Roma
20-23.03.2013
www.codemotionworld.com

Simone Scardapane

How to make smarter programs. A gentle introduction to Machine Learning

simone.scardapane@uniroma1.it

Argomenti di oggi

Simone Scardapane *simone.scardapane@uniroma1.it*

1. Il Machine Learning (ieri ed oggi)
2. Un esempio pratico: *spam detection*
3. Cenni su Weka

Cos'è il Machine Learning?

Simone Scardapane
simone.scardapane@uniroma1.it

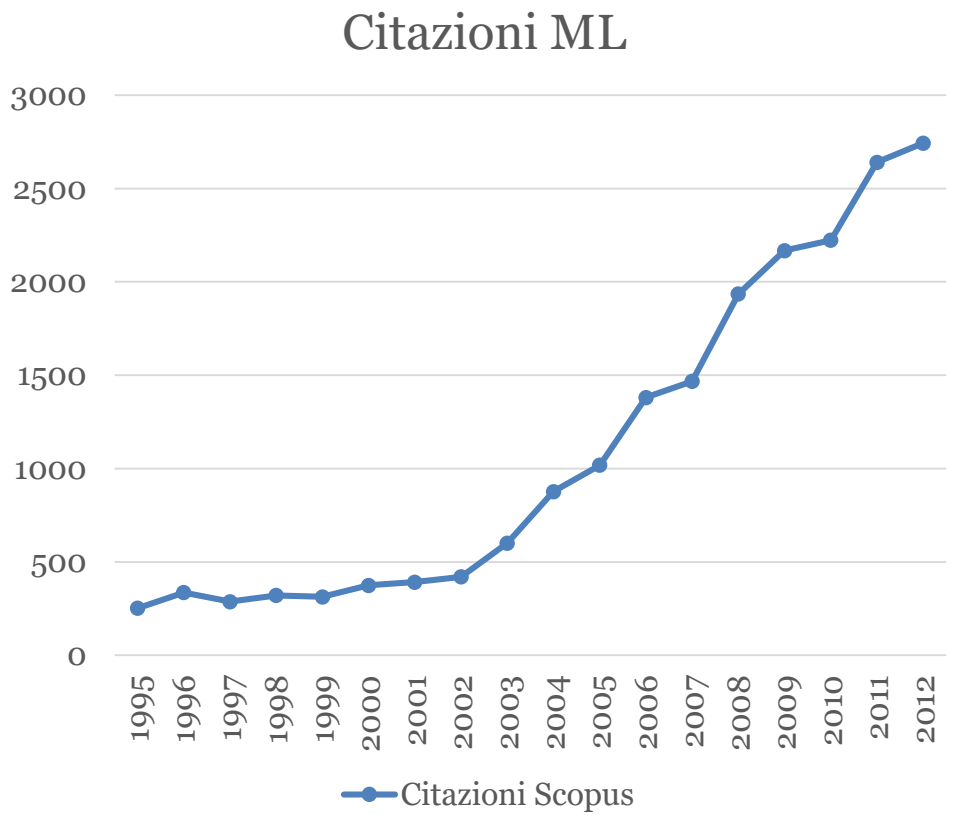
«Estrazione *automatica* di conoscenza a partire da un insieme di dati»

Chi ci lavora?

Simone Scardapane
simone.scardapane@uniroma1.it

65 anni di ricerche da parte di:

- Ingegneri
- Informatici
- Statistici
- Matematici
- Fisici
- Neuroscienziati...

