

# **An Introduction to the WEKA Data Mining System**

Zdravko Markov  
Central Connecticut State University  
[markovz@ccsu.edu](mailto:markovz@ccsu.edu)

Ingrid Russell  
University of Hartford  
[irussell@hartford.edu](mailto:irussell@hartford.edu)

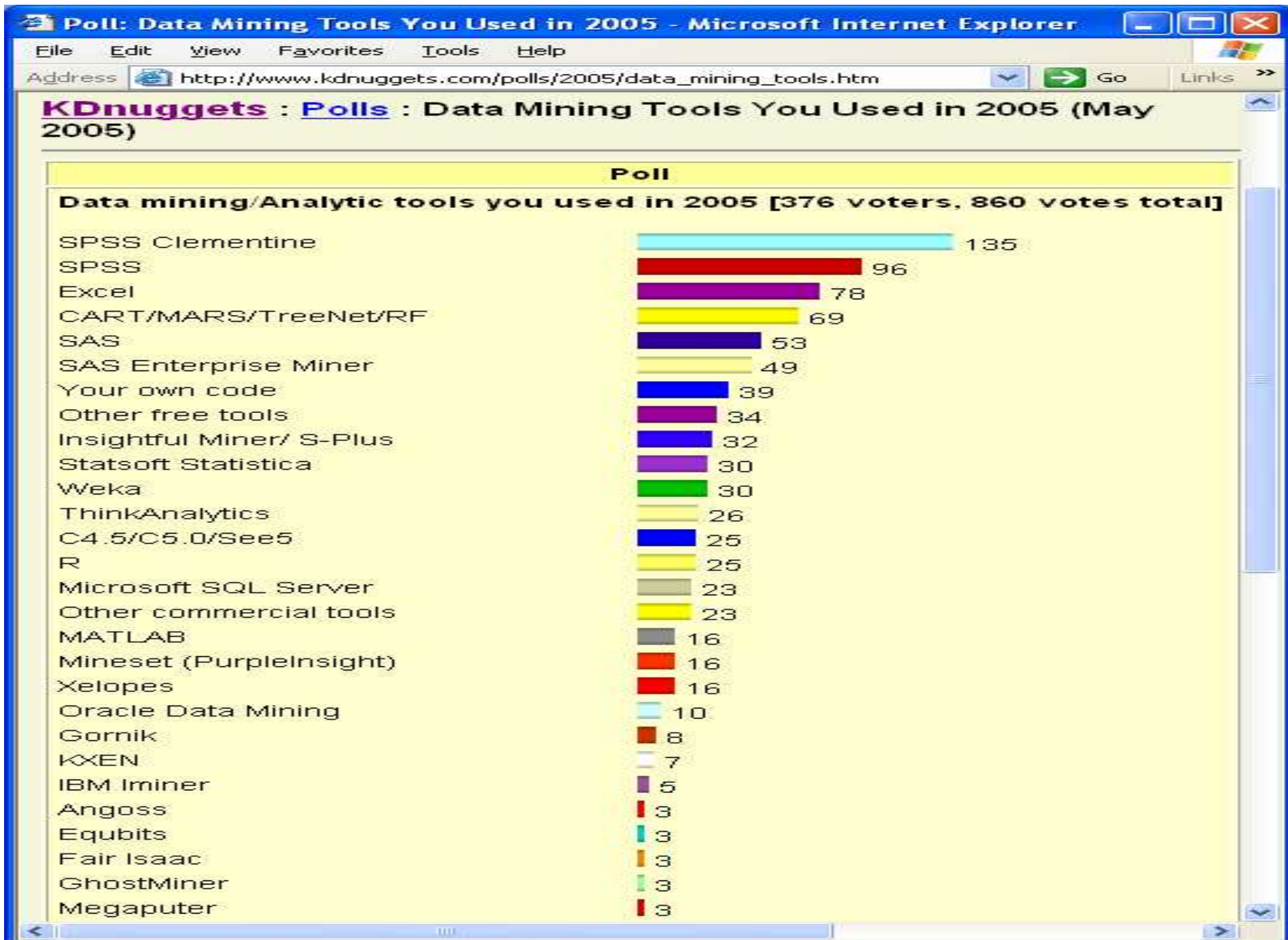
# Data Mining

- **"Drowning in Data yet Starving for Knowledge"**  
???
- **"Computers have promised us a fountain of wisdom but delivered a flood of data"**  
*William J. Frawley, Gregory Piatetsky-Shapiro, and Christopher J. Matheus*
- ***Data Mining: "The non trivial extraction of implicit, previously unknown, and potentially useful information from data"***  
*William J Frawley, Gregory Piatetsky-Shapiro and Christopher J Matheus*
- **Data mining finds valuable information hidden in large volumes of data.**
- **Data mining is the analysis of data and the use of software techniques for finding patterns and regularities in sets of data.**
- **Data Mining is an interdisciplinary field involving:**
  - Databases
  - Statistics
  - Machine Learning
  - High Performance Computing
  - Visualization
  - Mathematics

