HAP/LAP

Introduction to Machine Learning

Revision of ML process and some preprocessing in Weka

Olatz Arbelaitz: olatz.arbelaitz@ehu.eus
www.aldapa.eus

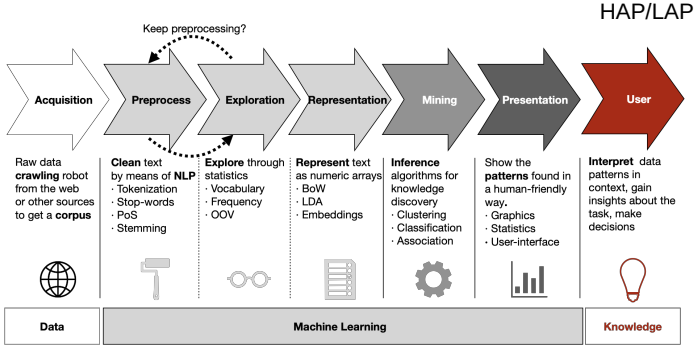
HAP/LAP

Topics

1.- **Introduction. Machine Learning for LNP**
2.- Learning with WEKA software:
- 2.1.-Introduction
- 2.2.-Preprocessing
  - Attribute (feature) selection
- 2.3.-Evaluation
- 2.4.- Basic ML algorithms: Naive Bayes, K-NN, Decision Trees, Rules, …

HAP/LAP

1-Introduction



HAP/LAP

Introduction

NLP Evolution

1960-
Rules:
Main companies use it but it is very expensive

1990 -
Machine Learning tasks with specific features.
Supervised ML.
Need to label documents by humans

2010 -
Machine Learning tasks with no task specific features.
Deep LEarning

Introduction

HAP/LAP

Why it is so useful nowadays:

- a lot of data available
- for 2011, 600 Exabytes = 600 x 10^{12} MB
- Only % 0,007 is in paper
- H igh computation capacity

Problems:

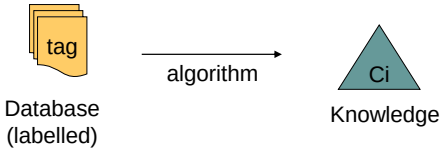
- data are not always clean
- complex algorithms, difficult to understand
- a lot and different features to learn → **not enough space in memory**

Introduction

HAP/LAP

What is machine learning?

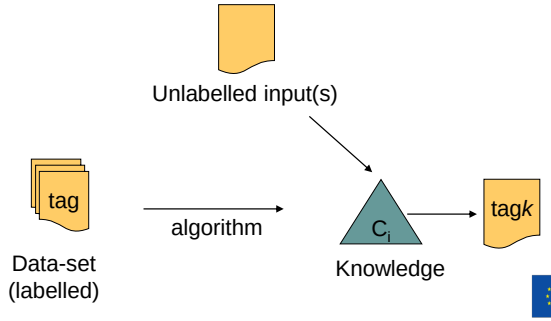
- It is a branch of artificial Intelligence.
- Main objective: to make the computers learn in an automatic way
- To extract knowledge in an automatic way from a concrete data domain
- Computers learn form the information given as input
- To obtain knowledge, it needs a **dataset** as input.
- Analysing data, **using induction,** the computer will obtain conclusions nad build a knowledge pattern **(C_i)**.
- This will be made using **programs or algorithms**.



Introduction

HAP/LAP

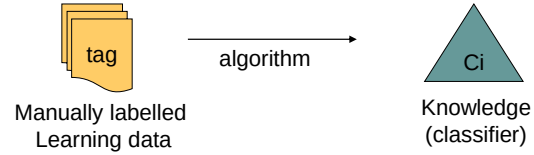
- The algorithms obtain input data, generalize the information found there, and generate models able to **make decisions based on "experience",** to solve the objective problem.
- The generated models can be used to label new



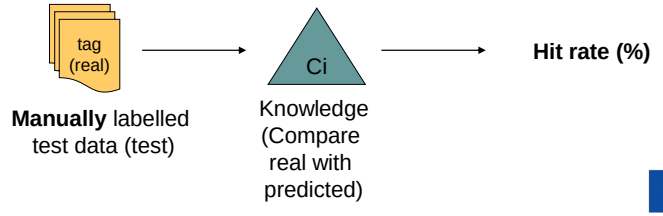
Introduction

HAP/LAP

Training→ inductive generalization of the input data



Decide how good is the built model *(test)*

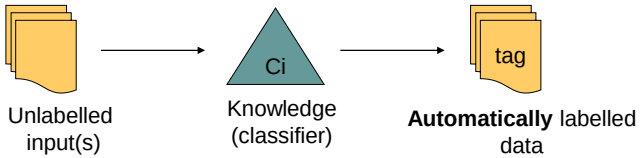


Introduction

HAP/LAP

Aim→ make decisions about unlabelled data in applications

obtain results



Introduction

HAP/LAP

Applications related to Language Processing:

- Information retrieval
- Natural language understanding
- Search engines
- Sentiment analysis (or opinion mining)
- Speech and handwriting recognition
- Syntactic pattern recognition

Introduction

HAP/LAP

LT Modules

Lexical / Morphological Analysis
PoS Tagging
Chunking
Syntactic Analysis
Word Sense Disambiguation
Semantic Role Labelling
Named Entity Recognition
Semantic Analysis
Coreference Resolution
Discourse Analysis

Texts

Knowledge

Applications

OCR
Spelling Error Correction
Information retrieval
Document Classification
Grammar Checking
Information Extraction
Summarization
Question Answering
Ontology Extraction
Dialogue Systems
Machine Translation

Introduction

HAP/LAP

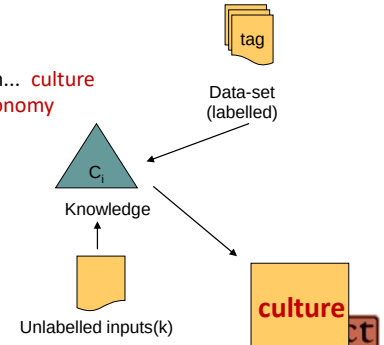
Example: document classification

What to learn: category (culture, politics, sports, …)
From where to learn: labelled documents
(newspapers, journals, news agencies…)
How to learn: rules, probabilities, …

Input (labelled documents)
Generation of a film library in the museum... culture
The plan against the economic crisis economy

Classifier (inductive generalisation)
Film, library, book, art museufm ... culture
Crisis, euro, bank, work. .. economy

Output (label new unlabelled data)
The film festival will start today with...



Introduction: Problems

HAP/LAP

- We will find problems to process and understand language:
  - Mistakes in the input: not found in the dictionary
    - Exchanged characters: lsat → last
    - Missing characters: prgram→ program
  - Not a lot of importance is given to tokenisation but
    - Blank spaces are not enough to tokenise
    - Percentages: 25 % (two tokens?)
    - Initials: J.M. / E.H.U.
    - Others: in sections a.1, a.2 and a.3
  - Ambiguity
    - *I saw the woman with the telescope (2 meanings)*
    - *I saw the woman on the hill with the telescope (4 meanings)*
    - *I saw the woman on the hill in Texas with the telescope (8 meanings)*

We have a very wide knowledge. We use further information than the one we see.

Introduction: Problems

HAP/LAP

Ambiguity wherever:

Speech recognition demo
"Recognize speech" vs. "Wreck a nice beach"
"vice president Gore" vs. "dice precedent core"

Syntactic analysis
I ate rice with tomato, vs, I ate rice with fork

Semantic analysis
Michel runs a company vs. Michel runs a marathon

Pragmatic analysis
– Aitor: Does your dog bite?
– Mikel: Not.
Aitor is gone to play with the dog and the dog bites him.
– Aitor: You told me that your dog does not bite
Mikel: That one is not my dog.

Introduction: Problems

HAP/LAP

The aim in many tasks is to solve ambiguity

For example to obtain correct translations:
We need to solve syntactic and semantic ambiguity
– John plays the guitar → John toca la guitarra
– John plays soccer → John juega al futbol

Knowledge to select the correct interpretation:
– Phonetic/phonologic
– Morphologic/syntactic
– Semantic
– Knowledge of the word/ know how

- Fork tool to eat
- Tomato type of food
- Guitar is a type of musical instrument
- Football is a game ....

Introduction: Problems

HAP/LAP

We might be able to read the following text but a machine…

f y cn rd ths thn y r dng btr thn
ny autmtc txt nrmlztion prgrm cn do.

if you can read this then you are doing better than any automatic text normalization program can do.

And also the next one:

Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in what oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at the rghit pclae. The rset can be a total mses and you can sitll raed it wouthit porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe.

Introduction

- We need to answer some questions:
  - **What** to learn? We need to have an **objective**
  - **Where** to learn it **from**? **Data** is required
    - Corpora
  - **How** to **represent** it?
    - Which characteristics to use to learn features, weight, …
  - **How** to **learn**? A methodology is required
    - Selection of the adequate algorithm
  - **How** to **evaluate**? Is it correct what we learnt?

Introduction

What to learn?

