra di WEKA Explorer.

WEKA: Text Mining

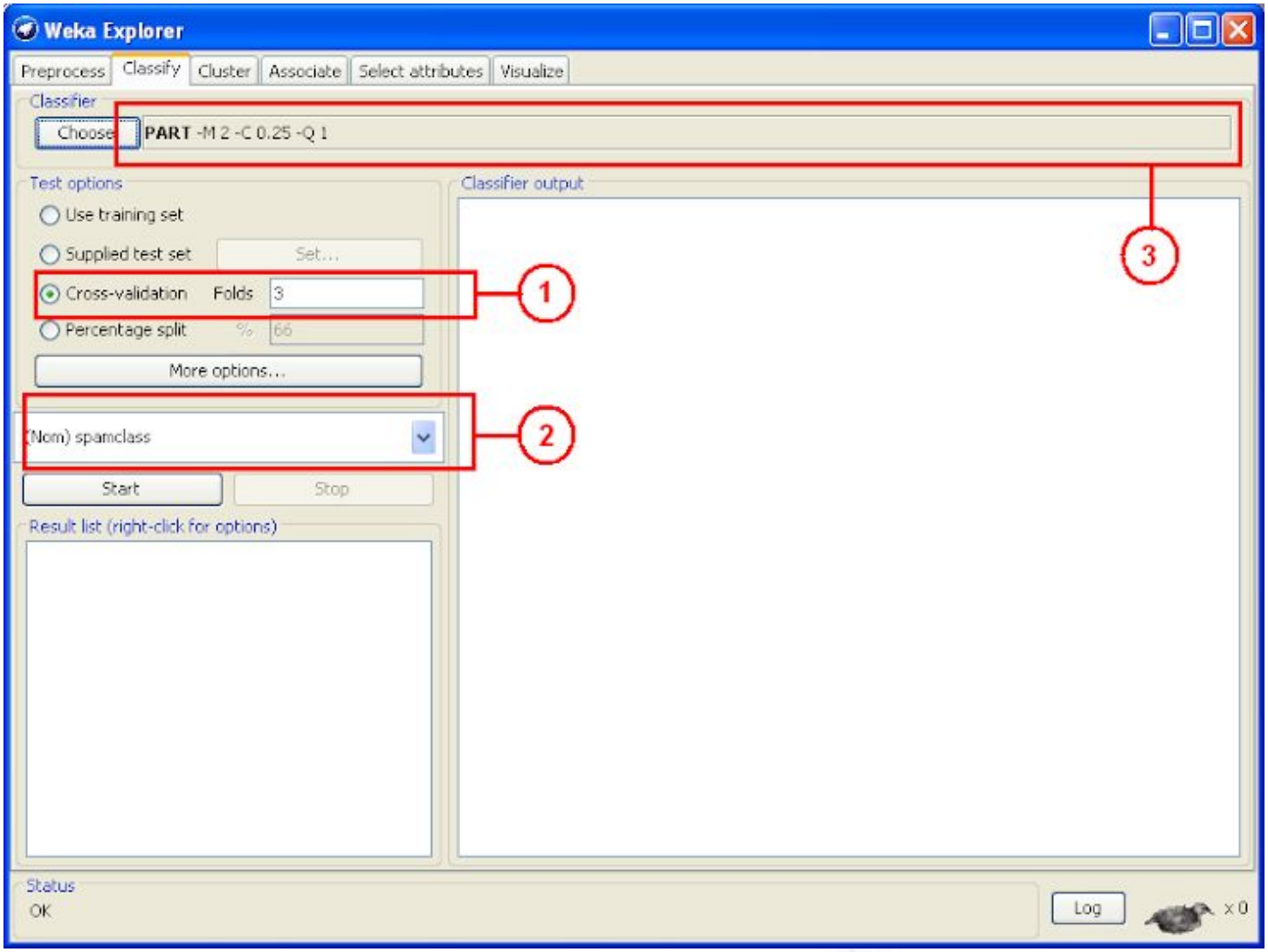


WEKA: Text Mining

Ora si possono fare gli esperimenti nella scheda Classify.

1. Selezionare la ***cross-validation*** usando 3 ***folds***,
2. Puntare sull'attributo appropriato da usare come classe (che è ***spamclass***).
3. Selezionare il ***rule learner*** chiamato **PART** nell'area del classificatore. Si trova nella cartella "weka> classifiers>rules" quando fai clic sul pulsante "Choose" nell'area "Classifier".

WEKA: Text Mining



WEKA: Text Mining

Il metodo di valutazione selezionato, cross-validation, incarica WEKA di dividere il dataset in 3 folds ed eseguire tre esperimenti.

Ogni esperimento viene eseguito utilizzando due delle folds per il training e il rimanente per testare il classificatore addestrato.

Le folds sono campionate casualmente, il modo che ogni messaggio da classificare appartiene solo a una di esse e la distribuzione della classe (50% nel nostro esempio) è conservata all'interno di ogni folds.

WEKA: Text Mining

Quindi, cliccando sul pulsante "Start", otterremo l'output del nostro esperimento, con il classificatore addestrato sull'intera collezione, i valori per le tipiche metriche di accuratezza mediate sui tre esperimenti, insieme alla matrice di confusione. Il classificatore addestrato sull'intero dataset.

WEKA: Text Mining

PART decision list

---

or <= 0 AND

to <= 0 AND

2 <= 0: **ham** (119.0/3.0)

£1000 <= 0 AND

FREE <= 0 AND

text <= 0: **ham** (26.0/2.0)

Number of Rules : 4

WEKA: Text Mining

Questa notazione può essere letta come:

if (("or" **not in** message)

and ("to" **not in** message)

and ("2" **not in** message))

then class(message) == ham

Nell'output viene restituita anche la ***confusion-matrix:***

```
=== Confusion Matrix ===

a b <-- classified as

17   16   | a = spam

12   155 | b = ham
```