# Data Mining Software

**KDnuggets : Polls : Data Mining Tools You Used in 2005 (May 2005) PollData mining/Analytic tools you used in 2005 [376 voters, 860 votes total]**

- **Enterprise-level: (US \$10,000 and more)**  
Fair Isaac, IBM, Insightful, KXEN, Oracle, SAS, and SPSS
- **Department-level: (from \$1,000 to \$9,999)**  
Angoss, CART/MARS/TreeNet/Random Forests, Equbits, GhostMiner, Gornik, Mineset, MATLAB, Megaputer, Microsoft SQL Server, Statsoft Statistica, ThinkAnalytics
- **Personal-level: (from \$1 to \$999):** Excel, See5
- **Free:** C4.5, R, Weka, Xelopes



# Weka Data Mining Software

KDnuggets : News : 2005 : n13 : item2

**SIGKDD Service Award** is the highest service award in the field of data mining and knowledge discovery. It is given to one individual or one group who has performed significant service to the data mining and knowledge discovery field, including professional volunteer services in disseminating technical information to the field, education, and research funding.

The **2005 ACM SIGKDD Service Award** is presented to **the Weka team** for their development of the freely-available Weka Data Mining Software, including the accompanying book *Data Mining: Practical Machine Learning Tools and Techniques* (now in second edition) and much other documentation.

The Weka team includes **Ian H. Witten** and **Eibe Frank**, and the following major contributors (in alphabetical order of last names): Remco R. Bouckaert, John G. Cleary, Sally Jo Cunningham, Andrew Donkin, Dale Fletcher, Steve Garner, Mark A. Hall, Geoffrey Holmes, Matt Humphrey, Lyn Hunt, Stuart Inglis, Ashraf M. Kibriya, Richard Kirkby, Brent Martin, Bob McQueen, Craig G. Nevill-Manning, Bernhard Pfahringer, Peter Reutemann, Gabi Schmidberger, Lloyd A. Smith, Tony C. Smith, Kai Ming Ting, Leonard E. Trigg, Yong Wang, Malcolm Ware, and Xin Xu.

The Weka team has put a tremendous amount of effort into continuously developing and maintaining the system **since 1994**. The development of Weka was funded by a grant from the New Zealand Government's Foundation for Research, Science and Technology.

The **key features** responsible for Weka's success are:

- it provides many different algorithms for data mining and machine learning
- it is open source and freely available
- it is platform-independent
- it is easily useable by people who are not data mining specialists
- it provides flexible facilities for scripting experiments
- it has kept up-to-date, with new algorithms being added as they appear in the research literature.

# Weka Data Mining Software

## KDnuggets : News : 2005 : n13 : item2 (cont.)

The Weka Data Mining Software has been downloaded **200,000 times** since it was put on SourceForge in April 2000, and is currently downloaded at a rate of 10,000/month. The Weka mailing list has over **1100 subscribers in 50 countries**, including subscribers from many major companies.

There are **15 well-documented substantial projects** that incorporate, wrap or extend Weka, and no doubt many more that have not been reported on Sourceforge.

Ian H. Witten and Eibe Frank also wrote a **very popular book "Data Mining: Practical Machine Learning Tools and Techniques"** (now in the second edition), that seamlessly integrates Weka system into teaching of data mining and machine learning. In addition, they provided **excellent teaching material** on the book website.

This book became one of the most popular textbooks for data mining and machine learning, and is **very frequently cited in scientific publications**.

Weka is a **landmark system in the history of the data mining and machine learning** research communities, because it is the only toolkit that has gained such widespread adoption and survived for an extended period of time (the first version of Weka was released 11 years ago). Other data mining and machine learning systems that have achieved this are individual systems, such as C4.5, not toolkits.


Since Weka is freely available for download and offers many powerful features (sometimes not found in commercial data mining software), it has become one of the most widely used data mining systems. Weka also became one of the favorite vehicles for data mining research and helped to advance it by making many powerful features available to all.

**In sum, the Weka team has made an outstanding contribution to the data mining field.**

Machine Learning Project - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://www.cs.waikato.ac.nz/~ml/index.html> Go Links >>



**Project**

---

[project](#) • [software](#) • [book](#) • [publications](#) • [people](#) • [related](#)

---

Interested in doing an MSc or PhD in Machine Learning here at Waikato and spending some time overseas while working on your project? Then check [this](#) out.

---

## Weka Machine Learning Project

An exciting and potentially far-reaching development in computer science is the invention and application of methods of machine learning. These enable a computer program to automatically analyse a large body of data and decide what information is most relevant. This crystallised information can then be used to automatically make predictions or to help people make decisions faster and more accurately.