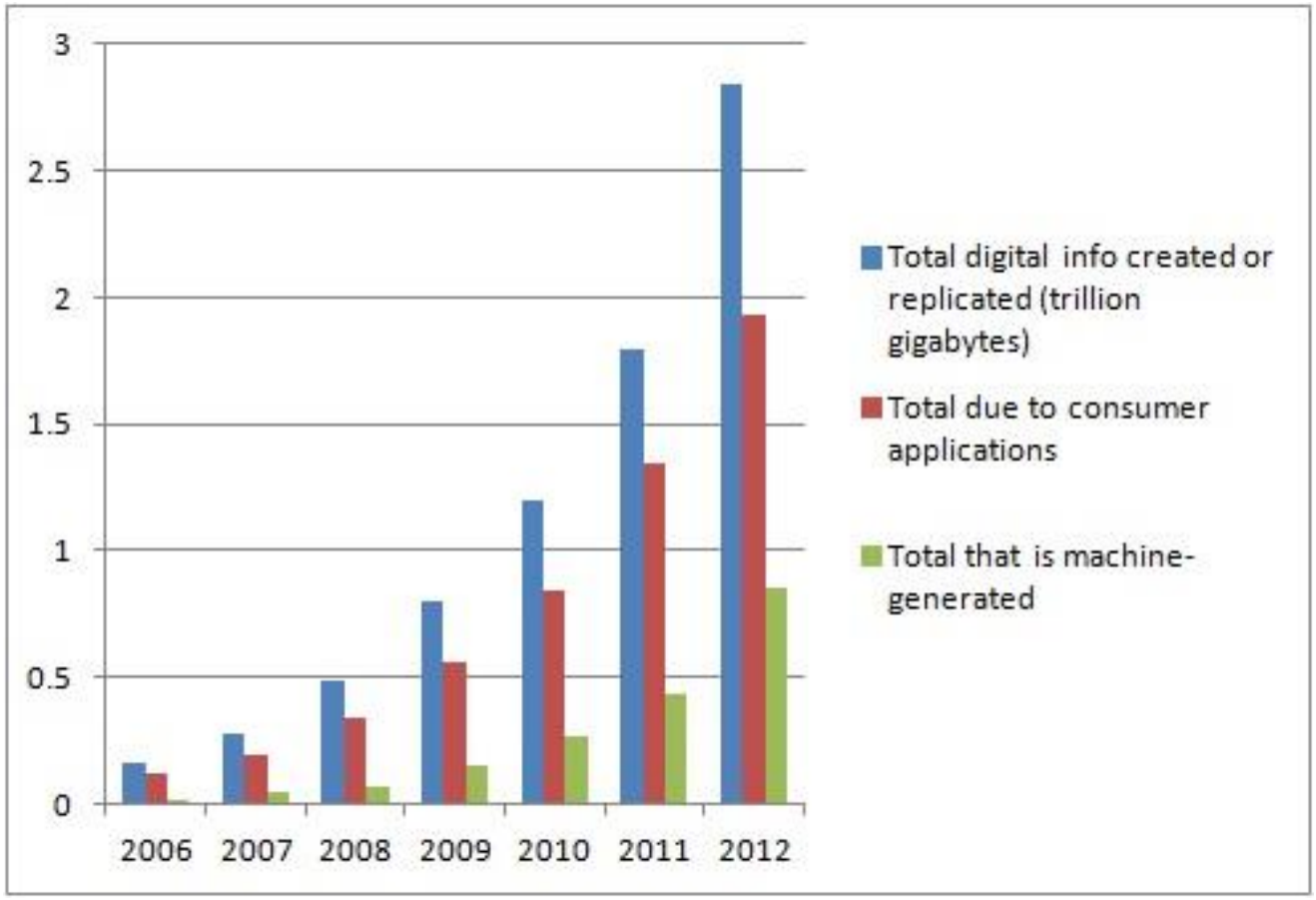


Il Machine Learning oggi

Simone Scardapane
simone.scardapane@uniroma1.it

Perché usare il Machine Learning?

Simone Scardapane
simone.scardapane@uniroma1.it



(Fonte: IDC)

Ma voi cosa potete farci?

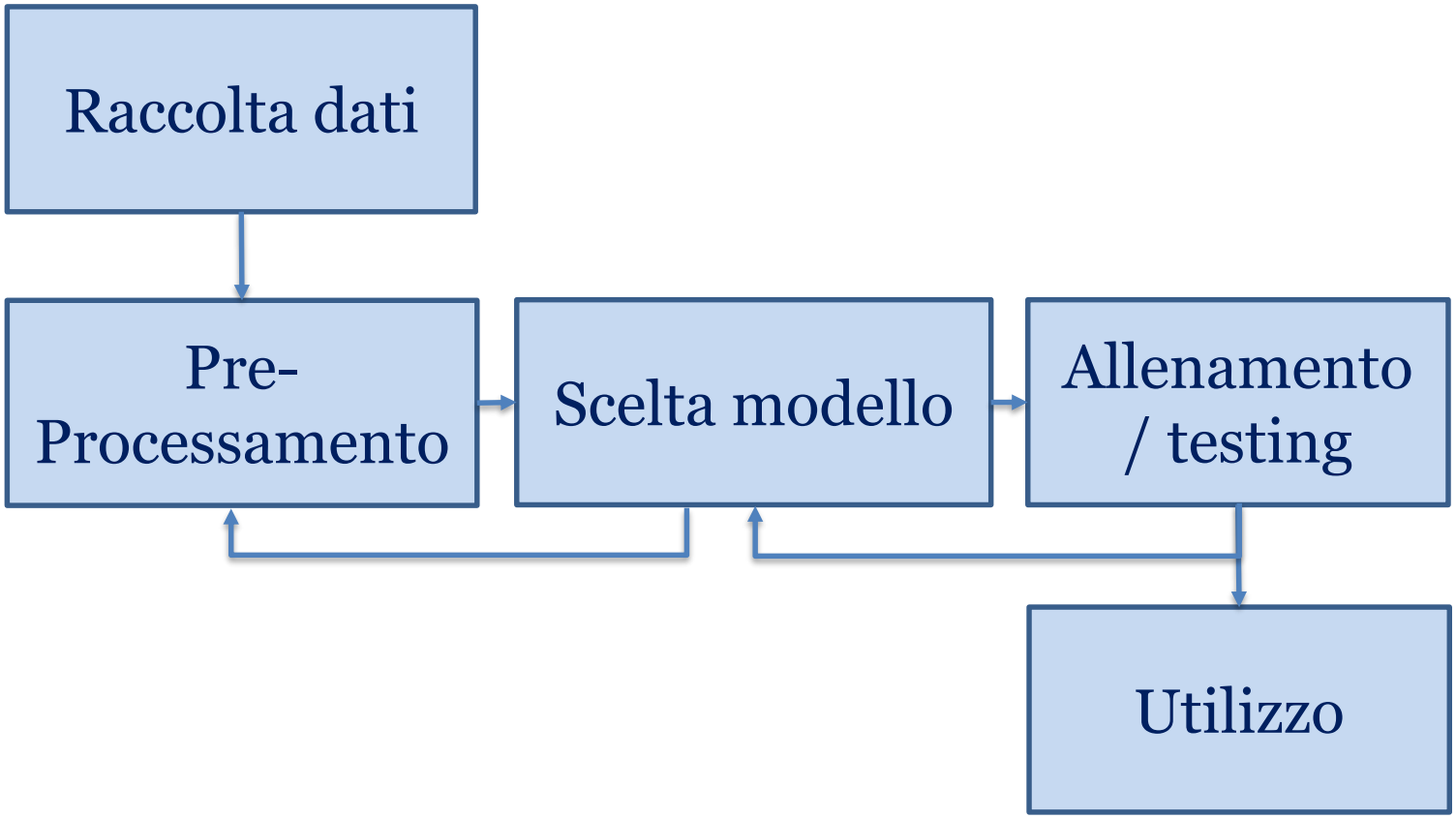
Simone Scardapane
simone.scardapane@uniroma1.it