WEKA: Text Mining

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

                 TP Rate   FP Rate   Precision   Recall   F-Measure   MCC      ROC Area   PRC Area   Cl
                 0,985     0,368     0,937       0,985    0,960       0,710    0,871      0,961      0
                 0,632     0,015     0,881       0,632    0,736       0,710    0,871      0,715      1
Weighted Avg.    0,931     0,315     0,929       0,931    0,926       0,710    0,871      0,924
```

WEKA: Text Mining

Se un attributo compare in uno solo dei messaggi in quale fold è?

Quando è su una fold di addestramento, la usiamo per il training (facendo sì che il classificatore cerchi di generalizzare da un token che non si trova nella raccolta di test).

E quando è sulla fold di test, il classificatore non dovrebbe nemmeno saperlo!

Inoltre, cosa succede con gli attributi che sono altamente predittivi per la raccolta completa (in base alle loro statistiche quando si calcola, ad esempio, la metrica di information gain)?

Possono avere statistiche peggiori (o migliori) quando non viene visto un sottoinsieme delle loro occorrenze, in quanto possono trovarsi nella raccolta di test!

WEKA: Text Mining

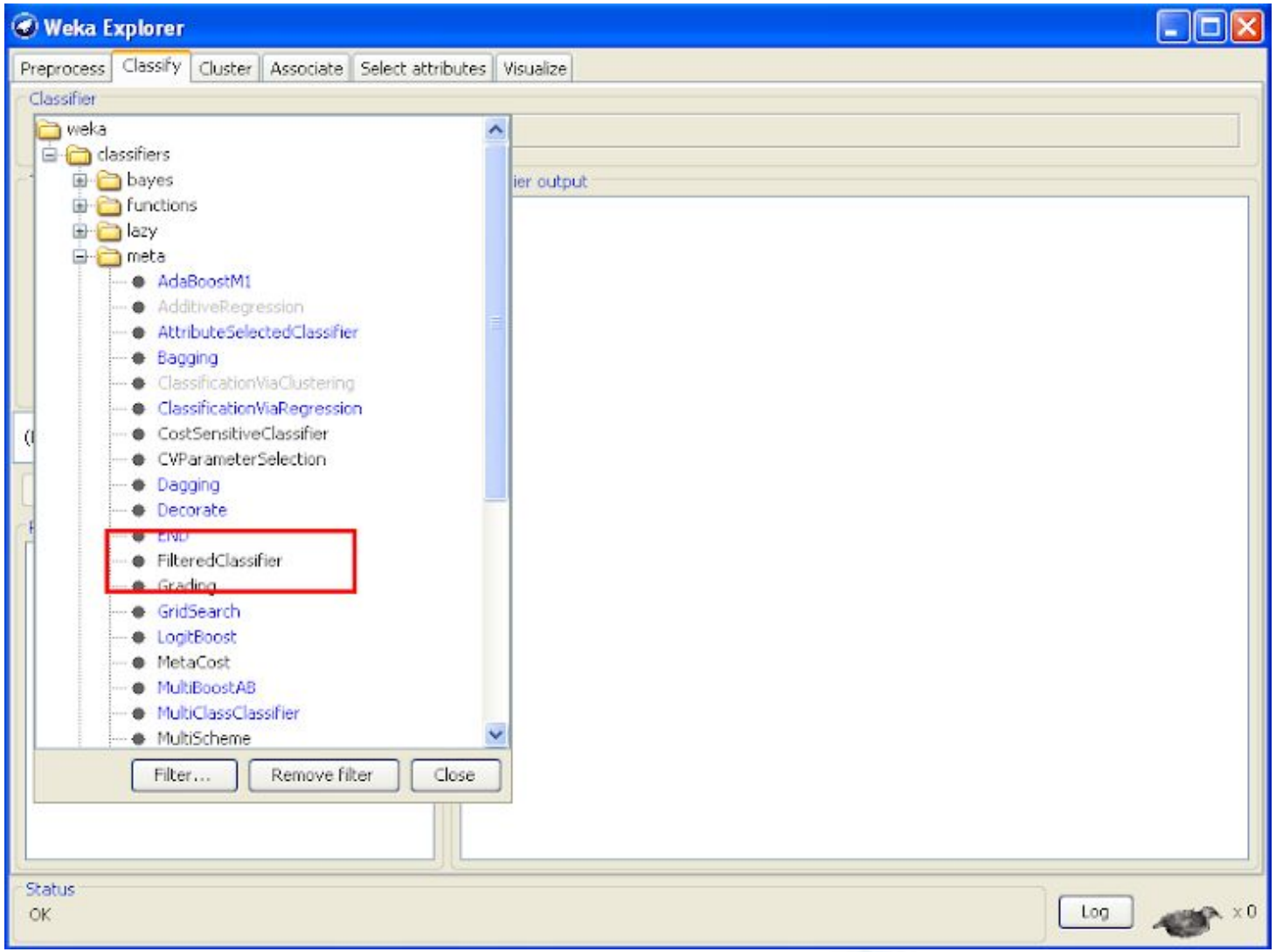
Il modo corretto di eseguire un esperimento di classificazione del testo con la cross-validation in WEKA è alimentare il processo di indicizzazione nel classificatore stesso, ovvero concatenare il filtro di indicizzazione (***StringToWordVector***) e il classificatore (***PART***).