- The program will always have an **objective**:
  - Category of the text
    - Health, economy, politics, …
  - Syntactical function of the words
    - Subject, object, verb,
  - Word sense disambiguation
    - Glass = material
    - Glass = container for drinks
  - Finding and classifying entities
    - place, person,
  - Number classification
    - dates, %, roman, …

Classification problems

Introduction

From where to learn? Corpora

- The machine needs corpora (labelled data) to learn:
  - For each task data needs specific labels
  - When bigger the amount of data better will the system learn
  - The situation is very different depending on the language
    - English (great volume)
    - Basque limited (small community language)
    - Generating corpora is expensive (prepare and validate)
  - Generated for competitions (task dependent)
    - SemEval (**Sem**antic **Eval**uation)
    - TREC (**T**ext **RE**trieval **C**onference)
    - MUC (**M**essage **U**nderstanding **C**onference)
    - TAC (**T**ext **A**nalysis **C**onference)

  Corpus generation requires craft work (linguists)

Introduction

From where to learn? Corpora

- Reuters Corpora (Reuters news agency)
- http://trec.nist.gov/data/reuters/reuters.html
  - Very used in text classification
  - RCV1 (Reuters Collection Volumen 1) 1996-1997
    - English
    - 810,000 Reuters News stories.
    - It requires about 2.5 GB for storage of the uncompressed files.
  - RCV2 (Reuters Collection Volume 2) 1996-1997
    - Multilingual
    - 487,000 Reuters News stories
    - Thirteen languages (Dutch, French, German, Chinese, Japanese, Russian, Portuguese, Spanish, Latin American Spanish, Italian, Danish, Norwegian, Swedish).
    - The stories are NOT PARALLEL, but are written by local reporters in each language.
  - TRC2 (Thomson Reuters Text Research Collection) 2008-2009
    - 1,800,370 news stories

Introduction

From where to learn? Corpora

TREC: Text Retrieval and Evaluation Conferences collections
http://trec.nist.gov/

- Selections from the Wall Street Journal, the New York Times, Ziff-Davis Public. …
- Documents for different tasks
- Question-answering, Spam, …

Text Analysis Conference-2015

- Entity-Discovery and Linking: English, Chinese, Spanish
- Slot-Filling: : English, Spanish

SemEval-2016-2017

- Textual Similarity and Question Answering
- Sentiment Analysis
- Semantic Parsing, Semantic Analysis, Semantic Taxonomy
- Learning sense of humor

NLTK: Natural Language Toolkit
http://www.nltk.org/nltk_data/

Introduction

From where to learn? Corpora

- **MEDLINE/P:** Data set from the National Institute of Health
  - abstracts on medical subjects parsed and indexed

  http://www.bioinformatics.nl/biometa/medlineparsed.html
- **LDC:** The Linguistic Data Consortium
  - An open consortium of universities, companies and government research laboratories.
  - It creates, collects and distributes speech and text databases, lexicons, and other resources for research and development purposes.
  - https://catalog.ldc.upenn.edu/LDC2013T19
- **Penn Tree Bank:**
  - Manually parsed sentences from the Wall Street Journal
  - http://www.cis.upenn.edu/~treebank/
- **Kaggle** is the place to do data science projects
  - https://www.kaggle.com/datasets

Introduction

From where to learn? Corpora

Problems

- Size of corpora → many features
  - Lack of labelled corpora → need of labelling manually?
  - Where to get the data from? publishers, television, web...
    Be careful! permissions, types of licences,...
  - The method used to learn affects to the result
- *data-sparseness*
  - Requires a lot of memory and the information is not always meaningful (half of the different words we can find in a book appears only once)
- Quality of the corpora:
  - Many not meaningful features
  - Errors and noise
  - Examples not in the corpora: probability 0 ???

Introduction

From where to learn? Instances

- What does corpora have?
  - **Instances** or examples of what we want to learn
  - Depending of the aim of the learning process the instances will have different information

    Text Classification → documents

    Word Sense Disambiguation → paragraphs or parts of documents containing the word to be disambiguated

    Named Entity → entities and surrounding words

    Syntactic Function → words do not have enough information, the same word can have different SF. Additional information is required : PoS, NE, …

    …

    Semantic Role Labeling, Coreference, temporal structures, …

Introduction

From where to learn? Features (variables)

HAP/LAP

What do instances have?

Instances are composed of **features**, that is, the data or characteristics we use to define the example.

The **Class** or the category is required

Depending on the learning objective:

Text Classification → documents→ words, lemmas, …

Syntactic Function → phrases→ words, cat, ...

Input: (TC features--> words)

The cyclist of Sky answerd to the questions …

Cylce, Sky, quest, …. **sports**

Introduction

From where to learn? Features (variables)

HAP/LAP

features

Original: (BoW, phonemes, ...)

Texts are not directly treated, they require conversion:

Calculations: obtained preprocessing the original corpus

lemma (stemming), category, case, capital letter, …

Distance from the verb, length, ...

Complete text : BoW (TC)

Text windows : global or local context (NE, WSD, punctuation, ...)

w_{-2}, w_{-1}, w_0, w_{+1}, w_{+2}

Feature value vectors

Category=NAME, ADJ, VERB,

subcategory=COM,

number=S, P, UNC

…

Introduction

From where to learn? Features (variables)

HAP/LAP

- Document classification.

  Input (labelled documents)

  The cyclist of Sky answered to the questions of the press in the hospital room... SPORT
- Finding phrases within a text.

  Input (labelled documents)

  $[\text{The cyclist of Sky}]_{NPH}$ answered to $[\text{the questions of the press}]_{NPH}$ in $[\text{the hospital room}]_{NPH}$ ...
- Identification of the syntactic function of the words

  Input (labelled documents)

  $[\text{The cyclist of Sky}]_{SUBJ}$ answered to $[\text{the questions of the press}]_{OBJ}$ in the hospital room ...
- Word sense disambiguation
- Web page classification

2.- Learning with WEKA software

HAP/LAP

2.- Learning with WEKA software:

2.1.-**Introduction**

2.2.-Preprocessing

Attribute (feature) selection

2.3.-Evaluation

2.4.- Basic ML algorithms: Naive Bayes, K-NN, Decision Trees, Rules, …

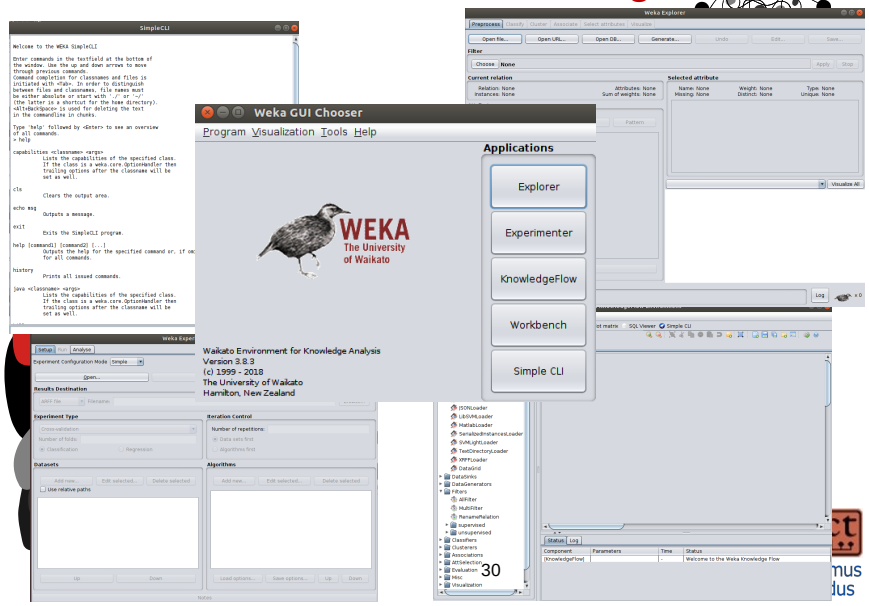
Learning with WEKA software

HAP/LAP

- Machine Learning tool written in Java (GNU license)
- Many types of algorithms:
  - Directly applied to a set of data
  - Possible to call them from your own java program
- Main characteristics:
  - Graphical interface
  - Option to preprocess data (feature selection, …)
  - A lot of learning algorithms (classification, clustering, …)
  - Different evaluation methods ...
- Parts:
  - Data processing part Explorer
  - Part to start experiments Knowledge Flow
  - Method comparison part Experimenter
  - All together Workbench

Erasmus Mundus

WEKA: tool for machine learning



WEKA: bibliography

HAP/LAP

- Book: Data Mining. Practical machine learning tools and techniques. Ian Witten and Eibe Frank, Mark Hall, Christopher Pal (4th Edition) The Morgan Kaufmann, 2016.
  http://www.cs.waikato.ac.nz/ml/weka
- Software
  http://www.cs.waikato.ac.nz/ml/weka
- Weka's wiki
  http://weka.wikispaces.com/
- Programs, data-files, …
  http://www.hakank.org/weka/index.html
  (arff files, programs, ...)
  http://archive.ics.uci.edu/ml/
  (264 databases for machine learning)
  http://www.kdnuggets.com/datasets/index.html
  (websites with databases)

Erasmus Mundus

WEKA: introduction

HAP/LAP

- Input files:
  - Format: ARFF
  - Accessible in the Internet:
    - http://www.hakank.org/weka/index.html
    - http://archive.ics.uci.edu/ml/

Data Set Characteristics:	Multivariate	**Number of Instances:**	569	**Area:**	Life
Attribute Characteristics:	Real	**Number of Attributes:**	32	**Date Donated**	1995-11-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	579838

  - Data from a URL (address where the dataset is stored)
  - From SQL datsets (weka/experimenter)

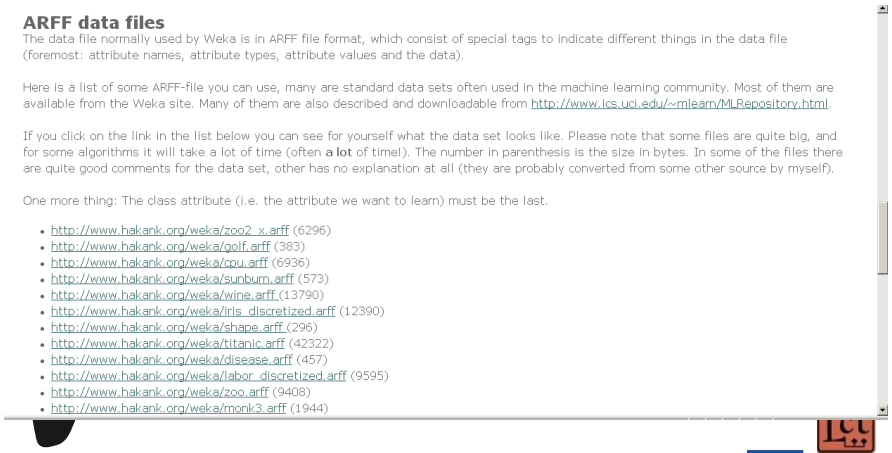
Erasmus Mundus

WEKA: introduction

Data files:

http://www.hakank.org/weka/index.html

HAP/LAP



ARFF data files

The data file normally used by Weka is in ARFF file format, which consist of special tags to indicate different things in the data file (foremost: attribute names, attribute types, attribute values and the data).