The overall goal of our project is to build a state-of-the-art facility for developing machine learning (ML) techniques and to apply them to real-world data mining problems. Our team has incorporated several standard ML techniques into a software "workbench" called WEKA, for Waikato Environment for Knowledge Analysis. With it, a specialist in a particular field is able to use ML to derive useful knowledge from databases that are far too large to be analysed by hand. WEKA's users are ML researchers and industrial scientists, but it is also widely used for teaching.

Our objectives are to

- make ML techniques generally available;
- apply them to practical problems that matter to New Zealand industry;
- develop new machine learning algorithms and give them to the world;
- contribute to a theoretical framework for the field.

Our machine learning package is publically available and presents a collection of algorithms for solving real-world data mining problems. The software is written entirely in Java and includes a uniform interface to a number of standard ML techniques. Please feel free to browse around.

Found only on the islands of New Zealand, the weka is a flightless bird with an inquistive nature. (How should you pronounce [WEKA](#)? What does the weka [sound](#) like.)





## Software

[project](#) • [software](#) • [book](#) • [publications](#) • [people](#) • [related](#)

### Home

#### Getting started

- [Requirements](#)
- [Download](#)
- [Documentation](#)
- [FAQ](#)
- [Citing Weka](#)

#### Further information

- [Datasets](#)
- [Related Projects](#)
- [Miscellaneous Code](#)
- [Other Literature](#)

#### Developers

- [Development](#)
- [History](#)
- [CVS](#)
- [Contributors](#)

## Weka 3: Data Mining Software in Java

Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

Weka is open source software issued under the [GNU General Public License](#).

### Pentaho's live forum for Weka

The open-source BI software company Pentaho has become major sponsor of Weka development and will take over the administration of Weka's Sourceforge site in the near future. Pentaho also provides a live [forum](#) for interaction among Weka project community members.

### The Weka mailing list

Please post Weka-related questions, comments, and bug reports to the [Weka mailing list](#) (don't forget to check out the [online documentation](#) first, before posting to the list). There is also the searchable mailing list [archive](#) (Mirrors: [news.gmane.org](#), [Nabble](#)). Please do not email individual members of our research group about Weka problems.

Also, please have in mind that your message will be sent to several thousand people, so please post according to the [Mailing List Etiquette](#). The administrator also removes members from the mailing list in case their mailboxes run full, since they apparently don't read their emails anymore.



Search

- [Wekalist archives](#)  [Wekalist at news.gmane.org](#)  [Wekalist at Nabble](#)  [Weka forum at Penthao](#)





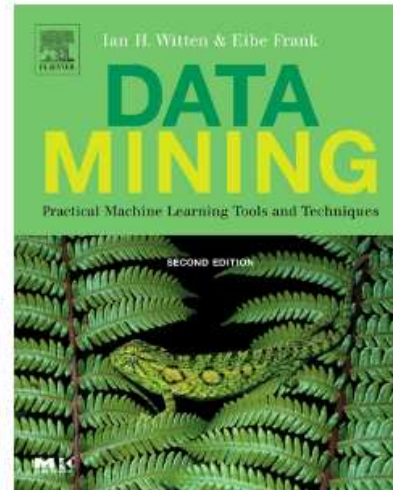
## Book

[project](#) • [software](#) • [book](#) • [publications](#) • [people](#) • [related](#)

### Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)

[Ian H. Witten](#), [Eibe Frank](#)

Morgan  
Kaufmann  
June 2005  
525 pages  
Paper  
ISBN 0-12-  
088407-0



[Eibe Frank and Ian Witten](#)

[Click here to order from Amazon.com](#)

#### Comments

"If you have data that you want to analyze and understand, this book and the associated Weka toolkit are an excellent way to start."

-Jim Gray, Microsoft Research

# Using Weka to teach Machine Learning, Data and Web Mining

<http://uhaweb.hartford.edu/compsci/ccli/>

Project MLExAI: Sample Projects - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://uhaweb.hartford.edu/compsci/ccli/samplep.htm> Go Links

## Machine Learning Experiences in Artificial Intelligence: A Multi-Institutional Project

NSF DUE-0716338

### Resources

- Home
- Overview
- Machine Learning
- ML Resources
- Publications
- Sample Projects
- Grants
- News
- Testimonials
- Contact

### People

- Faculty PIs
- Advisory Board
- Senior Personnel
- Students

### Institutions

- University of Hartford
- CCSU

### Sample Projects

\*Projects are posted as they become available

#### Web User Profiling

Web searches provide large amounts of information about web users. Data mining techniques can be used to analyze this information and create web user profiles. A key application of this approach is in marketing and offering personalized services, an area referred to as "data gold rush". The aim of this project is to develop a system that can be used to develop an intelligent web browser. This project focuses on the use of Decision Tree learning to create models of web users.