In modo che ogni fold venga tokenizzate separatamente. Per ottenere questo risultato si utilizza il ***FilteredClassifier*** fornito da WEKA.

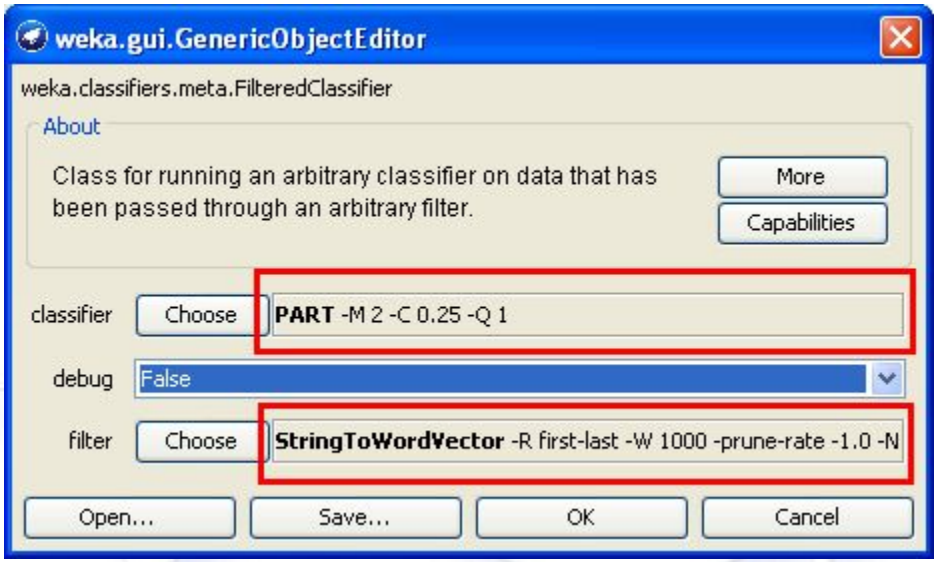
Torniamo alla raccolta di test originale nella scheda ***Preprocess***, che presenta due attributi: il messaggio (come stringa) e la classe.

Quindi puoi andare alla scheda ***Classify*** e scegliere il classificatore ***FilteredClassifier***, che è disponibile nella "weka> classifiers> meta".

WEKA: Text Mining



WEKA: Text Mining



WEKA: Text Mining

Se ora eseguiamo il nostro esperimento con la cross-validation utilizzando sempre 3 folds e il FilteredClassifier appena configurato, otteniamo risultati diversi.

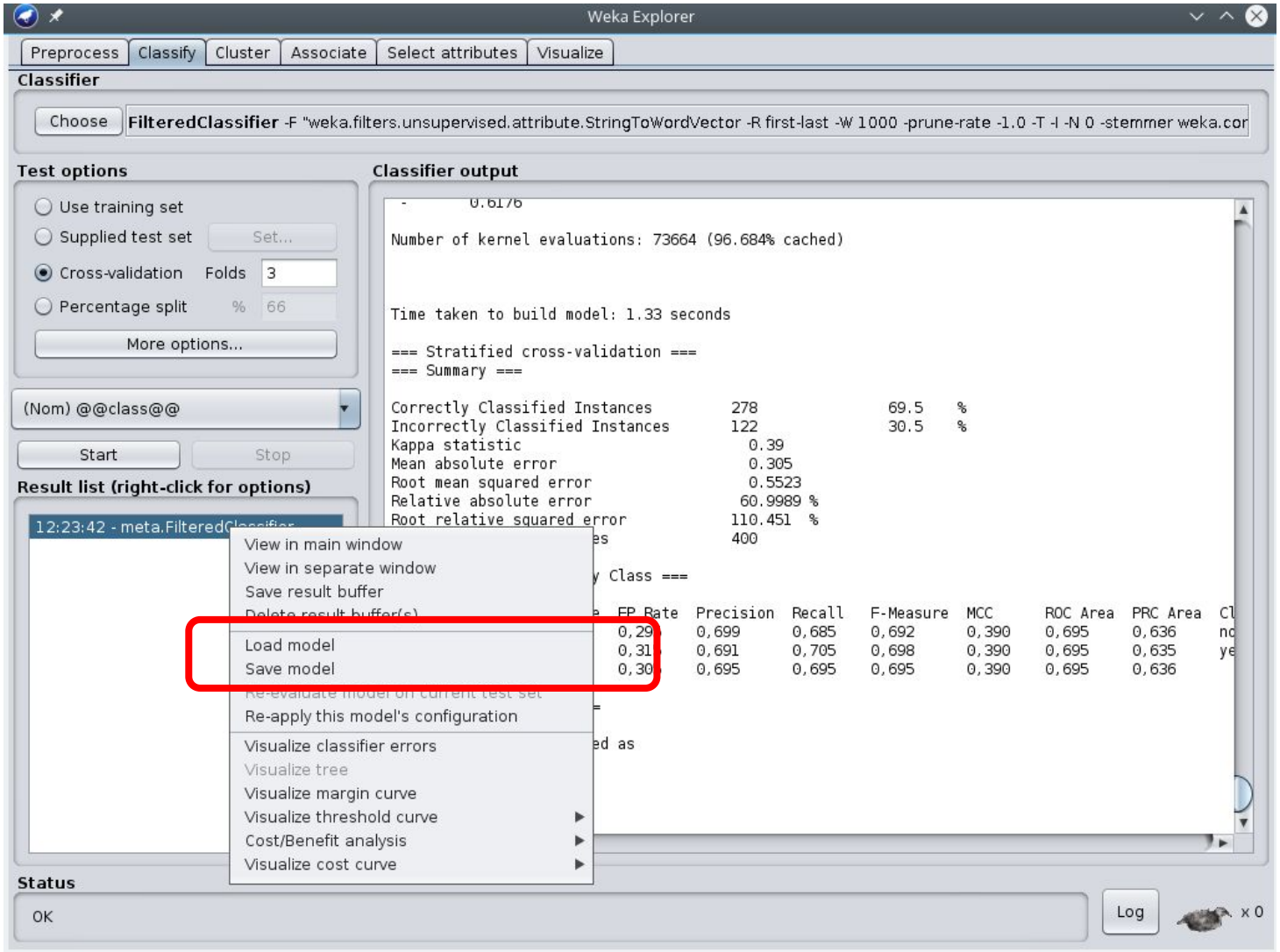
Per una precisione dell'0,929%, un po 'meglio di quella ottenuta con l'impostazione sbagliata.

Tuttavia, rileviamo 4 messaggi di spam in meno e il rapporto True Positive scende da 0,515 a 0,394. Questa configurazione è più realistica e riproduce meglio ciò che accadrà nel mondo reale, in cui troveremo eventi molto rilevanti ma invisibili e le statistiche potrebbero cambiare drasticamente nel tempo.

Quindi ora possiamo eseguire il nostro esperimento in modo sicuro, poiché nella classificazione non verranno utilizzati eventi non visibili.

Inoltre, se applichiamo qualsiasi tipo di filtro basato sulla teoria dell'informazione come per es. classifica gli attributi in base al loro valore dell’information gain, le statistiche saranno corrette, in quanto saranno basate sul training set per ogni esecuzione della cross-validation.