Here is a list of some ARFF-file you can use, many are standard data sets often used in the machine learning community. Most of them are available from the Weka site. Many of them are also described and downloadable from http://www.ics.uci.edu/~mlearn/MLRepository.html.

If you click on the link in the list below you can see for yourself what the data set looks like. Please note that some files are quite big, and for some algorithms it will take a lot of time (often **a lot** of time!). The number in parenthesis is the size in bytes. In some of the files there are quite good comments for the data set, other has no explanation at all (they are probably converted from some other source by myself).

One more thing: The class attribute (i.e. the attribute we want to learn) must be the last.

- http://www.hakank.org/weka/zoo2_x.arff (6296)
- http://www.hakank.org/weka/golf.arff (383)
- http://www.hakank.org/weka/cpu.arff (6936)
- http://www.hakank.org/weka/sunburn.arff (573)
- http://www.hakank.org/weka/wine.arff (13790)
- http://www.hakank.org/weka/iris_discretized.arff (12390)
- http://www.hakank.org/weka/shape.arff (296)
- http://www.hakank.org/weka/titanic.arff (42322)
- http://www.hakank.org/weka/disease.arff (457)
- http://www.hakank.org/weka/labor_discretized.arff (9595)
- http://www.hakank.org/weka/zoo.arff (9408)
- http://www.hakank.org/weka/monk3.arff (1944)

Erasmus Mundus

WEKA: introduction

ARFF format

HAP/LAP

- Input files:
  - Format: ARFF (from excel: http://exceltoarffconv.sourceforge.net/)
  - Data can be read from URLs (address where the dataset is stored) or SQL data-bases (weka/experimenter)
- ARFF (Attribute-Relation File Format)
  - **Header**:
    - Title `@relation <relation-name>`
    - features, feature-type `@attribute <attribute-name> <datatype>`
      - *numeric* (*integer, real*)
      - {value1, value2, …}: nominal, list of possible values
      - *string*: to work with text files (filters are required)
      - *date:* dates
    - The last feature is the category or class `@attribute <attribute-name> <datatype>`
  - **Data**
    - Instances or examples

`@data`

Erasmus Mundus

WEKA: introduction

input files (.arff)

HAP/LAP

```
@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
```

Numeric features

Nominal

Class

Instances

Erasmus Mundus

WEKA: introduction

example(.arff)

HAP/LAP

doc1:

German Chancellor Angela Merkel has re-affirmed her country's responsibility for the Holocaust, following controversial comments by Israel's prime minister.
Benjamin Netanyahu was criticized for saying Adolf Hitler had only wanted to expel Jews from Europe but that a Palestinian leader, the Grand Mufti of Jerusalem Haj Amin al-Husseini, told him to "burn **POLITICS**

doc2:

Britain's defending Tour de France champion Chris Froome has welcomed a "great" 2016 route which should help his bid for a third victory.
The course for the 2,187-mile race, which runs from 2-24 July, was announced in Paris on Tuesday. **SPORTS**

Which are the features?
How are they defined?

```
@relation  topics
@attribute doc string
@attribute class {sports, politics}

@data
   'German Chancellor Angela Merkel has re-affirmed...'  POLITICS
   'Britain's defending Tour de France champion Chris Froome …'  SPORTS
…
```


Erasmus Mundus

WEKA: introduction

simple example(.arff)

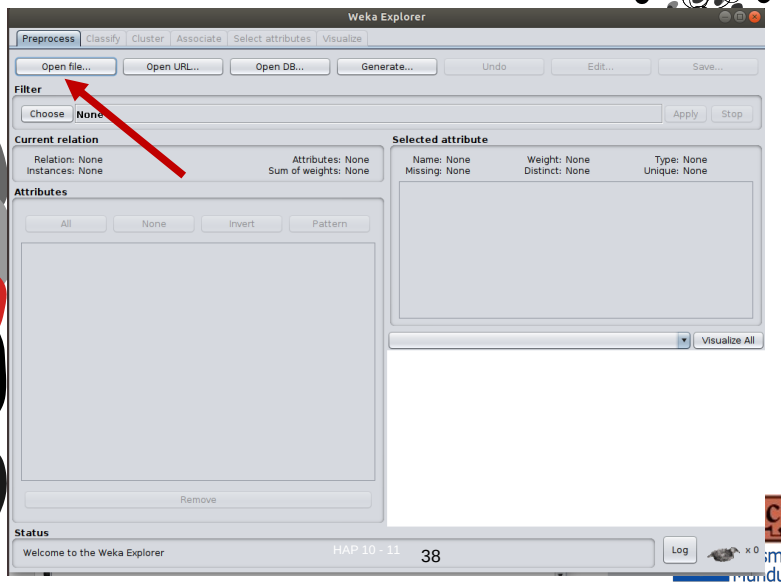
HAP/LAP

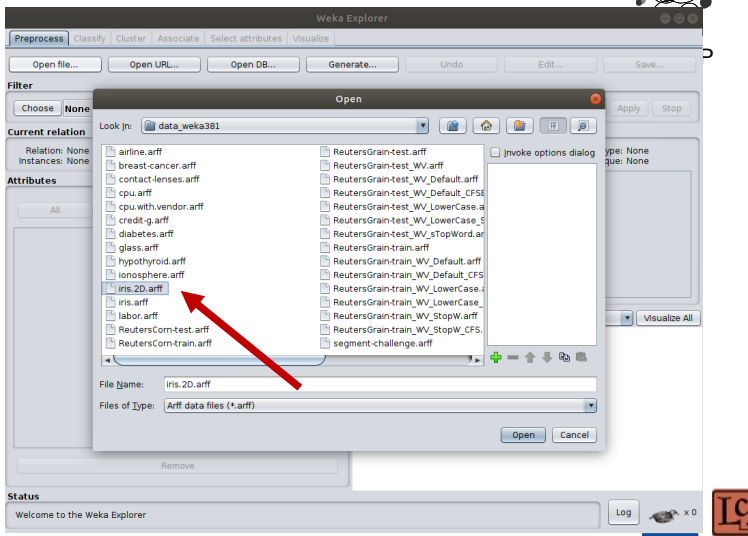
```
@RELATION iris
@ATTRIBUTE sepallength       REAL
@ATTRIBUTE sepalwidth        REAL
@ATTRIBUTE petallength       REAL
@ATTRIBUTE petalwidth        REAL
@ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica}
@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
...
7.0,3.2,4.7,1.4,Iris-versicolor
6.4,3.2,4.5,1.5,Iris-versicolor
...
6.3,3.3,6.0,2.5,Iris-virginica
5.8,2.7,5.1,1.9,Iris-virginica
...
```