Possibili operazioni su una libreria musicale:

1. Classificazione del genere (o del *mood*)
2. Raggruppamento automatico (clustering)
3. Tagging
4. Ricerca per similitudini (association rule)
5. Predizione del prossimo ascolto
6. ...

Il processo di Learning

Simone Scardapane
simone.scardapane@uniroma1.it



Variazioni

Simone Scardapane
simone.scardapane@uniroma1.it

- Online Learning: l'algoritmo riceve dati in real-time e si adatta di conseguenza.
- Active Learning: durante la fase di learning, è possibile richiedere attivamente nuove informazioni.
- Collaborative/Cooperative Learning...

Un esempio pratico

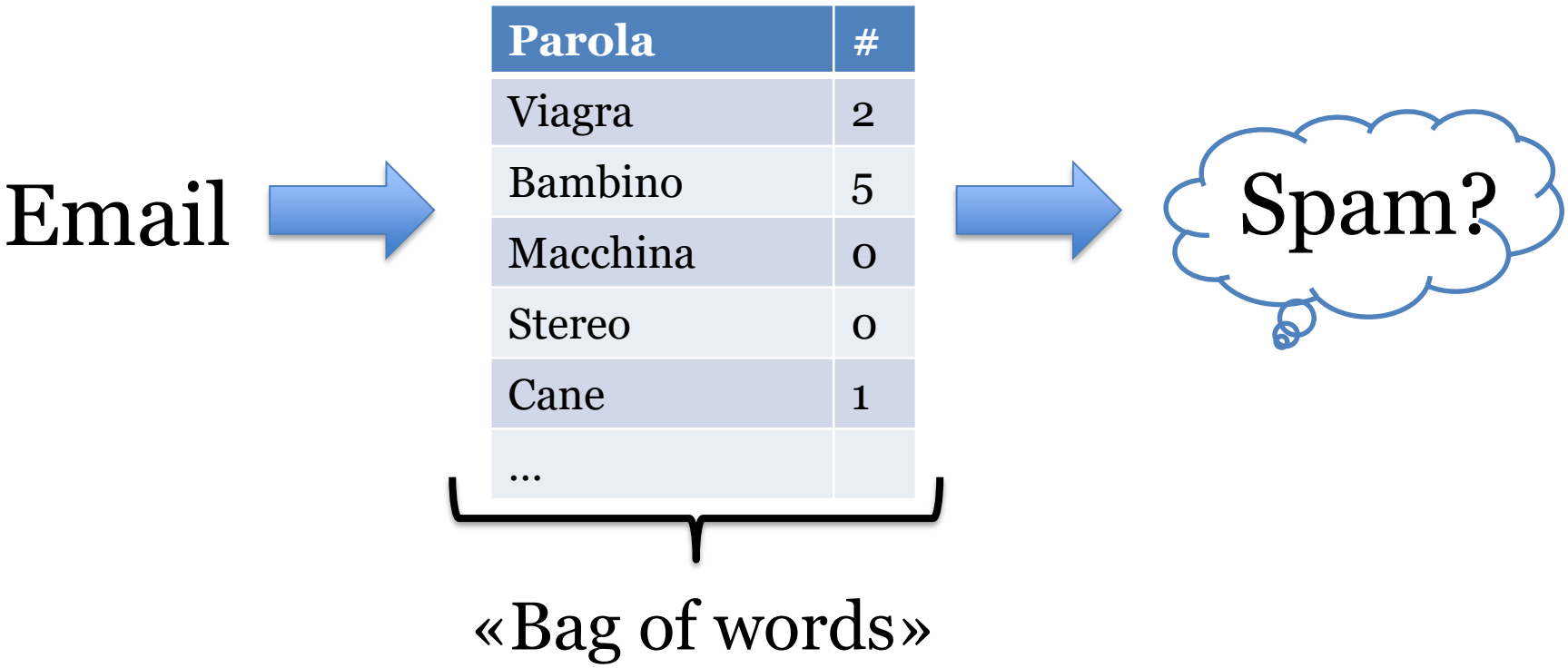
Simone Scardapane
simone.scardapane@uniroma1.it

- Dati: insieme *S* di emails taggate come spam / non spam.
- Obiettivo: metodo automatico per individuare spam.

- Problemi:
  1. Come rappresentare l'email?
  2. Che modello utilizzare?
  3. Come allenarlo?