WEKA: Save/Load model



WEKA: Feature Selection

I dati grezzi utilizzati nell'apprendimento automatico contengono una combinazione di attributi, alcuni dei quali sono rilevanti per fare previsioni.

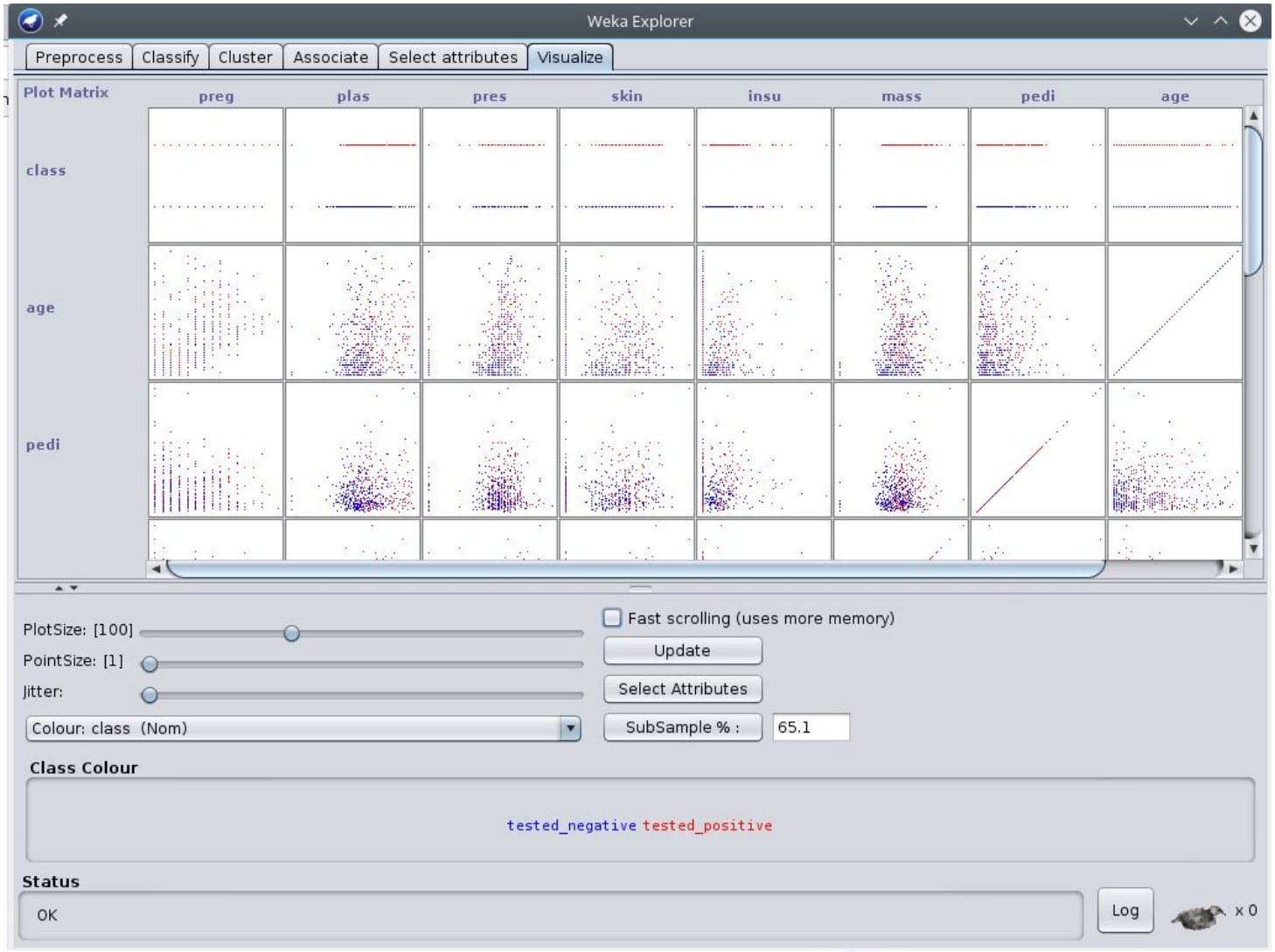
Come si fa a determinare quali utilizzare e quale rimuovere?

Il processo di selezione delle funzionalità nei dati per modellare il problema è chiamato ***feature selection***.

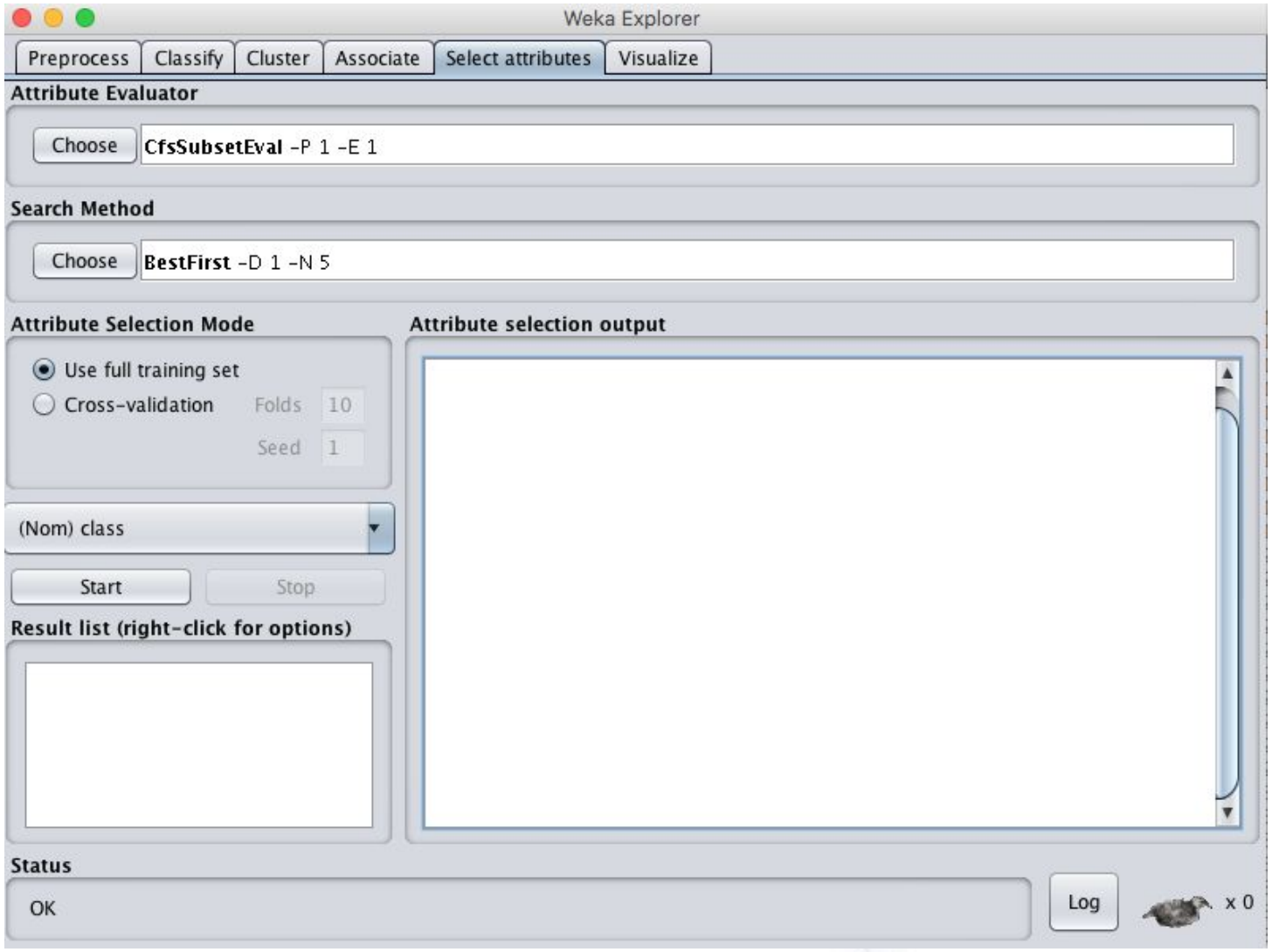
Weka supporta la feature selection con una scheda dedicata.

1. Aprire Weka GUI Chooser.
2. Click su "Explorer" per lanciare Explorer.
3. Nella scheda Preprocess caricare il dataset di "diabets" che si trova nella cartella ***data*** dell'installazione locale di Weka.
4. Click sulla scheda "Visualize"
5. Click sulla scheda "Select attributes".

WEKA: Feature Selection



WEKA: Feature Selection



WEKA: Feature Selection

La “feature selection” è divisa in due parti:

- ➔ Attribute Evaluator
- ➔ Metodo di ricerca

l’**Attribute Evaluator** è la tecnica con cui ogni attributo nel set di dati viene valutato nel contesto della variabile di output (ad esempio la classe).

Il **metodo di ricerca** è la tecnica con cui provare diverse combinazioni di attributi nel set di dati per arrivare a un elenco finale di features.