#### Character Recognition and Learning with Neural Networks

The power and usefulness of artificial neural networks have been demonstrated in several applications including speech synthesis, diagnostic problems, medicine, business and finance, robotic control, signal processing, computer vision and many other problems that fall under the category of pattern recognition. The goal of this project is to develop a character recognition system based on a neural network model.

#### Solving the N-Puzzle Problem

The N-puzzle game provides a good framework for illustrating conceptual AI search in an interesting and motivating way. The objective of this project is to introduce the student to Analytical (Explanation-Based) Learning using the classical AI framework of search. Hands-on experiments with search algorithms combined with an Explanation Based Learning (EBL) component give students a deep, experiential understanding of the basics of EBL.

#### Solving the Dice Game Pig

The jeopardy dice game Pig is very simple to describe, yet the optimal policy for play is far from trivial and was only recently solved. Using the computation of the optimal solution as a central challenge problem, we give the student a deep, experiential understanding of dynamic programming and value iteration through explanation, implementation examples, and implementation exercises.

#### Web Document Classification

Along with search engines, topic directories are the most popular sites on the Web. Topic directories organize web pages in a hierarchical structure according to their content. The aim of the project is to investigate the process of tagging web pages using the topic directory structures and apply Machine Learning techniques for automatic tagging. This would help in filtering out the responses of a search engine or ranking them according to their relevance to a topic.

Internet

*Machine Learning, Data and Web Mining*  
by Example  
(“learning by doing” approach)

- Data preprocessing and visualization
- Attribute selection
- Classification (OneR, Decision trees)
- Prediction (Nearest neighbor)
- Model evaluation
- Clustering (K-means, Cobweb)
- Association rules

# Data preprocessing and visualization

## Initial Data Preparation (Weka data input)

- Raw data (Japanese loan data)
- Web/Text documents (Department data)

# Data preprocessing and visualization

Japanese loan data (a sample from a loan history database of a Japanese bank)

Clients: s1,..., s20

- Approved loan: s1, s2, s4, s5, s6, s7, s8, s9, s14, s15, s17, s18, s19
- Rejected loan: s3, s10, s11, s12, s13, s16, s20

Clients data:

- unemployed clients: s3, s10, s12
- loan is to buy a personal computer: s1, s2, s3, s4, s5, s6, s7, s8, s9, s10
- loan is to buy a car: s11, s12, s13, s14, s15, s16, s17, s18, s19, s20
- male clients: s6, s7, s8, s9, s10, s16, s17, s18, s19, s20
- not married: s1, s2, s5, s6, s7, s11, s13, s14, s16, s18
- live in problematic area: s3, s5
- age: s1=18, s2=20, s3=25, s4=40, s5=50, s6=18, s7=22, s8=28, s9=40, s10=50, s11=18, s12=20, s13=25, s14=38, s15=50, s16=19, s17=21, s18=25, s19=38, s20=50
- money in a bank (x10000 yen): s1=20, s2=10, s3=5, s4=5, s5=5, s6=10, s7=10, s8=15, s9=20, s10=5, s11=50, s12=50, s13=50, s14=150, s15=50, s16=50, s17=150, s18=150, s19=100, s20=50
- monthly pay (x10000 yen): s1=2, s2=2, s3=4, s4=7, s5=4, s6=5, s7=3, s8=4, s9=2, s10=4, s11=8, s12=10, s13=5, s14=10, s15=15, s16=7, s17=3, s18=10, s19=10, s20=10
- months for the loan: s1=15, s2=20, s3=12, s4=12, s5=12, s6=8, s7=8, s8=10, s9=20, s10=12, s11=20, s12=20, s13=20, s14=20, s15=20, s16=20, s17=20, s18=20, s19=20, s20=30
- years with the last employer: s1=1, s2=2, s3=0, s4=2, s5=25, s6=1, s7=4, s8=5, s9=15, s10=0, s11=1, s12=2, s13=5, s14=15, s15=8, s16=2, s17=3, s18=2, s19=15, s20=2



# Data preprocessing and visualization

Attribute-Relation File Format (ARFF) - <http://www.cs.waikato.ac.nz/~ml/weka/arff.html>

The screenshot shows a Microsoft Internet Explorer browser window displaying the "Attribute-Relation File Format (ARFF)" page. The page title is "Attribute-Relation File Format (ARFF)" and the address bar shows "http://www.cs.waikato.ac.nz/~ml/weka/arff.html". The page content includes a date "April 4th, 2006", a note that the documentation is superseded by the WekaDoc Wiki, and a list of links for versions 3.4.x and 3.5.x. Below this, there is a date "April 1st, 2002" and a paragraph explaining that an ARFF file is an ASCII text file describing a list of instances sharing a set of attributes. It mentions that the document describes the version of ARFF used with Weka versions 3.2 to 3.3, which is an extension of the ARFF format as described in the document written by Ian H. Witten and Eibe Frank. Another paragraph explains that this explanation was cobbled together by Gordon Paynter and Eibe Frank, and has been edited by Richard Kirkby. The "Overview" section states that ARFF files have two distinct sections: the Header and the Data. The Header contains the name of the relation, a list of attributes, and their types. An example of a standard IRIS dataset is provided, showing the title, sources, and the ARFF header and data sections.