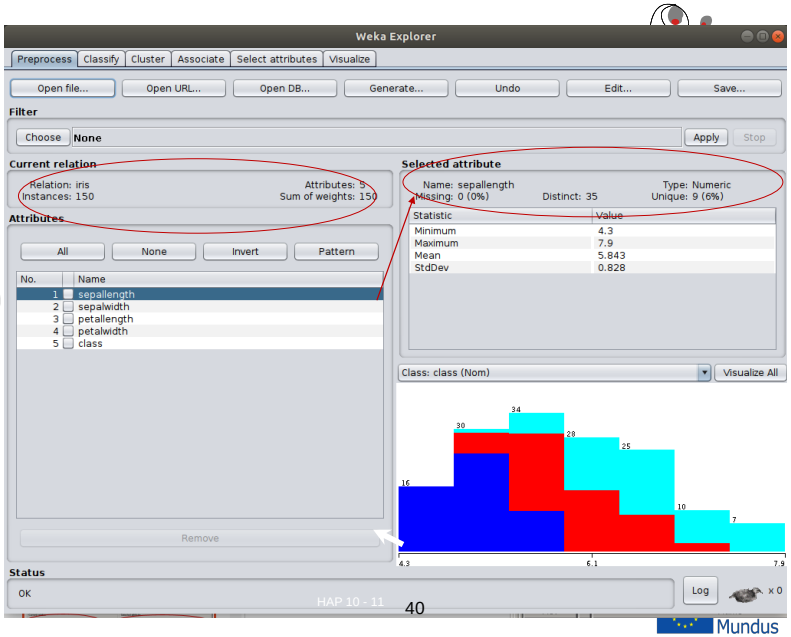
sepallength

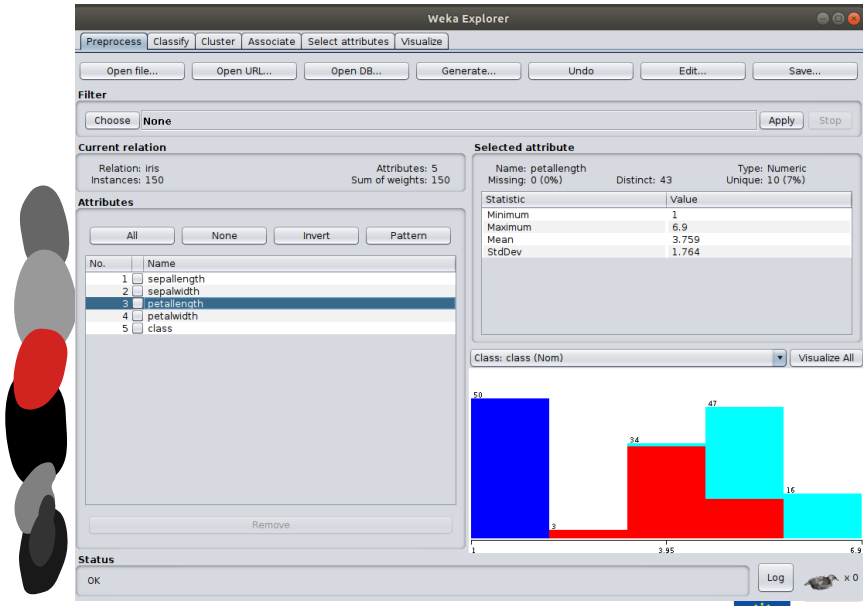
class

- **Open files:** Preprocess / Open file



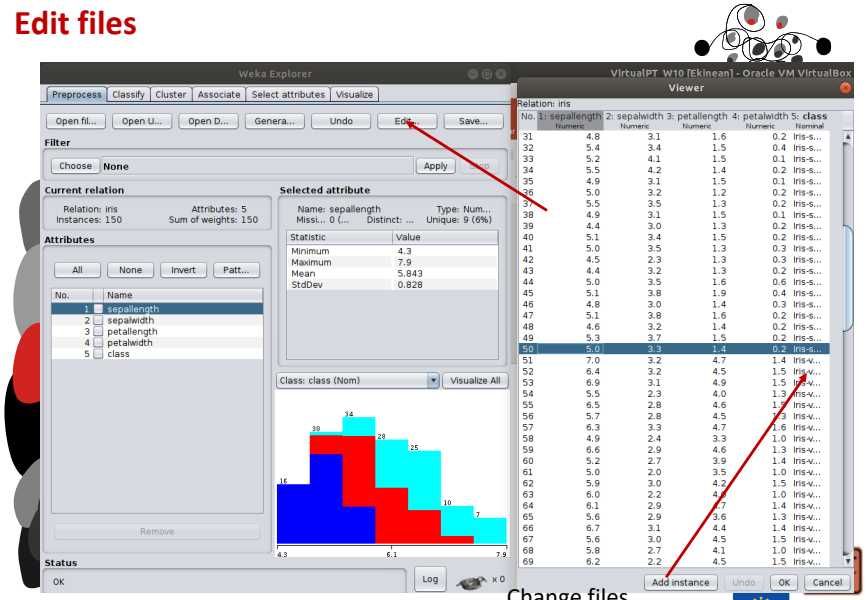




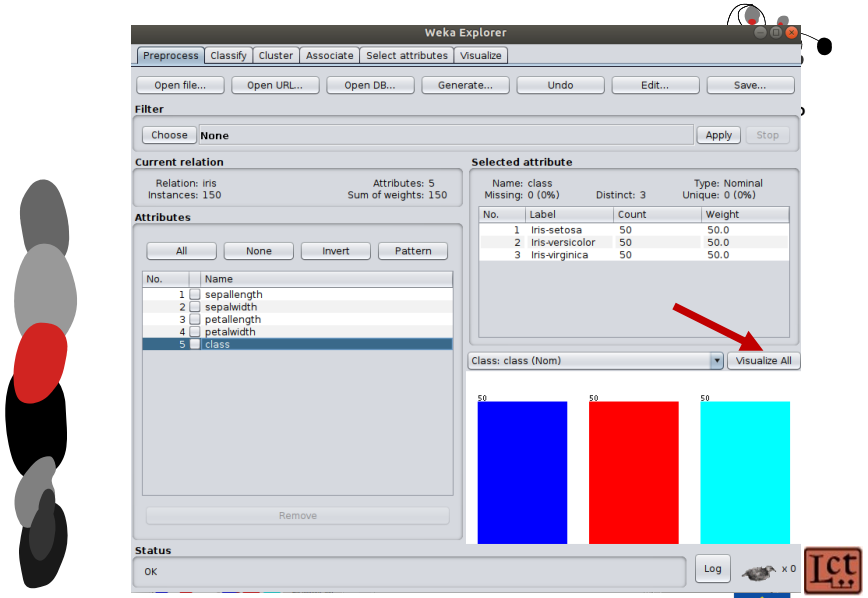


41

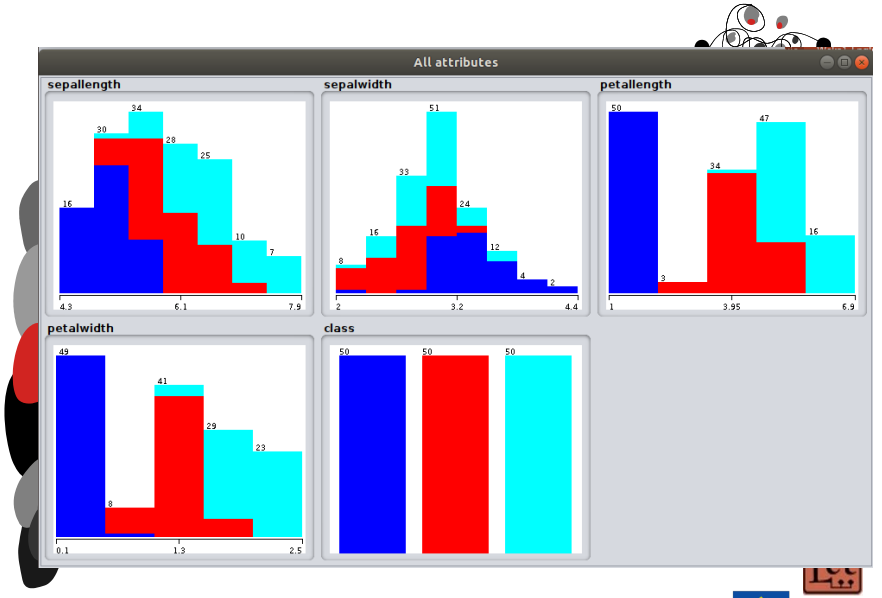
Edit files



42 Change files



43



44

WEKA: introduction

HAP/LAP

Is the dataset balanced?

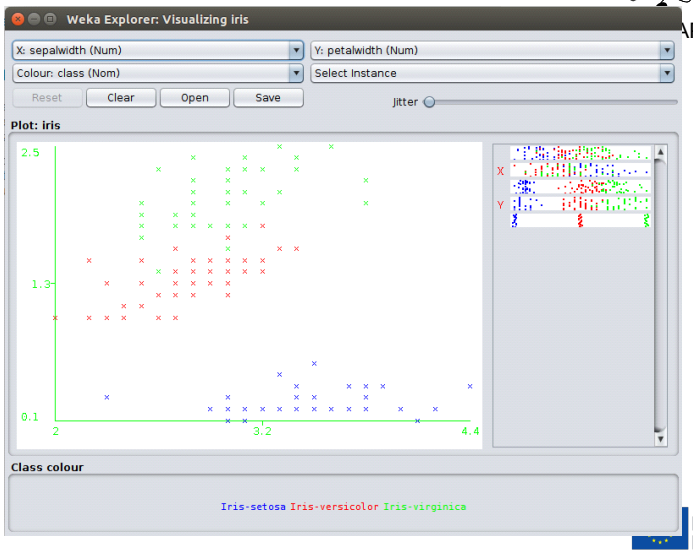
Which type of analysis do the graphics show?

Which variables seem to be more related to the class?

Which variables would you use for classification?

Erasmus Mundus

Visualize:



soybean.arff

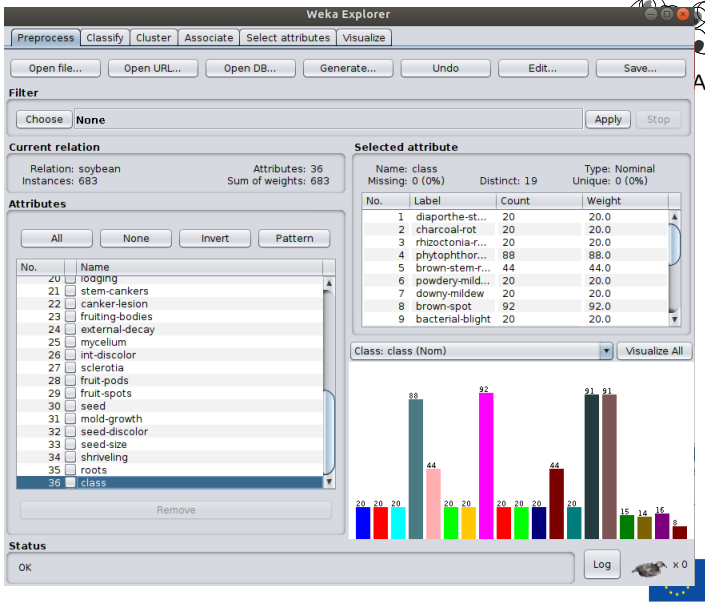
35 features 683 instances 19 classes

HAP/LAP

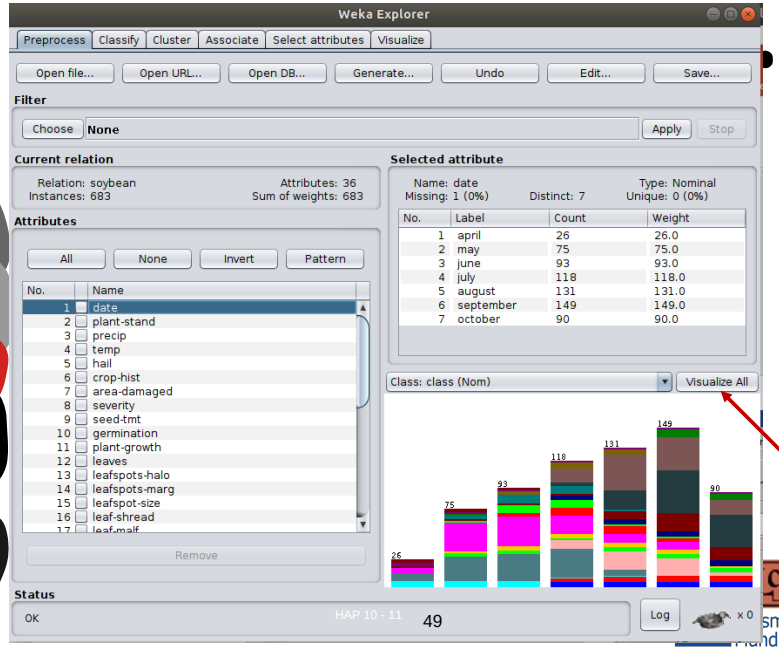
```
@RELATION soybean
@ATTRIBUTE date    {april,may,june,july,august,september,october}
@ATTRIBUTE plant-stand      {normal,lt-normal}
@ATTRIBUTE precip  {lt-norm,norm,gt-norm}
@ATTRIBUTE temp    {lt-norm,norm,gt-norm}
...
@ATTRIBUTE class {diaporthe-stem-canker, charcoal-rot, rhizoctonia-root-rot,
    phytophthora-rot, brown-stem-rot, powdery-mildew, downy-mildew, brown-
    spot, bacterial-blight, bacterial-pustule, purple-seed-stain, anthracnose,
    phyllosticta-leaf-spot, alternarialeaf-spot, frog-eye-leaf-spot,
    diaporthe-pod-&-stem-blight, cyst-nematode, 2-4-d-injury, herbicide-
    injury}

@DATA
october, normal, gt-norm, norm, yes, same-lst-yr, low-areas, pot-severe, none,
    90-100, abnorm, abnorm, absent, dna, dna, absent, absent, absent, abnorm,
    no, above-sec-nde, brown, present, firm-and-dry, absent, none, absent,
    norm, dna, norm, absent, absent, norm, absent, norm, diaporthe-stem-canker
august, normal, gt-norm, norm, yes, same-lst-two-yrs, scattered, severe,
    fungicide, 80-89, abnorm, abnorm, absent, dna, dna, absent, absent,
    absent, abnorm, yes, above-sec-nde, brown, present, firm-and-dry, absent,
    none, absent, norm, dna, norm, ....
```