Alcune tecniche di Attribute Evaluator richiedono l'uso di metodi di ricerca specifici. Ad esempio, la tecnica ***CorrelationAttributeEval*** può essere utilizzata solo con un metodo di ricerca del ***Ranker***, che valuta ogni attributo ed elenca i risultati in ordine di classificazione.
Quando si selezionano diversi Attribute Evaluator, l'interfaccia potrebbe chiedere di cambiare il metodo di ricerca in uno compatibile con la tecnica scelta.

WEKA: Feature Selection

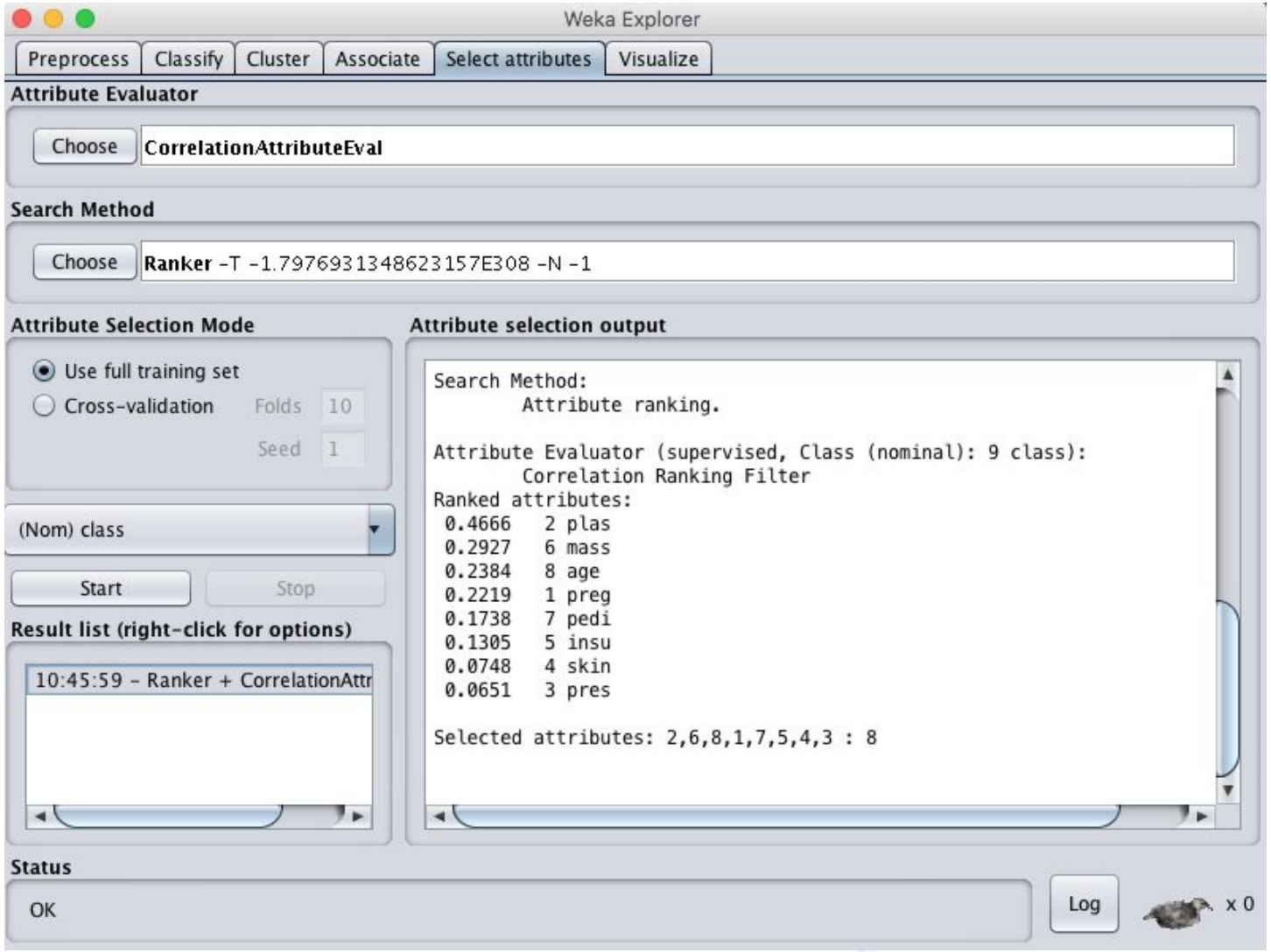
Una tecnica popolare per selezionare gli attributi più rilevanti nel set di dati è sfruttare la correlazione.

La correlazione viene indicata più formalmente come il coefficiente di correlazione di Pearson nelle statistiche.

È possibile calcolare la correlazione tra ciascun attributo e la variabile di uscita e selezionare solo quegli attributi che hanno una correlazione positiva o negativa da moderata ad alta (vicino a -1 o 1) ed eliminare quegli attributi con una bassa correlazione (valore vicino allo zero).

Weka supporta la selezione di funzionalità basate sulla correlazione con la tecnica ***CorrelationAttributeEval*** che richiede l'uso di un metodo di ricerca di Ranker.

WEKA: Feature Selection



WEKA: Feature Selection

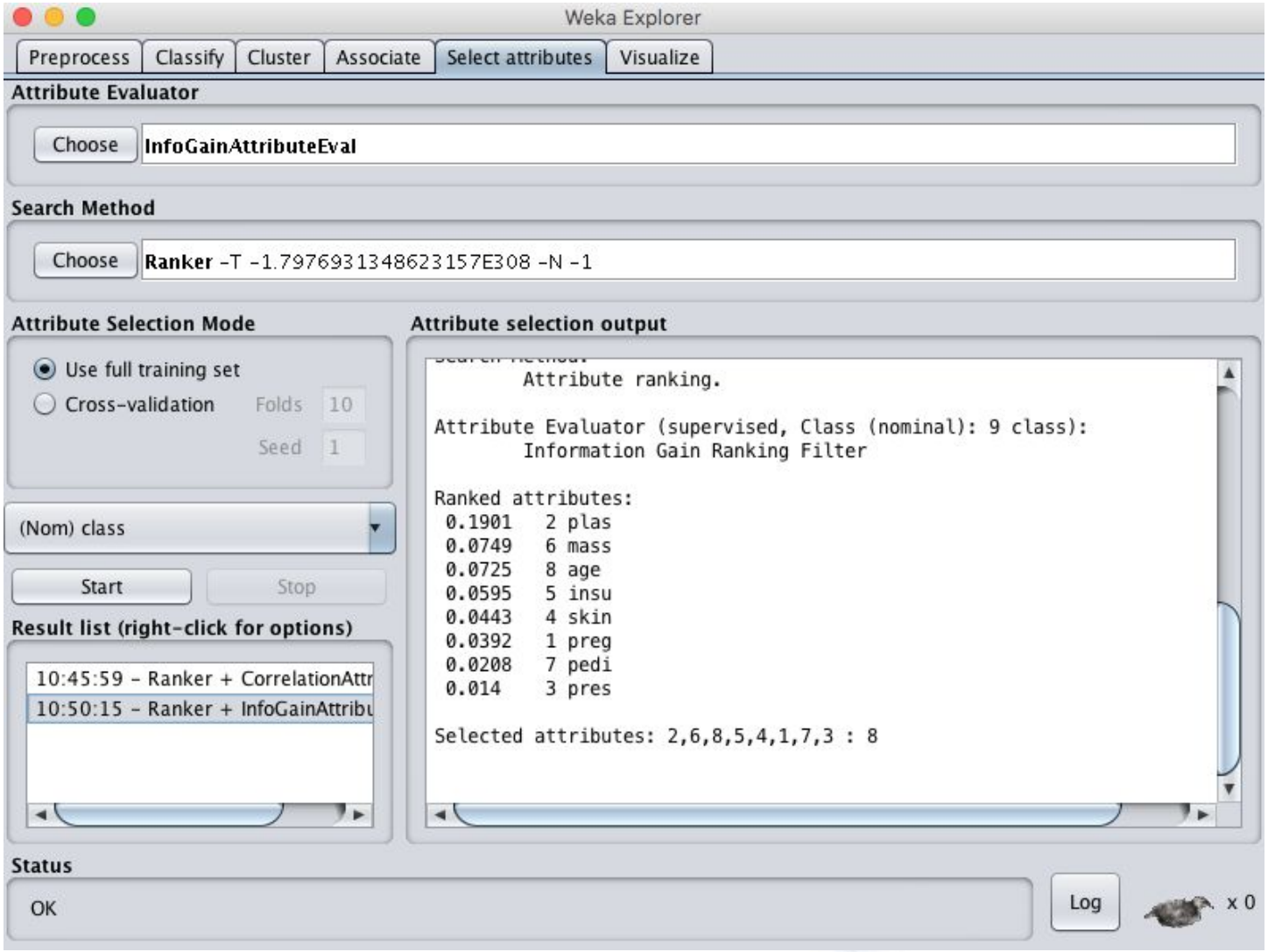
Un'altra tecnica popolare per la selezione delle caratteristiche è il calcolo dell'***information gain***.

È possibile calcolare information gain (chiamato anche entropia) per ogni attributo per la variabile di uscita. I valori variano da 0 (nessuna informazione) a 1 (massima informazione). Gli attributi che forniscono maggiori informazioni avranno un valore di guadagno di informazioni più elevato e possono essere selezionati, mentre quelli che non aggiungono molte informazioni avranno un punteggio inferiore e possono essere rimossi.