The Notepad window, titled "LoanData - Notepad", displays the following ARFF file content:

```
@relation LoanData

@attribute ID numeric
@attribute sex {f,m}
@attribute married {n,y}
@attribute age numeric
@attribute money numeric
@attribute pay numeric
@attribute months numeric
@attribute buy {pc,car}
@attribute emp {y,n}
@attribute lastemp numeric
@attribute area {good,bad}
@attribute approved {y,n}

@data

1, f, n, 18, 20, 2, 15, pc, y, 1, good, y
2, f, n, 20, 10, 2, 20, pc, y, 2, good, y
3, f, y, 25, 5, 4, 12, pc, n, 0, bad, n
4, f, y, 40, 5, 7, 12, pc, y, 2, good, y
5, f, n, 50, 5, 4, 12, pc, y, 25, bad, y
6, m, n, 18, 10, 5, 8, pc, y, 1, good, y
7, m, n, 22, 10, 3, 8, pc, y, 4, good, y
8, m, y, 28, 15, 4, 10, pc, y, 5, good, y
9, m, y, 40, 20, 2, 20, pc, y, 15, good, y
10, m, y, 50, 5, 4, 12, pc, n, 0, good, n
11, f, n, 18, 50, 8, 20, car, y, 1, good, n
12, f, y, 20, 50, 10, 20, car, n, 2, good, n
13, f, n, 25, 50, 5, 20, car, y, 5, good, n
14, f, n, 38, 150, 10, 20, car, y, 15, good, y
15, f, y, 50, 50, 15, 20, car, y, 8, good, y
16, m, n, 19, 50, 7, 20, car, y, 2, good, n
17, m, y, 21, 150, 3, 20, car, y, 3, good, y
18, m, n, 25, 150, 10, 20, car, y, 2, good, y
19, m, y, 38, 100, 10, 20, car, y, 15, good, y
20, m, y, 50, 50, 10, 30, car, y, 2, good, n
```

# Data preprocessing and visualization


Download and install Weka - <http://www.cs.waikato.ac.nz/~ml/weka/>

Weka 3 - Data Mining with Open Source Machine Learning Software in Java - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites

Address <http://www.cs.waikato.ac.nz/ml/weka/> Go Links



**WEKA**  
The University  
of Waikato

**Software**

[project](#) - [software](#) - [book](#) - [publications](#) - [people](#) - [related](#)

**Home**

**Getting started**  
[Requirements](#)  
[Download](#)  
[Documentation](#)  
[FAQ](#)  
[Citing Weka](#)

**Further information**  
[Datasets](#)  
[Related Projects](#)  
[Miscellaneous Code](#)  
[Other Literature](#)

**Developers**  
[Development](#)  
[History](#)  
[CVS](#)  
[Contributors](#)

**Downloading and installing Weka**

- **Snapshots**

Every night a snapshot of the CVS is taken, compiled and put together in ZIP files. For those who want to have the latest bugfixes, they can download these snapshots [here](#).
- **Book version**

Weka 3.4 is the latest stable version of Weka, and the one described in the [data mining book](#). There are different options for downloading and installing it on your system:

- **Windows**

Click [here](#) to download a self-extracting executable that includes Java VM 1.4 (weka-3-4-12jre.exe; 24,445,809 bytes)

Click [here](#) to download a self-extracting executable without the Java VM (weka-3-4-12.exe; 10,330,491 bytes)

These executables will install Weka in your Program Menu. Download the second version if you already have Java 1.4 (or later) on your system.
- **Mac OS X**

Click [here](#) to download a disk image for OS X (weka-3-4-12.dmg; 13,565,484 bytes)
- **Other platforms (Linux, etc.)**