Passo 1: Pre-processamento

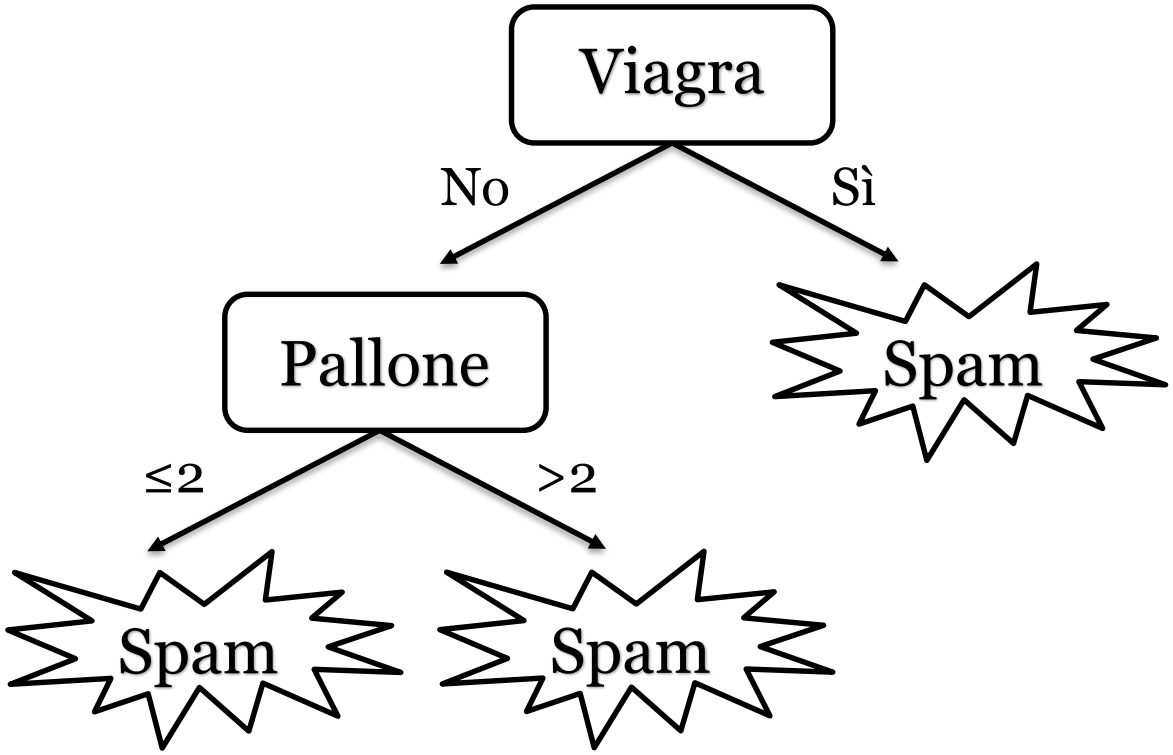
Simone Scardapane
simone.scardapane@uniroma1.it



Passo 2: La scelta del modello

Simone Scardapane
simone.scardapane@uniroma1.it

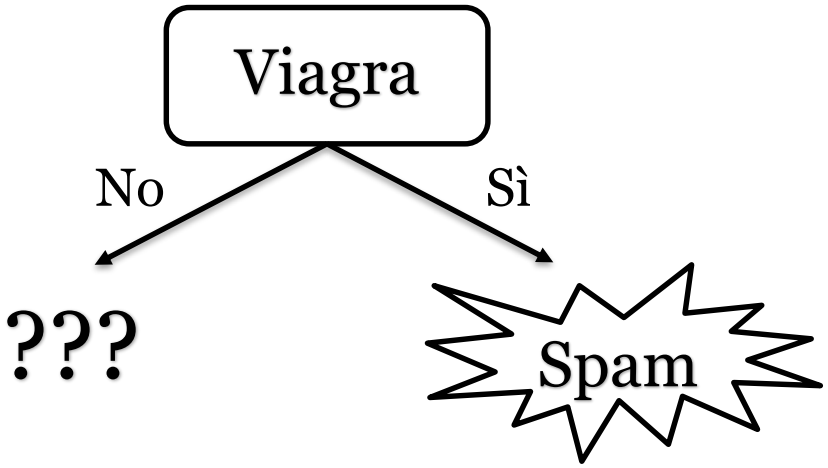
Decision Tree:



Passo 3: Allenamento

Simone Scardapane
simone.scardapane@uniroma1.it

Come costruirlo?



Allenamento /2

Simone Scardapane
simone.scardapane@uniroma1.it

Consideriamo l'algoritmo C4.5:

1. Scegliamo per il nodo l'attributo *a* che «divide» meglio i dati.
2. Suddividiamo l'insieme lungo i nodi.
3. Ci fermiamo quando i dati sono perfettamente divisi.

(Difficoltà: gestire dati continui, mancanti...)