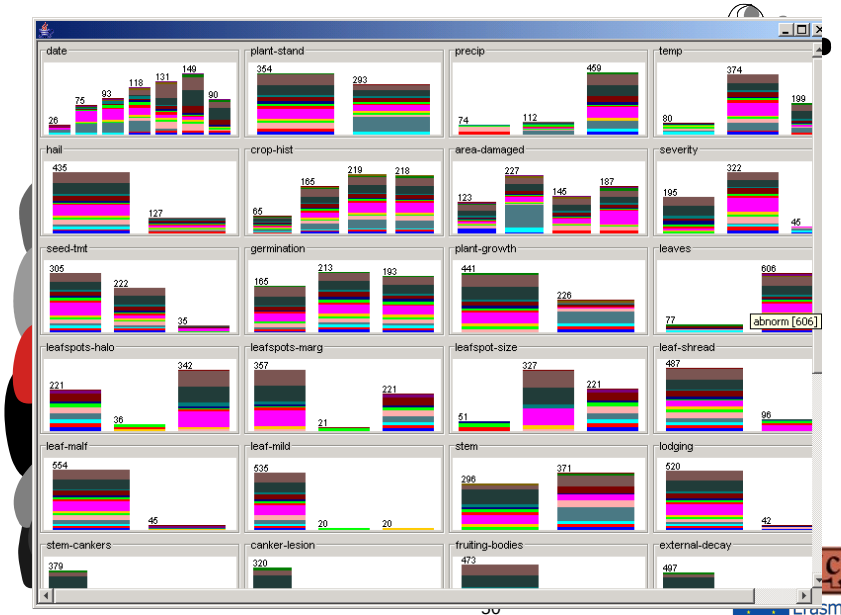

Erasmus Mundus

soybean.arff



Erasmus Mundus





WEKA: introduction

HAP/LAP

How many classes does the dataset have?

Is the dataset balanced?

Which variables seem to be more related to the class?

Which variables would you use for classification?

WEKA: introduction

HAP/LAP

Visualize. Texts

Reuters_Corn

```
@attribute Text string
@attribute class-att {0,1}

@data
'ASIAN EXPORTERS … dispute.\n REUTER\n&#3;',0
'CHINA DAILY … details.\n REUTER\n&#3;',0
'THAI TRADE …  35 pct.\n REUTER\n&#3;',1
```

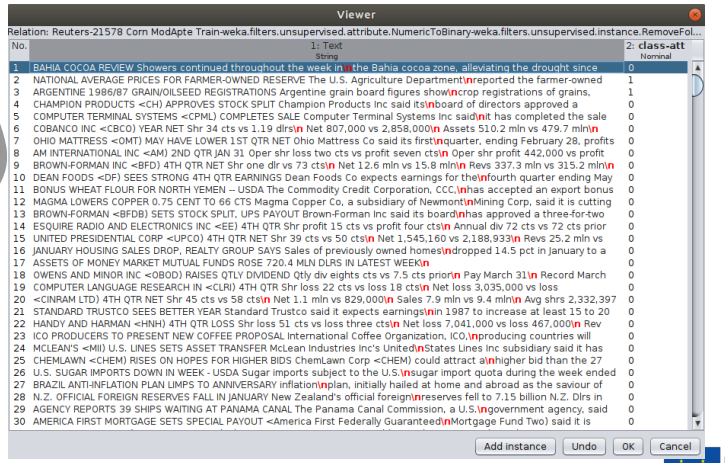
Assignment: try to see reuters_corn_test database with weka
- Visualize and edit

WEKA: introduction

Texts

`Reuters_Corn: special characters\n, <, -, numbers`

HAP/LAP

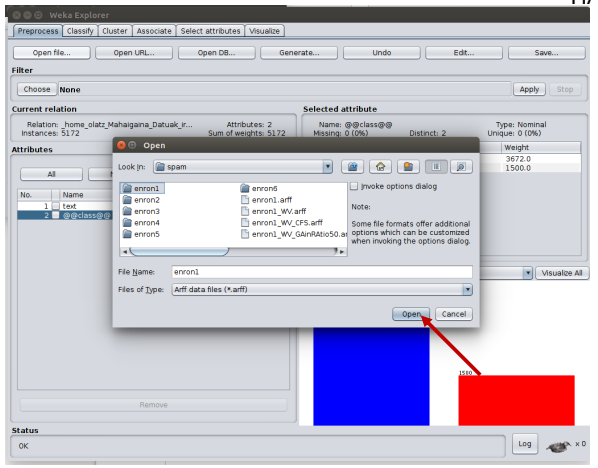


WEKA: introduction

Texts from directories

Databases in folders. Spam (enrom1-6)

HAP/LAP

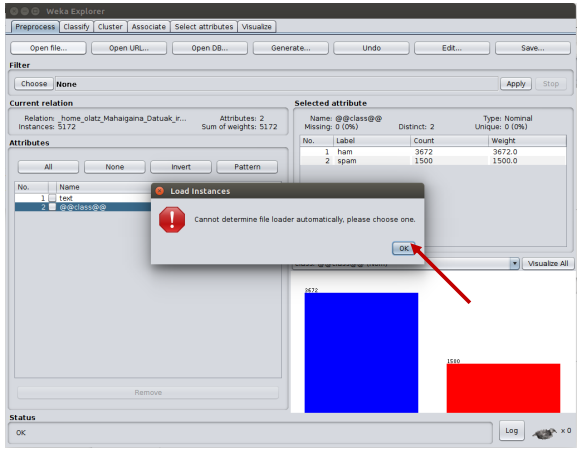


WEKA: introduction

Texts from directories

Databases in folders. Spam (enrom1-6)

HAP/LAP

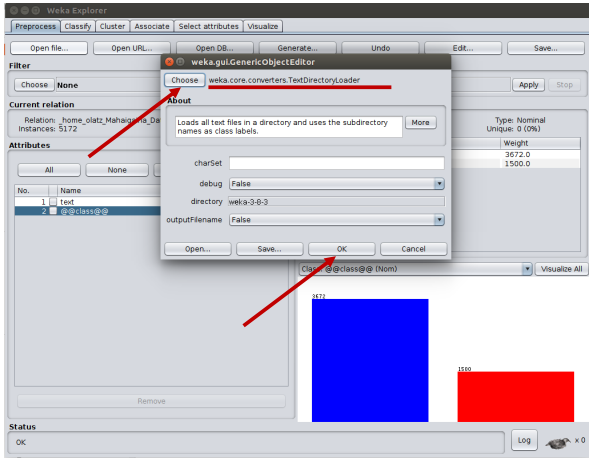


WEKA: introduction

Texts from directories

Databases in folders. Spam (enrom1-6)

HAP/LAP



WEKA: preprocessing

HAP/LAP

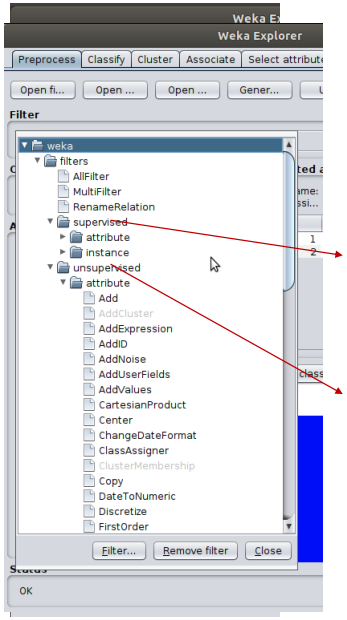
Modifications in features/instances

Instance or feature modifications are often required:

- To adapt to the needs of algorithms (numbers)
- Remove no meaningful features (stop word vector)
- Balance corpora (re-sampling) → delete instances
- …

Options to modify instances and features:

- **Trasnform words in numbers** (String to word vector)
- **Feature selection** (supervised/unsupervised)
- Discretize features (discretize)
- Normalize words (String to nominal)
- Convert nominal to number(Nominal to binary)
- Normalize numeric features (normalize)
- Clean numeric features (delete very big or very small values)
- Generate/change features with mathematical operations
- Principal compoonent analysis (PCA)
- …



Filters

HAP/LAP

Filters:

to make "changes"

in the features or instances

Supervised

based on the data

- features
- Instances

Unsupervised

use the information

given by the user

- features
- Instances

WEKA: preprocessing

HAP/LAP

Working with texts: features

- Problems with words:
  - Many algorithms do not treat them as texts
  - Transformation. Convert into number: bag of words (BoW) or vector space model (VSM)

document *x* is represented by vector $\phi(x)$ (pre)defined dictionary

- Ex: document classification
  - d_i = book presentation in Koldo Mitxelenan …
  - d_j = the writer will talk about the book in Koldo M.
  - d_k = Europes' economy … the euro has risen …
- Representation (lemmas)
  - d_i = book present Koldo Mitxelena …
  - d_j = writer talk about book Koldo Mitxelena …
  - d_k = Europe economy … euro rise…