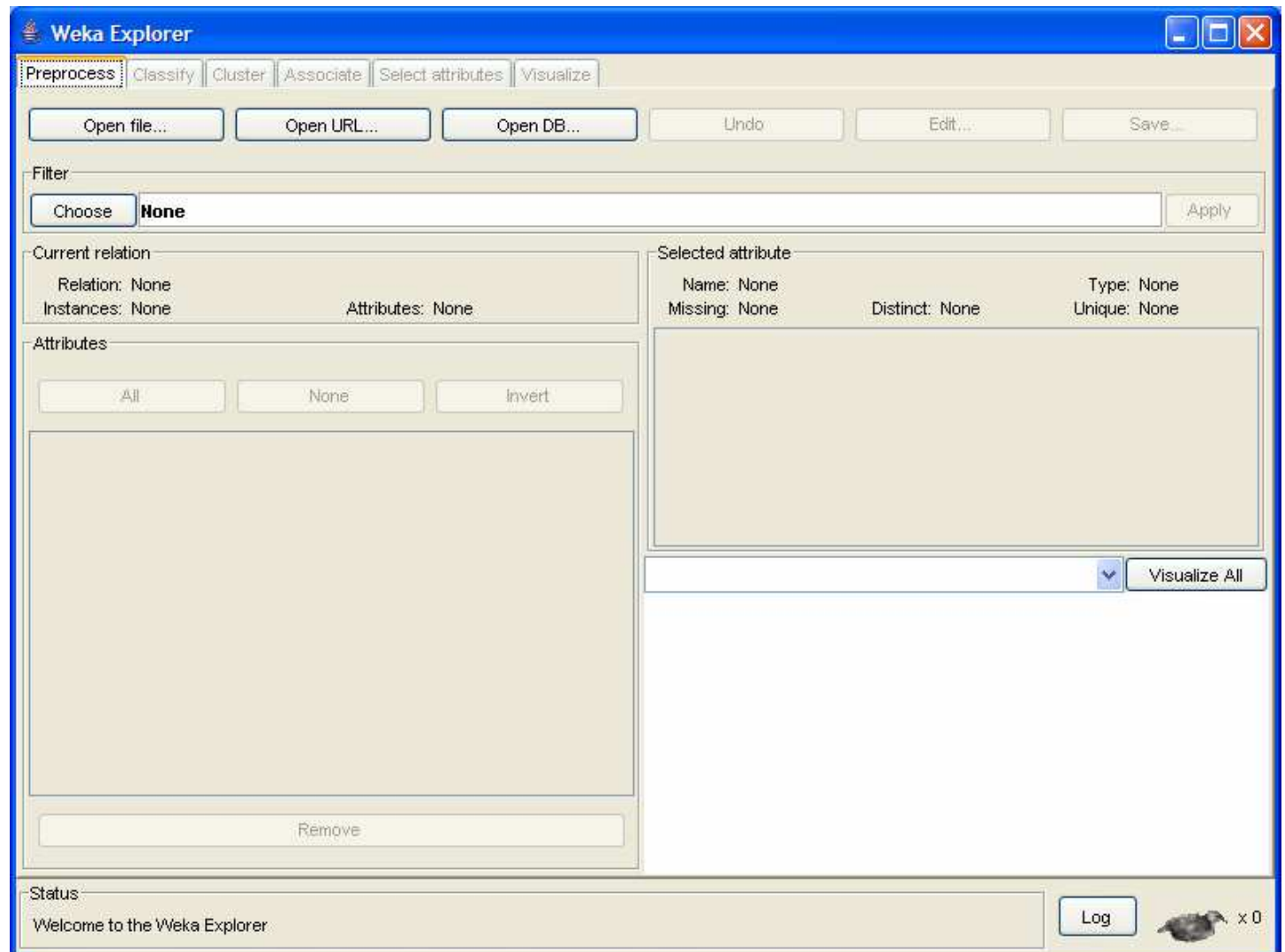
Click [here](#) to download a zip archive containing Weka (weka-3-4-12.zip; 10,421,962 bytes)

First unzip the zip file. This will create a new directory called weka-3-4-12. To run Weka, change into that directory and type