Allenamento /3

Simone Scardapane
simone.scardapane@uniroma1.it

Matematicamente, scegliamo l'attributo che riduce maggiormente l'*entropia*:

$$E(S) = -p_{spam} \log p_{spam} - p_{\neg spam} \log p_{\neg spam}$$

(In realtà ci basiamo su una misura correlata)

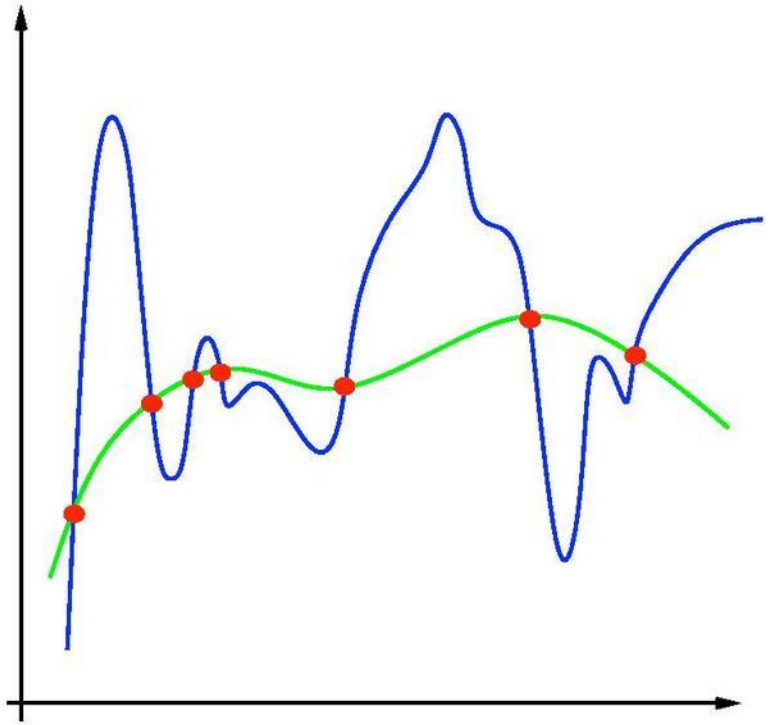
Il tema di *massimizzare* un criterio è cruciale nel ML.

Overfitting

Simone Scardapane
simone.scardapane@uniroma1.it

Problema principale: *overfitting!*

(Immagine con Copyright Tomaso Poggio)



Pruning

Simone Scardapane
simone.scardapane@uniroma1.it

Possibile soluzione (per i decision trees):

- Si tiene da parte un insieme di dati.
- Si eliminano i rami non necessari (*pruning*) in base a quei dati *(error-based pruning*).

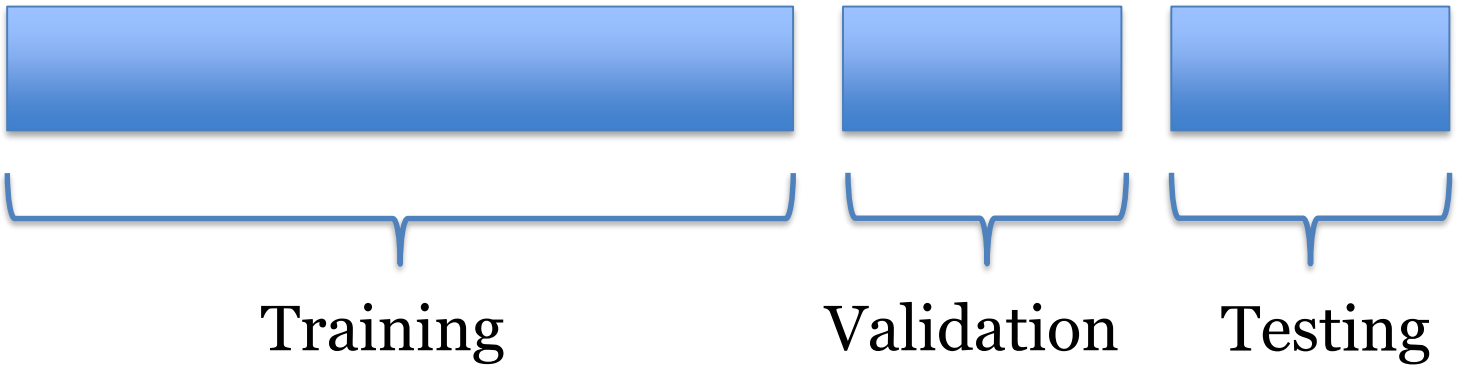
Più generalmente si usano tecniche di *cross-validation*.

Testare l'algoritmo

Simone Scardapane
simone.scardapane@uniroma1.it

Possiamo tenere da parte un secondo insieme per *testare* l'accuratezza dell'algoritmo.

Dividiamo quindi i nostri dati in tre parti:



Weka

Simone Scardapane
simone.scardapane@uniroma1.it

Tool di *data mining* sviluppato dalla Waikato University in Java:

1. Ampio numero di funzioni.
2. Estremamente portabile.
3. Interfaccia di facile utilizzo.

SpamBase

Simone Scardapane
simone.scardapane@uniroma1.it

Usiamo il dataset *SpamBase* dal repository UCI: http://archive.ics.uci.edu/ml/datasets/Spambase