Weka supporta la feature selection tramite l'information gain utilizzando l'Attribute Evaluator di attributo ***InfoGainAttributeEval***. Come la tecnica di correlazione sopra, deve essere usato il metodo di ricerca dei Ranker.

Eseguendo questa tecnica dataset del diabete possiamo vedere che un attributo fornisce più informazioni di tutti gli altri (plas). Se usiamo un taglio arbitrario di 0.05, dovremmo anche selezionare gli attributi massa, età e insu e lasciare il resto dal nostro set di dati.

WEKA: Feature Selection



WEKA: Feature Selection

Una tecnica di feature selection molto popolare è quella di utilizzare un algoritmo di apprendimento generico per valutare le performance del classificatore con diversi sottoinsiemi di features.

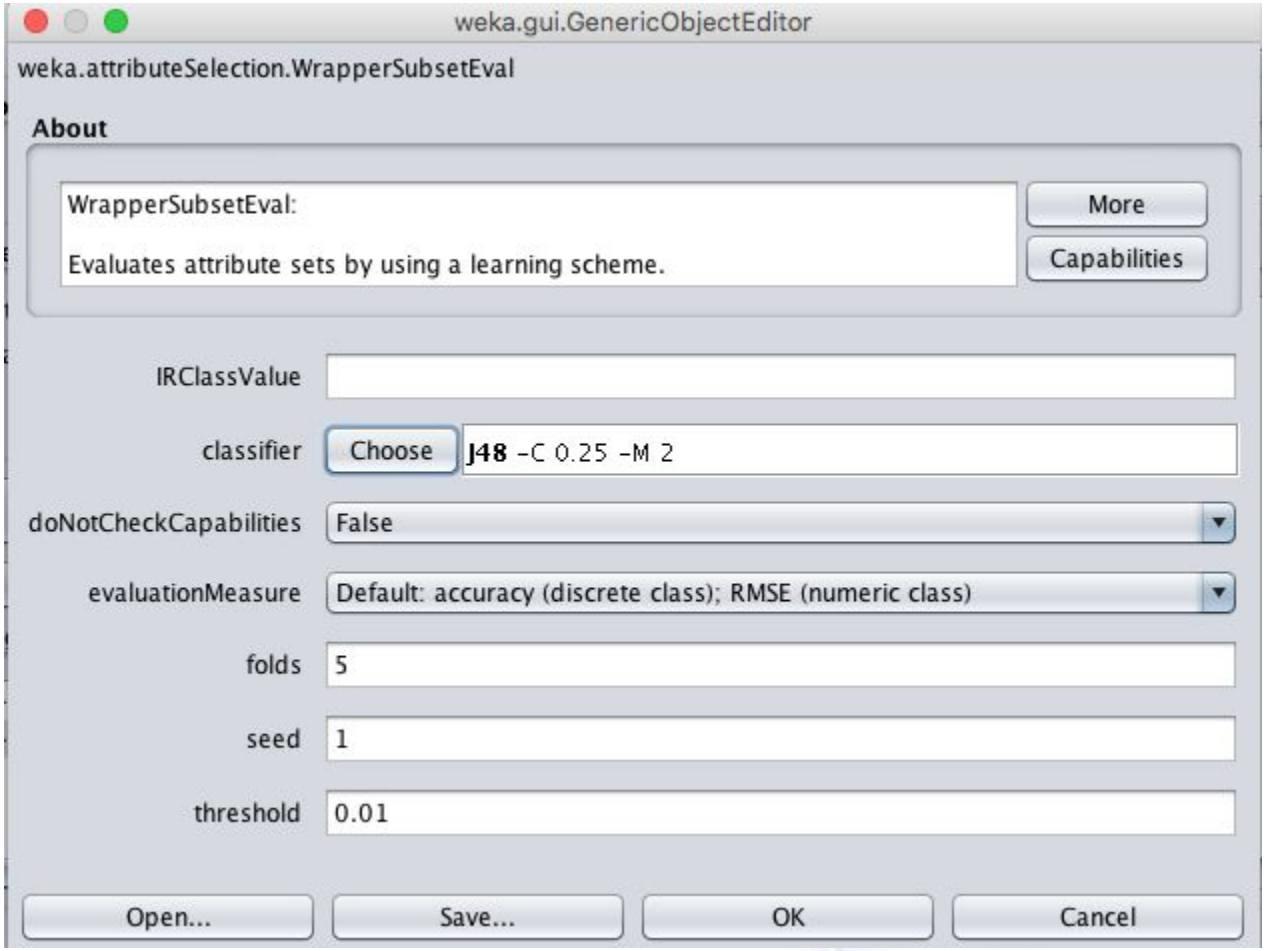
Il sottoinsieme che ha come risultato la migliore prestazione viene preso come sottoinsieme selezionato. L'algoritmo da utilizzare non deve essere per forza quello finale ma deve essere semplice e veloce come ad esempio un ***Decision Tree***.

In Weka questo tipo di selezione è supportato dalla tecnica ***WrapperSubsetEval*** e deve usare un metodo di ricerca ***GreedyStepwise*** o ***BestFirst***.
Quest'ultimo, ***BestFirst***, è preferibile perchè è possibile risparmiare il tempo di elaborazione.