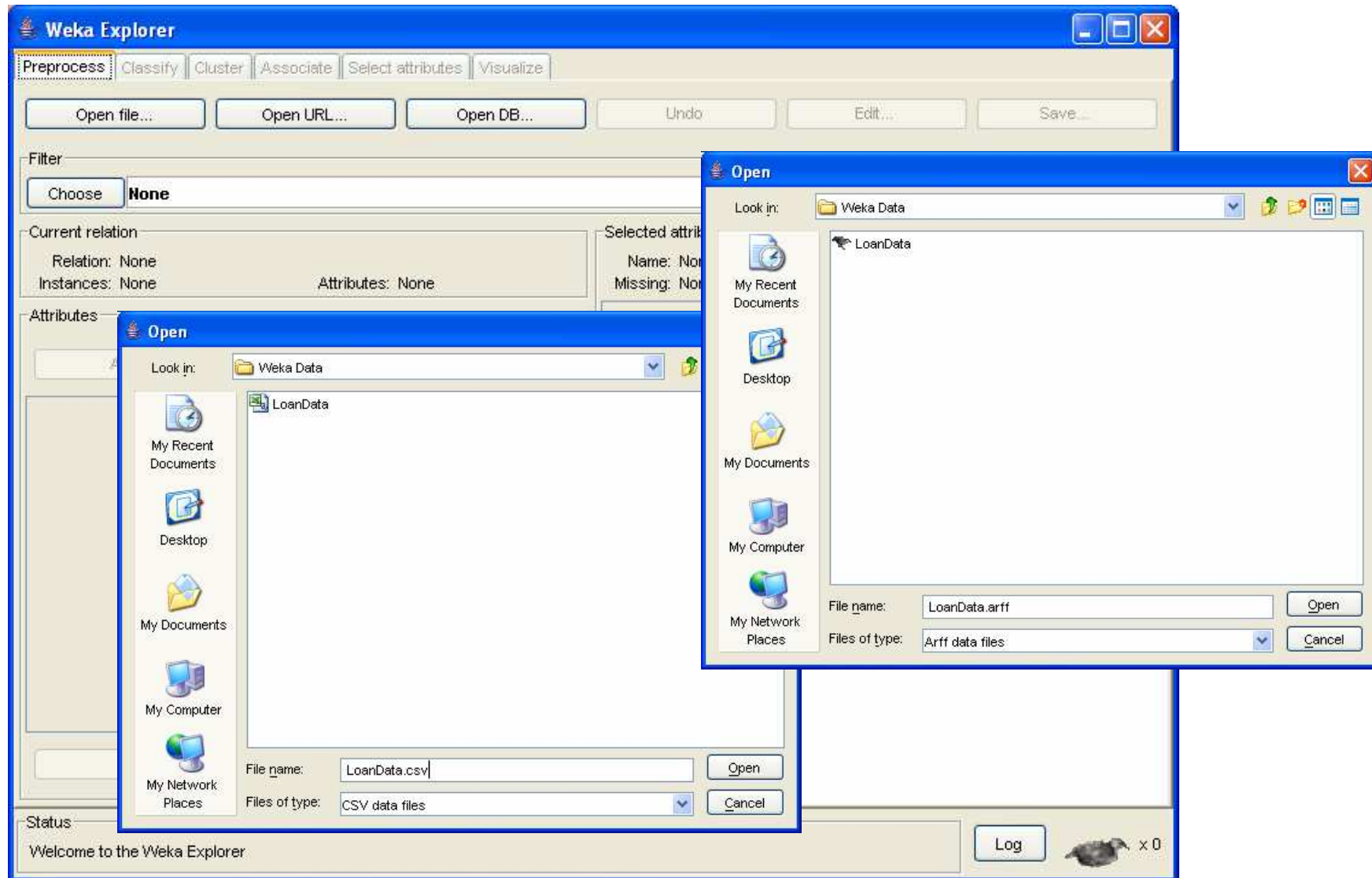
# Data preprocessing and visualization

Run Weka and select the Explorer



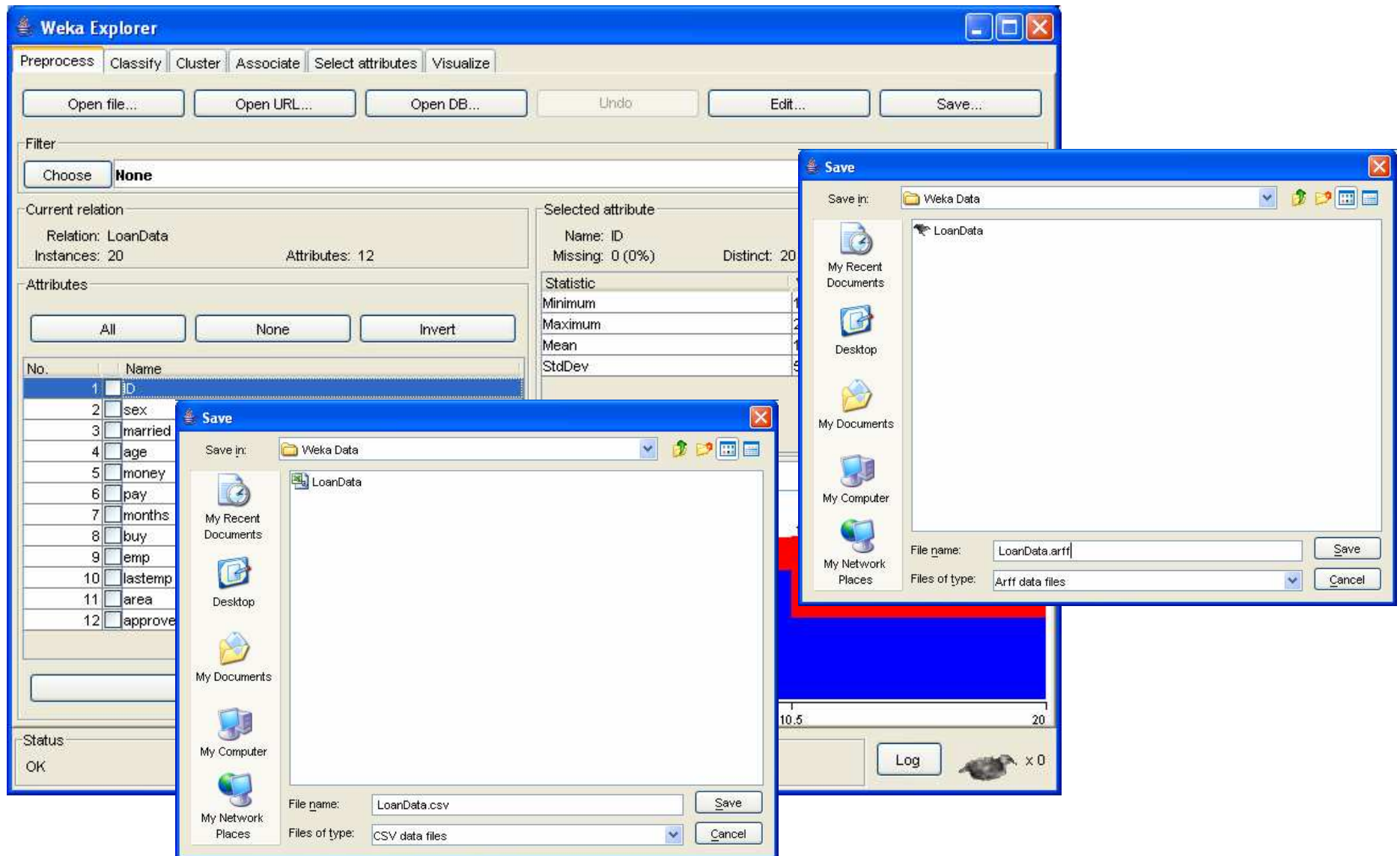
# Data preprocessing and visualization

Load data into Weka – ARFF format or CVS format (click on “Open file...”)