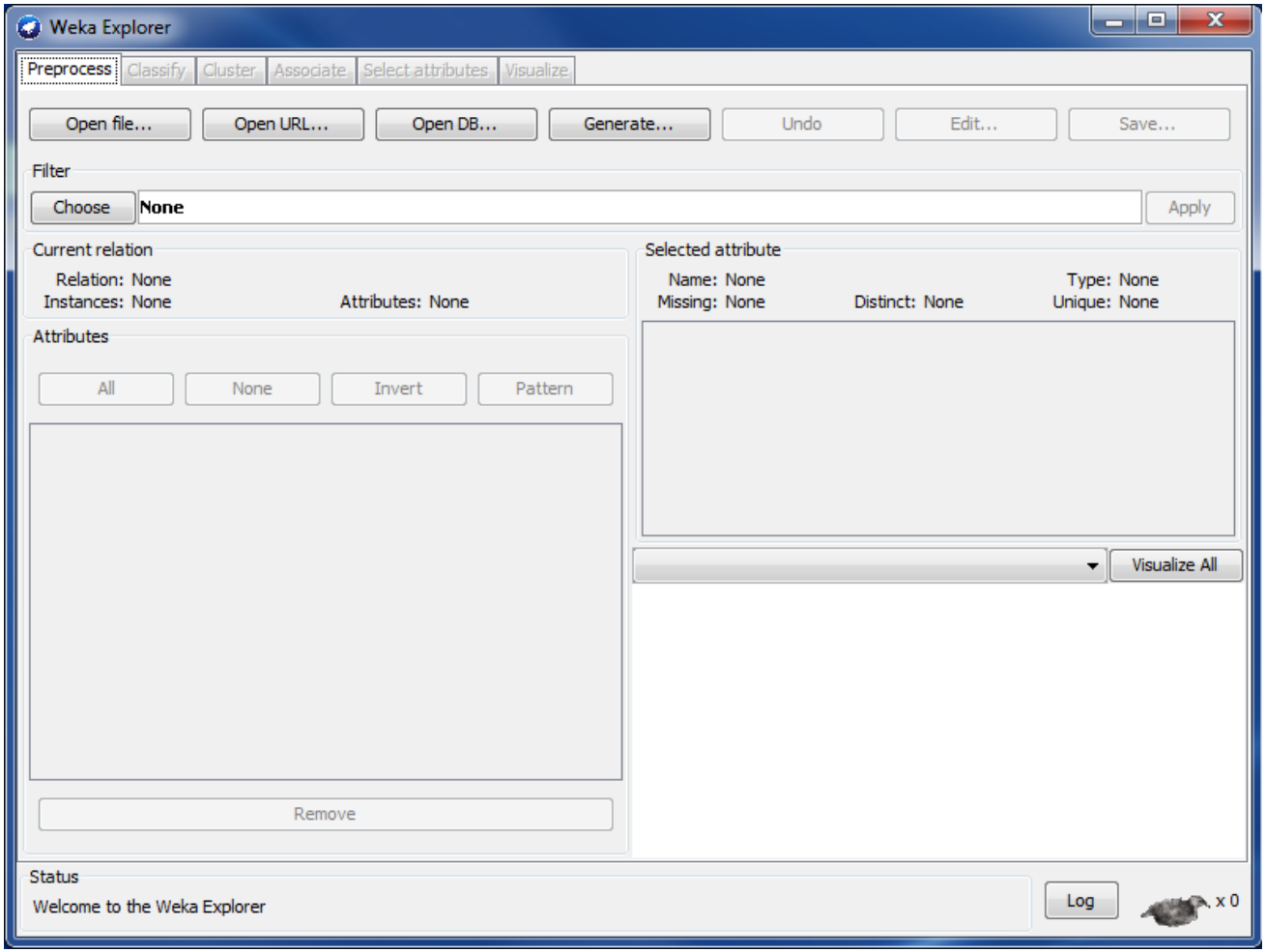
4601 email rappresentate da 48 frequenze di parole (più qualche informazione aggiuntiva).

I dati sono salvati in formato ARFF (file di testo):

1. Header con descrizione degli attributi.
2. Elenco delle email.

Interfaccia di Weka

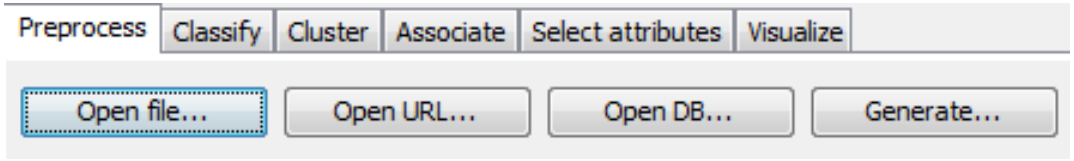
Simone Scardapane
simone.scardapane@uniroma1.it