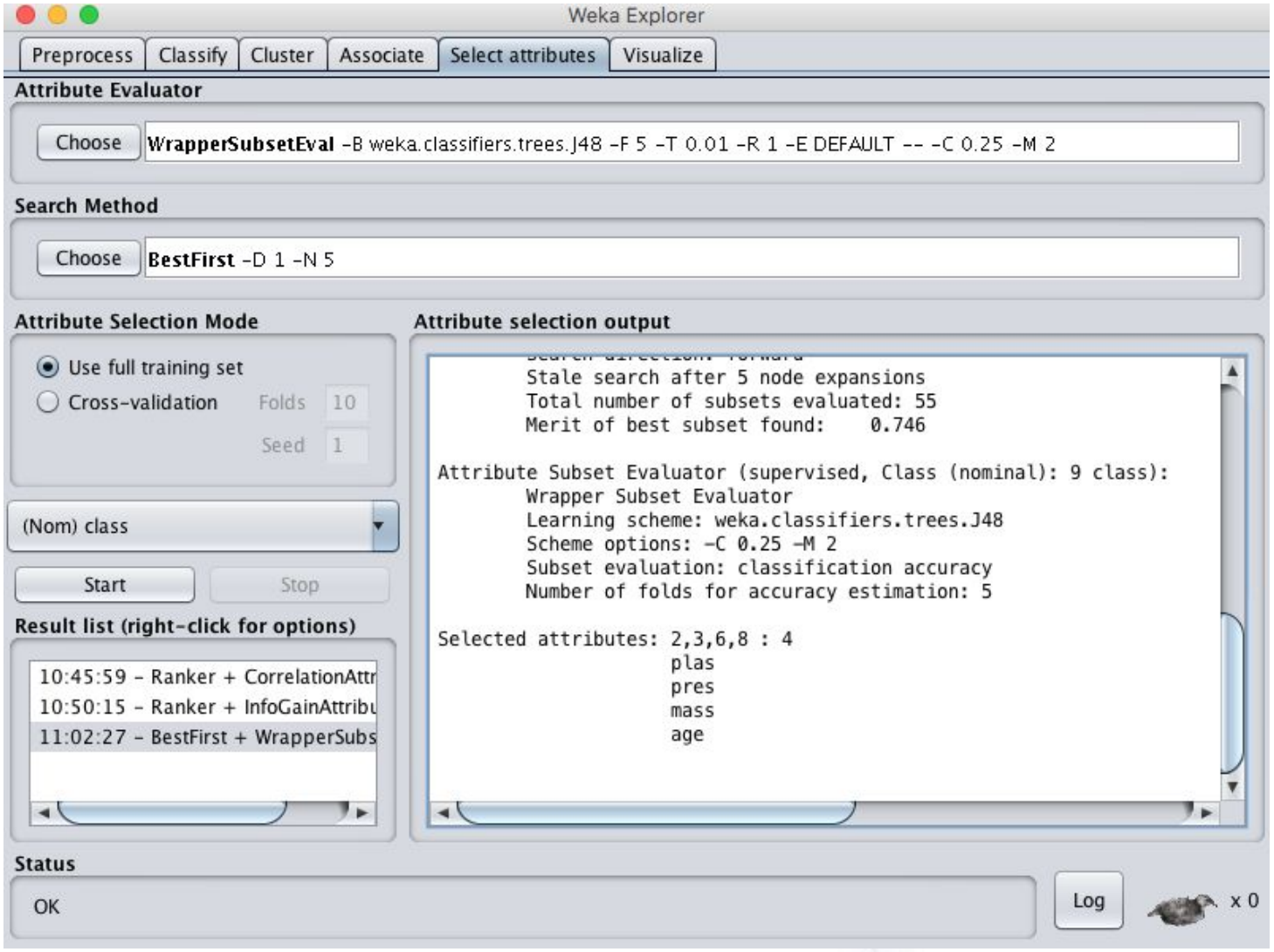
WEKA: Feature Selection

1. Selezionare “WrapperSubsetEval” come Attribute Evaluator.
2. Click sul “WrapperSubsetEval” per aprire le opzioni di configurazione del metodo.
3. Click “Choose” per selezionare il classifier e scegliere ***trees/J48***.
4. Click “OK” per confermare la selezione.
5. Cambiare il ***Search Method*** in ***BestFirst***.
6. Click “Start” per iniziare la valutazione.

WEKA: Feature Selection



WEKA: Feature Selection



WEKA: Feature Selection

Controllando i risultati delle tre tecniche, possiamo vedere alcune sovrapposizioni nelle features selezionate (ad esempio plas), ma anche differenze.

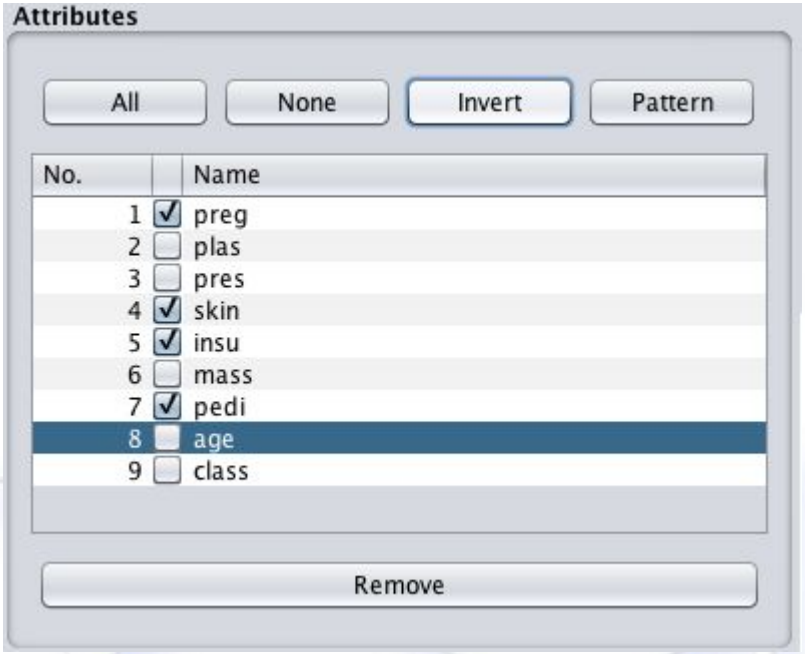
È una buona idea valutare un numero di sottoinsiemi differenti del set di features. Una vista del set di dati non è altro che un sottoinsieme di feature selezionate da una determinata tecnica di feature selection.

Ad esempio, prendendo i risultati dall'ultima tecnica di selezione delle funzionalità, supponiamo di voler creare una vista del set di dati con solo i seguenti attributi: plas, pres, mass e age:

1. Fare clic sulla scheda "Preprocess".

2. Nella selezione "Attributi" selezionare tutti gli attributi di plas, pres, mass, age e class.

WEKA: Feature Selection



WEKA: Feature Selection

Quale Tecnica utilizzare?

Non si può sapere quali sottoinsiemi di features produrranno i modelli più accurati.

Pertanto, è una buona pratica provare diverse tecniche di selezione delle features sui dati e creare a loro volta molte visualizzazioni diverse dei dati.