- Features (dictionary):

{writer, book, present, author, novel, …, read, Koldo, Mitxelena, have, Europe, economy, euro, rise}

$d_1 = x \rightarrow \phi(x) = \{0,1,1,0,0,....,0,1,1,0,0,0,0,0\}$
$d_2 = z \rightarrow \phi(z) = \{1,1,0,0,0,....,0,1,1,1,0,0,0,0\}$
$d_3 = v \rightarrow \phi(x) = \{0,0,0,0,0,....,0,0,0,1,1,1,1,1\}$

WEKA: preprocessing

HAP/LAP

Texts

Reuters_Corn

Weka **can not treat** the data as it is represented here**: they are not numeric attributes neither nominal**

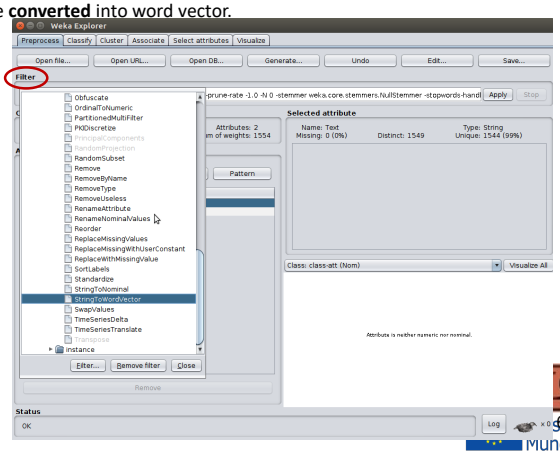
→ The **text** needs to be **converted** into word vector.

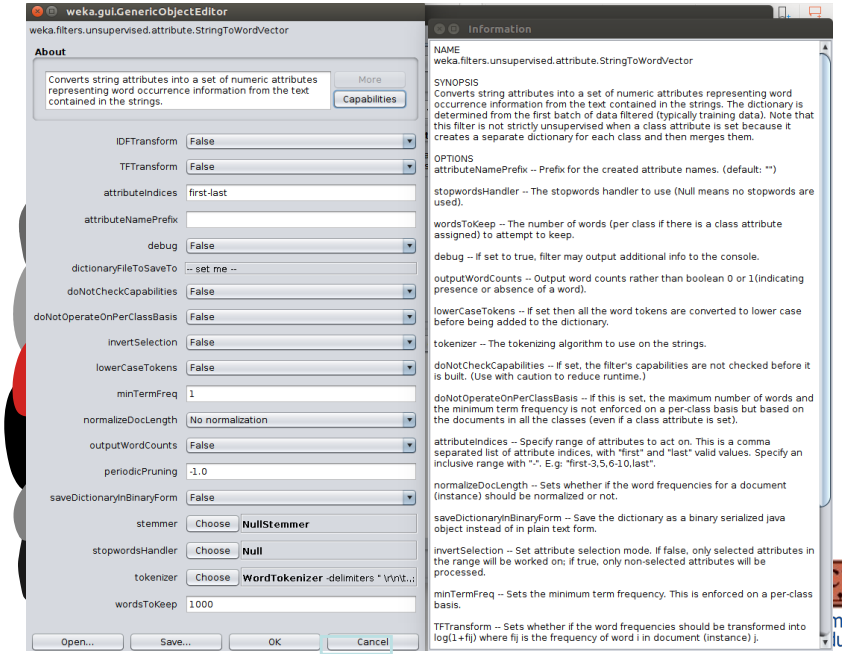
→ **Filter**

→ Unsupervised

→ Attribute

→ StringtoWordVector





WEKA: preprocessing

Texts

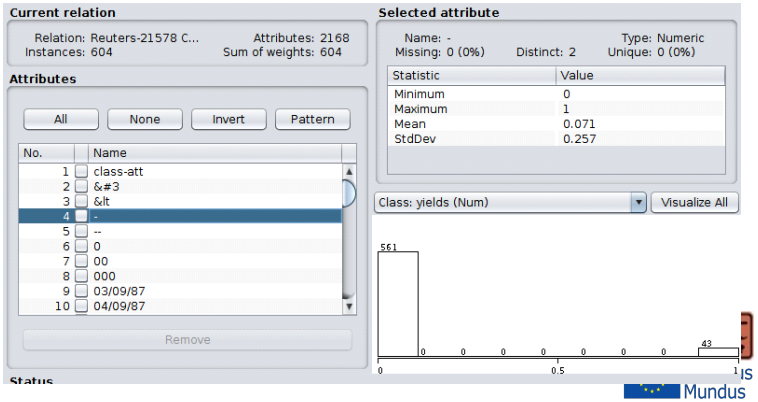
Reuters_Corn

HAP/LAP

Weka **can not treat** the data as it is represented here**: they are not numeric attributes neither nominal**

→ The **text** needs to be **converted** into word vector

→ 2168 features (dictionary)



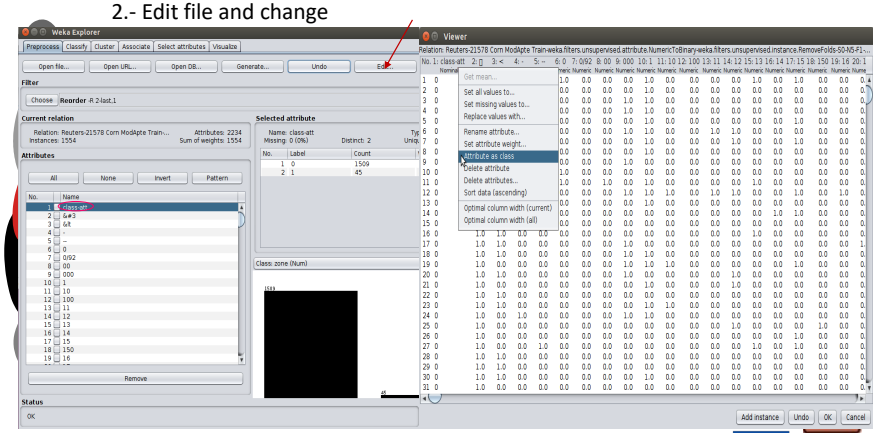
Mundus

WEKA: preprocessing

Filters: StringToWordVector

HAP/LAP

- To move the category to the last position:
  1.- weka.filters.unsupervised.attribute.Reorder (Reorder 2-last,1)
  2.- Edit file and change

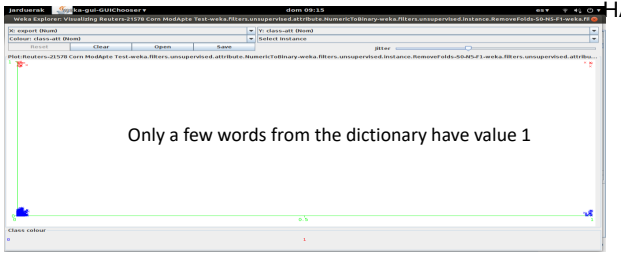


Erasmus Mundus

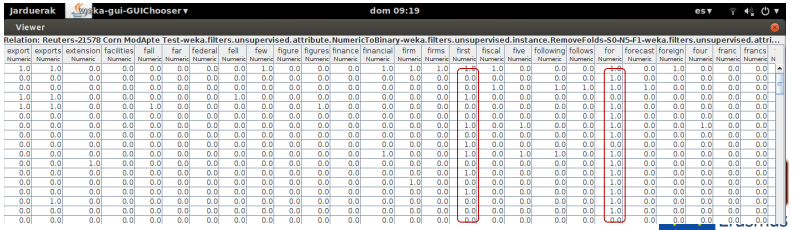
WEKA: preprocessing

Weka. Texts

HAP/LAP



Only a few words from the dictionary have value 1

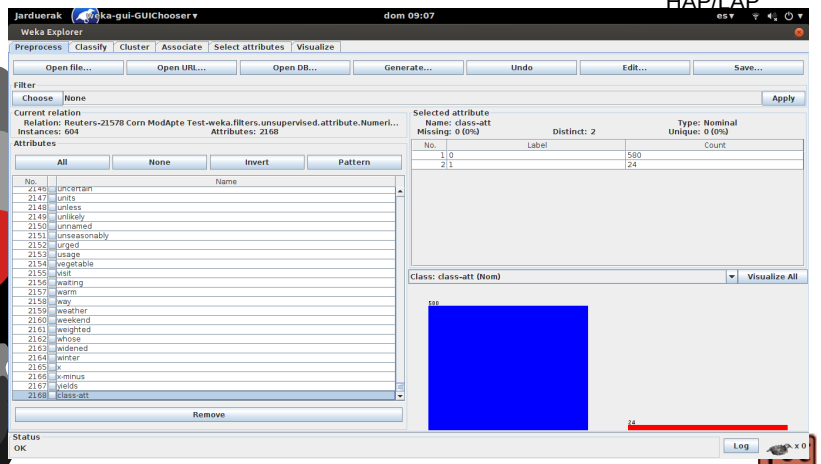


Mundus

WEKA: preprocessing

Weka. Texts

Reuters_Corn: unbalanced



HAP/LAP

Erasmus Mundus

WEKA: preprocessing

Working with texts: attribute selection

HAP/LAP

Selecting features

Stop word-list:

- Delete functional words
- Delete very frequent words (they do not inform)
- Delete rare words (errors, …)

Document frequency #Tr(tk): in how many documents the word appears

- It is possible to reduce the size of the datasets 10 times
- No efficiency lost
- If #Tr is very small or very big, the word is not important.
- Delete words appearing in 1 or two documents

Erasmus Mundus

WEKA: preprocessing

Filters: StringToWordVector

HAP/LAP

- Options of the filter:
  - Word frequency: weight for the words
  - Number of occurrences
  - Capital letters

Tokenizer: divide the words of the test
limits: blank. , ; : ‘ “ ( ) ? !

Stemmer: root of the word
Porter algorithms (most used)
http://snowball.tartarus.org/algorithms/porter/stemmer.html
http://weka.wikispaces.com/Stemmers

StopList: words without “value”, functional words (depends on the task)
or, and, is, … http://www.ranks.nl/stopwords

WordsToKeep: maximum number of words to keep for each class.