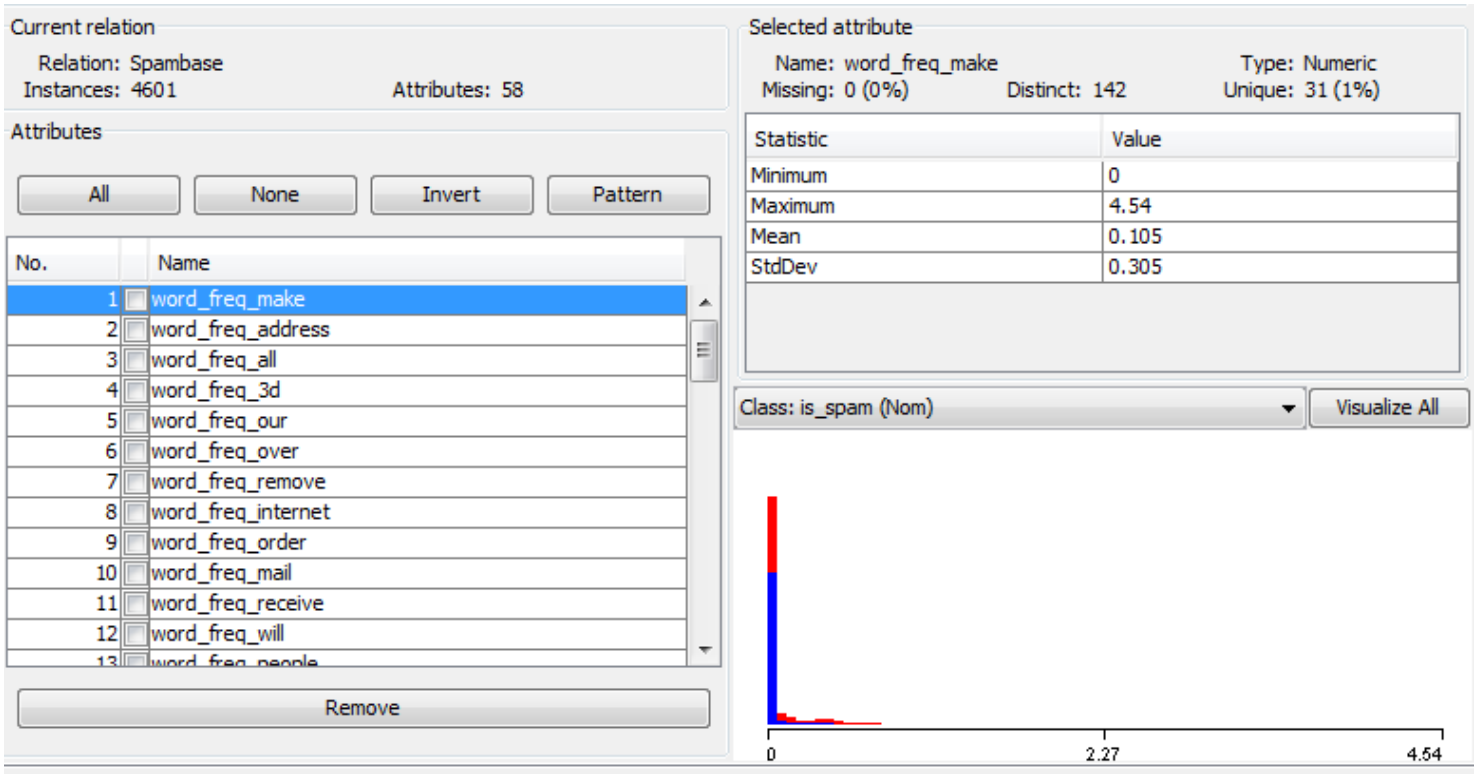


Apertura file

Simone Scardapane
simone.scardapane@uniroma1.it

Apriamo il file:

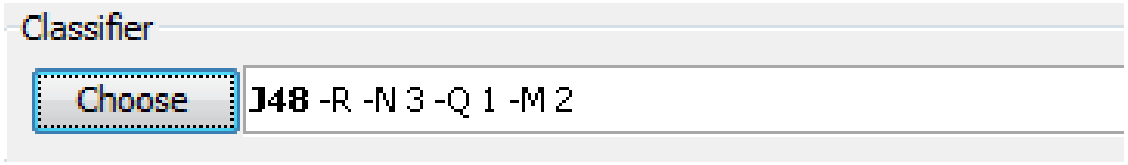


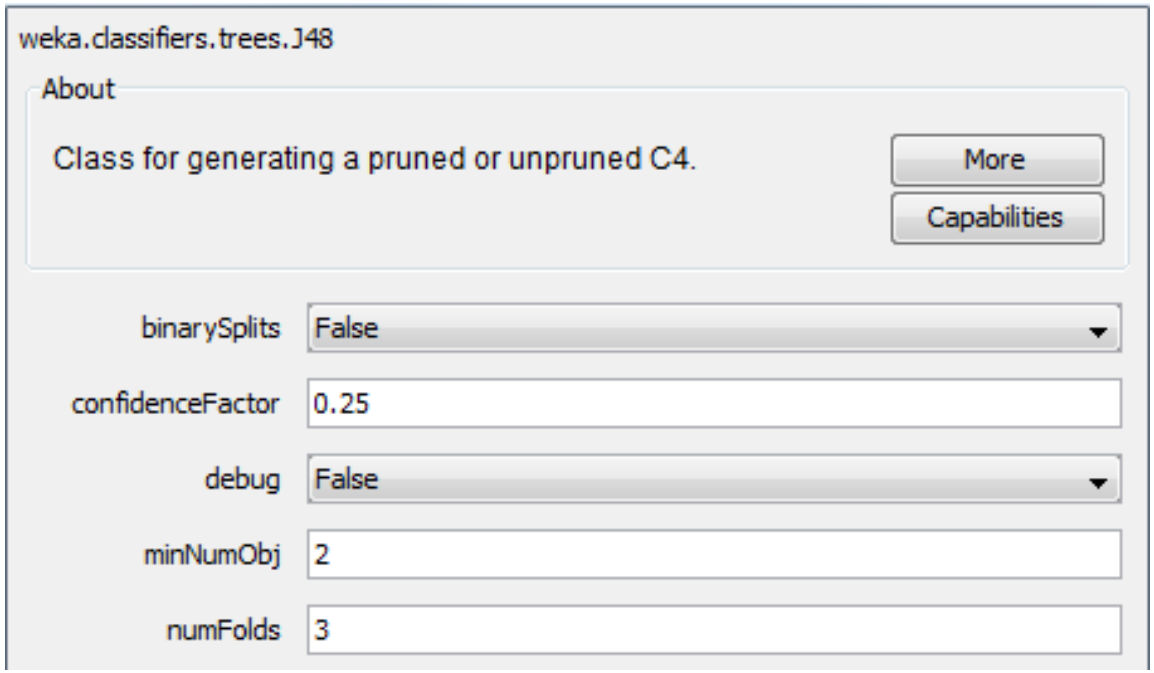


Scelta del classificatore

Simone Scardapane
simone.scardapane@uniroma1.it

Scegliamo il modello:





Simone Scardapane
simone.scardapane@uniroma1.it

Risultati:

```
Time taken to build model: 0.42 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances         836               90.8696 %
Incorrectly Classified Instances        84                9.1304 %
Kappa statistic                          0.8105
Mean absolute error                      0.1235
Root mean squared error                  0.2763
Relative absolute error                 25.7409 %
Root relative squared error             56.2152 %
Total Number of Instances              920
```

Albero finale

Simone Scardapane *simone.scardapane@uniroma1.it*

```
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   word_freq_meeting > 0.35: 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   word_freq_address > 0.12: 0 (3.0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   word_freq_receive > 1.15: 0 (8.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   word_freq_email > 0.19: 1 (3.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   word_freq_edu > 0.26: 0 (35.0/1.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   word_freq_85 > 0.04: 0 (6.0)
|   |   |   |   |   |   |   |   |   |   |   |   word_freq_hp > 0.1: 0 (34.0)
|   |   |   |   |   |   |   |   |   |   |   word_freq_over > 0.52
|   |   |   |   |   |   |   |   |   |   |   |   word_freq_edu <= 0.39: 1 (5.0)
|   |   |   |   |   |   |   |   |   |   |   |   word_freq_edu > 0.39: 0 (2.0)
|   |   |   |   |   |   |   |   |   |   word_freq_internet > 0.25: 0 (33.0/11.0)
|   |   |   |   |   |   |   |   |   word_freq_650 > 0.23
|   |   |   |   |   |   |   |   |   |   word_freq_parts <= 0.05
|   |   |   |   |   |   |   |   |   |   |   word_freq_report <= 1.61: 0 (38.0/9.0)
|   |   |   |   |   |   |   |   |   |   |   word_freq_report > 1.61: 1 (2.0)
|   |   |   |   |   |   |   |   |   |   word_freq_parts > 0.05: 1 (3.0)
|   |   |   |   |   |   |   |   word_freq_hpl > 0.39: 0 (222.0)
|   |   |   |   |   |   |   word_freq_george > 0: 0 (466.0)
|   |   |   |   |   |   char_freq_! > 0.391
|   |   |   |   |   |   |   word_freq_internet <= 0.33
|   |   |   |   |   |   |   |   capital_run_length_longest <= 20
|   |   |   |   |   |   |   |   |   word_freq_business <= 0.26
|   |   |   |   |   |   |   |   |   |   capital_run_length_average <= 3.277
|   |   |   |   |   |   |   |   |   |   |   char_freq_! <= 2.443: 0 (95.0/5.0)
|   |   |   |   |   |   |   |   |   |   |   char_freq_! > 2.443
|   |   |   |   |   |   |   |   |   |   |   |   word_freq_re <= 0.47
|   |   |   |   |   |   |   |   |   |   |   |   |   capital_run_length_average <= 1.394: 0 (6.0/2.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   capital_run_length_average > 1.394: 1 (4.0)
|   |   |   |   |   |   |   |   |   |   |   |   word_freq_re > 0.47: 0 (2.0)
|   |   |   |   |   |   |   |   |   |   capital_run_length_average > 3.277: 1 (7.0/1.0)
|   |   |   |   |   |   |   |   |   word_freq_business > 0.26: 1 (11.0/1.0)
|   |   |   |   |   |   |   |   capital_run_length_longest > 20
|   |   |   |   |   |   |   |   |   capital_run_length_total <= 64: 0 (2.0)
|   |   |   |   |   |   |   |   |   capital_run_length_total > 64: 1 (23.0/3.0)
|   |   |   |   |   |   |   word_freq_internet > 0.33: 1 (8.0)
```

API di Weka

Simone Scardapane
simone.scardapane@uniroma1.it

```java
import weka.core.converters.ConverterUtils.DataSource;
...
DataSource source = new DataSource("/some/where/data.arff");
Instances data = source.getDataSet();
// setting class attribute if the data format does not provide this information
// For example, the XRFF format saves the class attribute information as well
if (data.classIndex() == -1)
  data.setClassIndex(data.numAttributes() - 1);
```

```java
import weka.classifiers.trees.J48;
...
String[] options = new String[1];
options[0] = "-U";            // unpruned tree
J48 tree = new J48();         // new instance of tree
tree.setOptions(options);     // set the options
tree.buildClassifier(data);   // build classifier
```

http://weka.wikispaces.com/Use+WEKA+in+your+Java+code

Letture consigliate

Simone Scardapane *simone.scardapane@uniroma1.it*

Programming Collective Intelligence, di **Toby Segaran**. Publisher: O'Reilly Media (2007).

Data Mining: Practical Machine Learning Tools and Techniques, di **Witten, Frank et Hall**. Publisher: Morgan Kaufmann (2011).

Introduction to Machine Learning, di **Alpaydin**. Publisher: the MIT Press (2009).

Fine!

Simone Scardapane
simone.scardapane@uniroma1.it

Grazie per l'attenzione! ☺