Seleziona una buona tecnica generica, come un albero decisionale, e crea un modello per ciascuna vista dei tuoi dati.

Confronta i risultati per avere un'idea di quale vista dei tuoi dati porta alla migliore performance. Questo darà un'idea della features che meglio si adattano alla struttura del problema di apprendimento.

WEKA: Language Detection

1. Dal repository ***https://github.com/marcoortu/WAAT-2019*** fare il checkout del branch ***weka.***

2. Utilizzare il file ***data/language.arff*** con almeno 3 algoritmi di classificazione e identificare quello con le performance migliori per l'identificazione della lingua di un testo.

WEKA: Sentiment Analysis

1. Dal repository ***https://github.com/marcoortu/WAAT-2020*** fare il checkout del branch ***weka.***

2. Utilizzare il file ***data/sentiment.arff*** per creare un classificatore in grado di predire il sentiment di una review con WEKA.
   a. Importare il file su Weka utilizzando l'explorer e analizzare la struttura
   b. Creare un ***FilteredClassifier*** composto da:
      i. ***StringToWordVector*** utilizzando le trasformazioni TF e IDF
      ii. ***SMO*** (Sequential Minimum Optimization, implementazione del ***SVM***) classifier
   c. Creare un esperimento con la cross validation utilizzando 10 folds e valutare le performance del classificatore.

Referenze

http://transact.dl.sourceforge.net/sourceforge/weka/WekaManual-3.6.0.pdf

http://www.esp.uem.es/~jmgomez/tmweka/index.html