Erasmus Mundus

WEKA: preprocessing

Filters: StringToWordVector

HAP/LAP

- Options of the filter:
- Analyse how the number of features changes for Reuters_Corn_train including
  - LowerCase
  - Stopwords
  - Stemmer (use package manager to add stemmer)
  - Minimum term frequency....

- Have a look to the database, do you consider the number of features adequate?

Erasmus Mundus

WEKA: preprocessing

HAP/LAP

Working with texts : features

- The vectors can be **binary** or **weighted**
  - **Binary**: the term appears (1), does not appear (0).
    → default option
  - **Term frequency (tf)**: relative number of occurrences of the term
    → f_{ij} (where f_{ij} is the frequency of word i in document j).
  - **Document frequency (df)** : number of training documents where the term t_k appears at least once
  - **Inverse document frequency (Idf)** : to decrease the weight of words that appear very often in the corpus and increase it for words appearing few times
  - **term frequency – inverse document frequency**

    tf-idf: if the term appears often→ it is meaningful

    if it appears in many texts → it is not discriminator

Erasmus Mundus

WEKA: preprocessing

HAP/LAP

Working with texts : features

Term frecuency – inverse document frecuency (tf-idf)

tf-idf: if it appears often in the text→ is meaningful

if it appears in many texts → it is not discriminant

$$tfidf(t_k, d_j) = \#(t_k, d_j).\log \frac{|Tr|}{\#Tr(t_k)}$$

$\#(t_k, d_j)$ → number of occurrences of term *k* in document *j*

$\#Tr(t_k)$ → number of documents in train where t_k appears at least once

(often 1 + $\#Tr(t_k)$ to avoid divisions with 0

Tfidf	t1	t2	t1	t2	t1	t2
Tr	100	100	100	100	100	100
# (tk, dj)	3	12	3	3	20	20
# Tr (tk)	50	50	90	10	10	90
tfidf (t1, dj)	0,9	3,6	0.13	3	20	0.9
	t1 appears few times	t2 appears often	t1 appears few times in many documents	t2 appears few times in few documents	t1 appears many times in few documents	t2 appears many times in manydocuments

Erasmus Mundus

WEKA: preprocessing

HAP/LAP

Example

```
@relation tc

@attribute dok string
@attribute kat {kirola kultura politika}

@data
'Euskaltel Euskadi txirrindulari prentsaurreko' kirola
'liburu aurkeztu koldo mitxelena' kultura
'liburu berri koldo mitxelena idazle' kultura
'liburu idazle koldo mitxelena' kultura
'baloia Euskadi partida' kirola
'baloia partida' kirola
'lehendakaria hauteskundeak' politika
```

weka.filters.unsupervised.attribute.StringToWordVector -R first -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -M 1 -tokenizer "weka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\'\\"()?!\""

Erasmus Mundus

WEKA: preprocessing

HAP/LAP

Example

```
@relation tc

@attribute dok string
@attribute kat {kirola kultura politika}

@data
'Euskaltel Euskadi txirrindulari
prentsaurreko' kirola
'liburu aurkeztu koldo mitxelena' kultura
'liburu berri mitxelena idazle' kultura
'liburu idazle mitxelena' kultura
'baloia Euskadi partida' kirola
'baloia partida' kirola
'lehendakaria hauteskundeak' politika
```

```
   @relation 'tc-weka.filters.unsupervised…

   @attribute kat {kirola,kultura,politika}
0  @attribute Euskadi numeric
1  @attribute Euskaltel numeric
2  @attribute baloia numeric
3  @attribute partida numeric
4  @attribute prentsaurreko numeric
5  @attribute txirrindulari numeric
6  @attribute aurkeztu numeric
7  @attribute berri numeric
8  @attribute idazle numeric
9  @attribute koldo numeric
10 @attribute liburu numeric
11 @attribute mitxelena numeric
12 @attribute hauteskundeak numeric
13 @attribute lehendakaria numeric
```

binary

```
@data
{0 1,1 1,4 1,5 1}
{6 1,9 1,10 1,11 1,14 kultura}
{7 1,8 1,10 1,11 1,14 kultura}
{8 1,10 1,11 1,14 kultura}
{0 1,2 1,3 1}
{2 1,3 1}
{12 1,13 1,14 politika}
```

tfidf

```
@data
{0 0.868349,1 1.348802,4 1.348802,5 1.348802}
{6 1.348802,9 1.348802,10 0.587302,11 0.587302,14 kultura}
{7 1.348802,8 0.868349,10 0.587302,11 0.587302,14 kultura}
{8 0.868349,10 0.587302,11 0.587302,14 kultura}
{0 0.868349,2 0.868349,3 0.868349}
{2 0.868349,3 0.868349}
{12 1.348802,13 1.348802,14 politika}
```


Erasmus Mundus

WEKA: preprocessing

Working with texts : attribute selection

HAP/LAP

Relationship between the word (t_k) and the category (c_i)

Use of a function to select features:

InformationGain

Mutual Information

Chi-square, ...

It takes into account the distribution of the word in different categories

It is possible to reduce the size of the datasets 100 times

Use of more complex techniques

(SVD) Singular Value Decomposition

Erasmus Mundus

WEKA: preprocessing

Filters (attribute selection)

HAP/LAP

Feature selection: **KISS**

- **Attribute selection**:
  - **Objective**: to find the most significant feature or set of features to do the prediction
  - It often happens to have features in the data set that hardly affect to the classification
  - **Theoretically** classifiers do not take these features into account but practically they do
  - Doing the selection the efficiency of the classifier is improved. Time reduction
  - The wrong selection can reduce the classifier's performance in a % 5 - % 10
- **Options:**
  - ***Filter***: selection before the learning process starts
  - ***Wrapper***: the learning algorithm participates in the selection

Erasmus Mundus

WEKA: preprocessing

Filters (attribute selection)

HAP/LAP

Optimisation problem

Two phases:

Evaluation: measure the quality or the adequacy of the feature

1.- analyse set of features (...SubsetEval)

2.- analyse features one by one (...AttributeEval)

Search: to search for the most adequate set of features in the feature space

- Ranker
- Best First....

Not every evaluation and search combination is possible

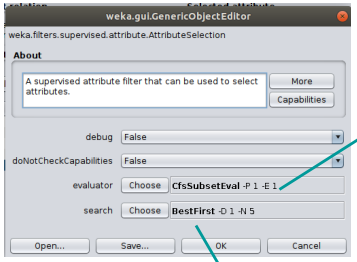
Erasmus Mundus

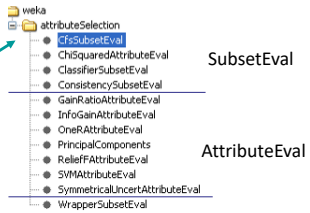
WEKA: preprocessing

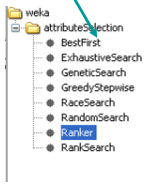
Filters (attribute selection)

Evaluation + search

HAP/LAP







Search

Evaluation

Erasmus Mundus

WEKA: preprocessing

Filters (attribute selection)
Search. Weka

HAP/LAP

Optimisa

- BestFirst: If the evaluation is worse when expanding a set, the process continues with the best subset of the previous stage. (backtracking)
- GreedyStepwise: includes each feature independently to the set and selects the one with the best evaluation
- Ranker: the cheapest. It just orders the features without searching for the best order.

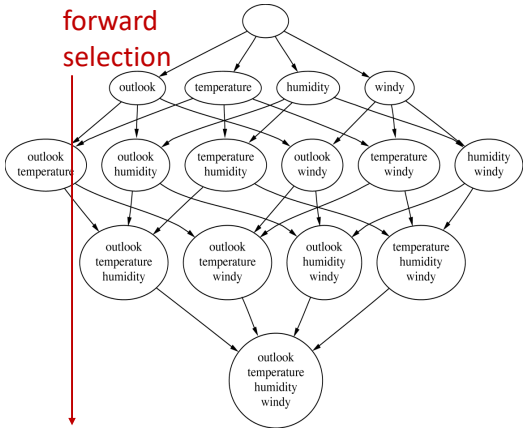
Erasmus Mundus

WEKA: preprocessing

Filters (attribute selection)
Best first (attribute subsets for weather database)

HAP/LAP

forward selection



1. Quality of the subset evaluated when a new feature is included/deleted.
2. Select the best one and continue with the next feature.
3. If no improvement is obtained finish with the search.

backward elimination

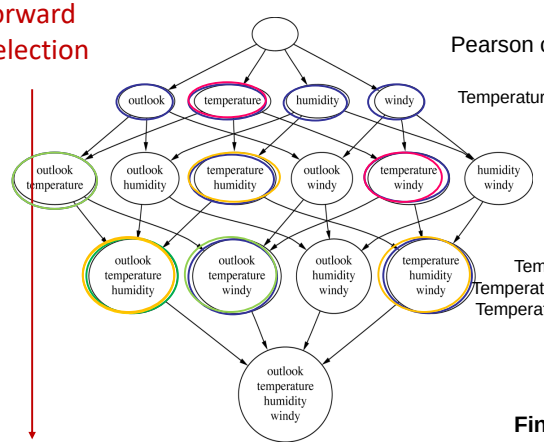
Erasmus Mundus

WEKA: preprocessing

Filters (attribute selection)
Best first

HAP/LAP

forward selection



Pearson correlation coeficient

Temperature > outlook, humidity and windy

Temperature + windy > others

Temperature + windy > (o+t+w) and (t+h+w)
Temperature + windy > (o+t+h) and (t+h+w)
Temperature + windy > (o+t+h) and (o+t+w)

↓

Temperature + windy

Finds the local optimum

Erasmus Mundus

WEKA: preprocessing

Filters (attribute selection)
Evaluation. Weka

HAP/LAP

Sets of features

CfsSubsetEval: selets features with high correlation with the class.

ClassifierSubsetEval: uses a classifier for selection.

WrapperSubsetEval: similar to the previous one with Cross-validation.

Individual features:

InfoGainAttributeEval:

Evaluates the quality of the features by analysing the information gain obtained for the class
InfoGain(Class, Attribute) = H (Class | Attribute) where H is entropy

GainRatioAttributeEval:

Evaluates the quality of the features by analysing the information gain ratio obtained for the class
InfoGain(Class, Attribute) = H(class)-H (Class | Attribute) where H is entropy

ChiSquaredAttributeEval: c2

Symmetrical Uncertainty.

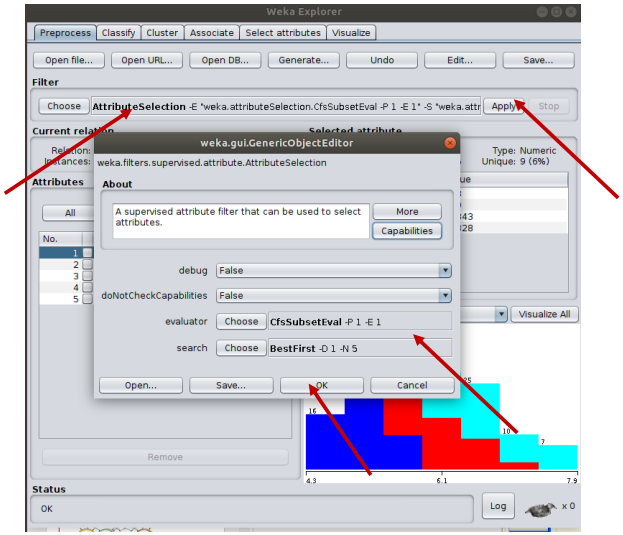
SymmU(Class, Attribute) = 2 * (H(Class) - H(Class | Attribute)) / H(Class) + H(Attribute).

Relief AttributeEval...

Erasmus Mundus

Filters (attribute selection)

BestFirst + CfsSubsetEval

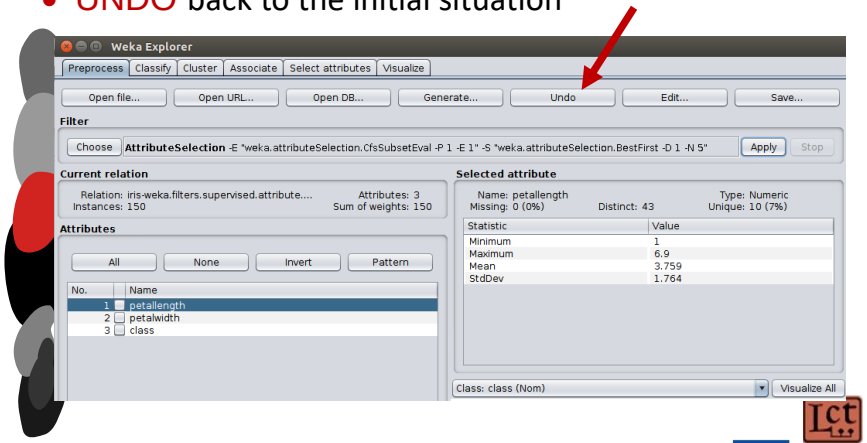


HAP/LAP

WEKA: preprocessing

Filters (attribute selection)

- UNDO back to the initial situation

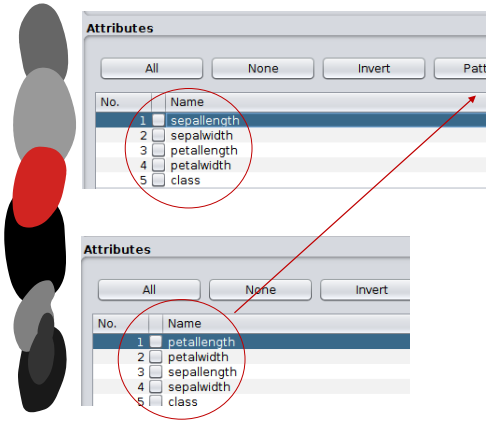


HAP/LAP

WEKA: preprocessing

Filters (attribute selection)

InfoGain (evaluation) + Ranker (search)



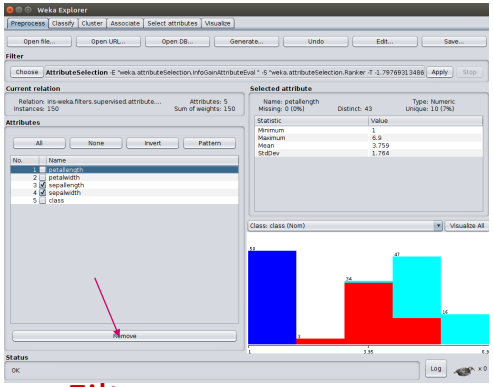
- Features ordered according to the evaluation method
- Organized in the new order
- If we want to delete them:
  - manually (remove)
  - use threshold (menu)
  - limit number (numToSelect)

HAP/LAP

WEKA: preprocessing

Filters (attribute selection)

InfoGain + Ranker

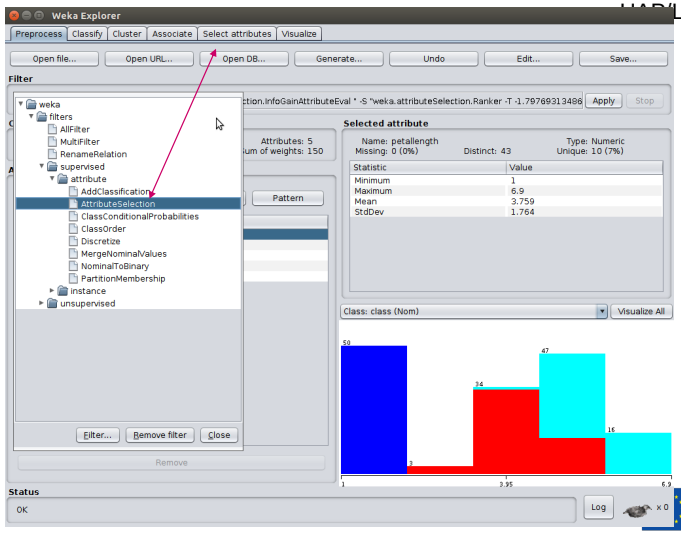


Filters: Remove

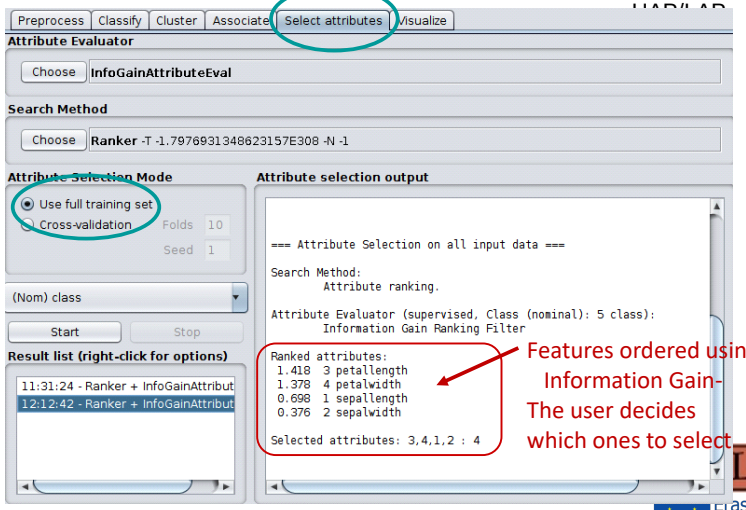
Filter/unsupervised/attribute/remove `attributeIndices`

HAP/LAP

Filters (attribute selection)
Menu. Select attributes



Filters (attribute selection)
Menu. Select attributes



WEKA: preprocessing

Attribute selection
Example (CFSubsetEval + BestFirst)

HAP/LAP

Iris:
4 features (% 96) → 2 features (%96)

Credit_g:
20 features (% 72,6) → 3 features (% 73,2)

Hit rates with j48 algorithm

WEKA: preprocessing

Assignment: Soybean

HAP/LAP

Algorithm: J48
Percentaje split: % 66
- All features Correctly classified instances = % 90,5
- Feature selection:
  - `BestFirst + CfsSubsetEval`
  - `InfoGain + Ranker` changing threshold

	Threshold	N-feat.	%	F_1
BF + CSE				
IG + Rank				
IG + Rank				
IG + Rank				

WEKA: preprocessing

Assignment: Reuters_Grain_Test_WV

Algorithm: J48
Percentaje split: % 66
- All features
  Correctly classified instances = % 95,122 (F_1 = 94,5)
  - Feature selection:
    `BestFirst + CfsSubsetEval`
    `InfoGain + Ranker` (changing *Threshold*)

	Threshold	N-feat.	%	F_1
BF + CSE				
IG + Rank				
IG + Rank				
IG + Rank				

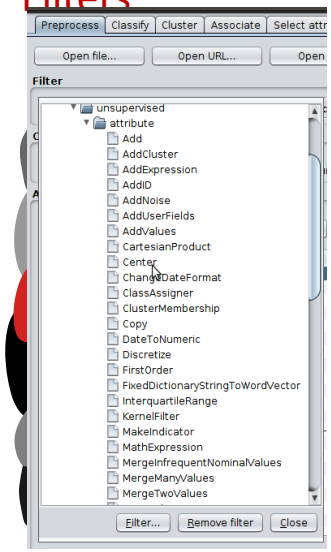
StringToWordVector

Assignment
- Open ReutersGrainTrain
- Apply filter StringToWordVector with default values
- Move the class to the end
- Try J48 classifier (train-test option: *Percentage split)*
- Apply filter StringToWordVector including richer options
- Move the class to the end
- Try J48 classifier
- Select attributes (different options)
- Try J48

	Threshold	N_features	%	F_1
BF + CSE				
I_Gain+Ranker				
I_Gain+Ranker				
SymmetricalUn…				
SymmetricalUn…				

WEKA: preprocessing

Filters



Unsupervised

Attribute:

- **Add:** add feature
- **AddExpresion:** generate new feature based on operation between features
- **AddNoise:** add noyse in attribute values
- **Copy:** copy attributes
- **Discretize**
- **Normalize**
- **NumericToBinary**
- **Remove:** remove features
- **StringToWordVector:** to work with texts