# Data preprocessing and visualization

Converting data formats through Weka (click on “Save...”)



# Data preprocessing and visualization

Editing data in Weka (click on "Edit...")

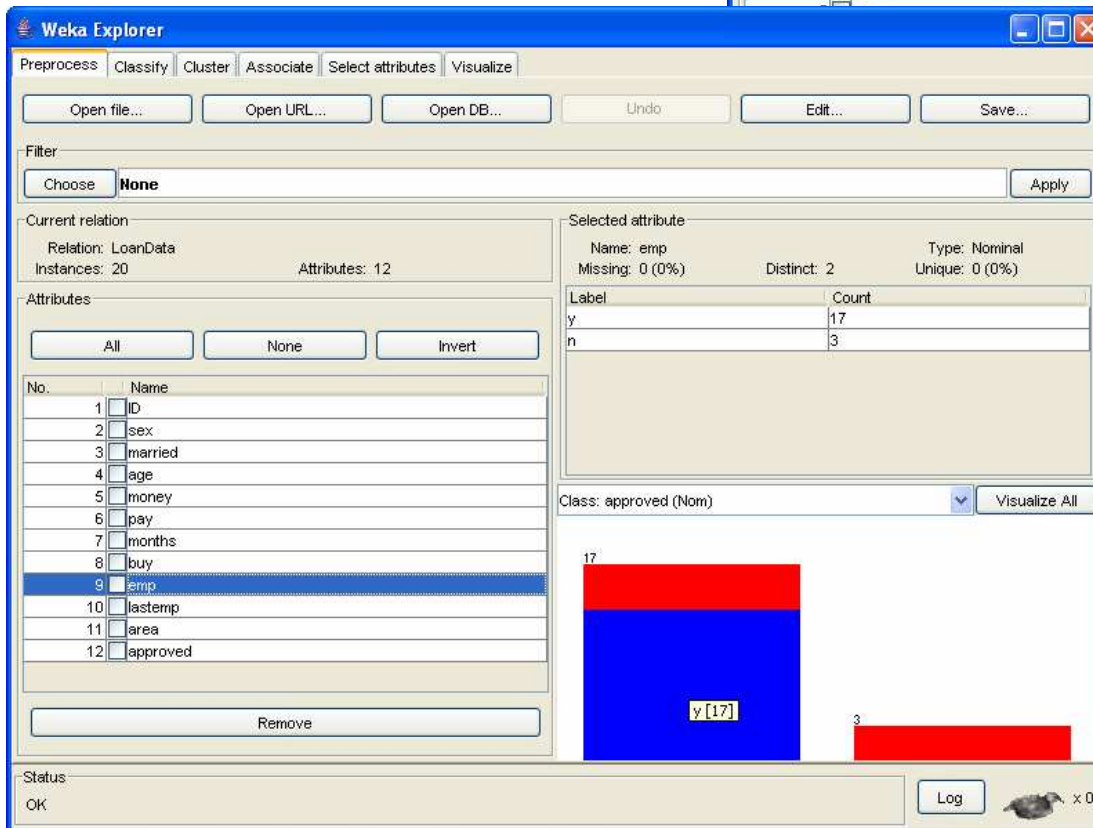
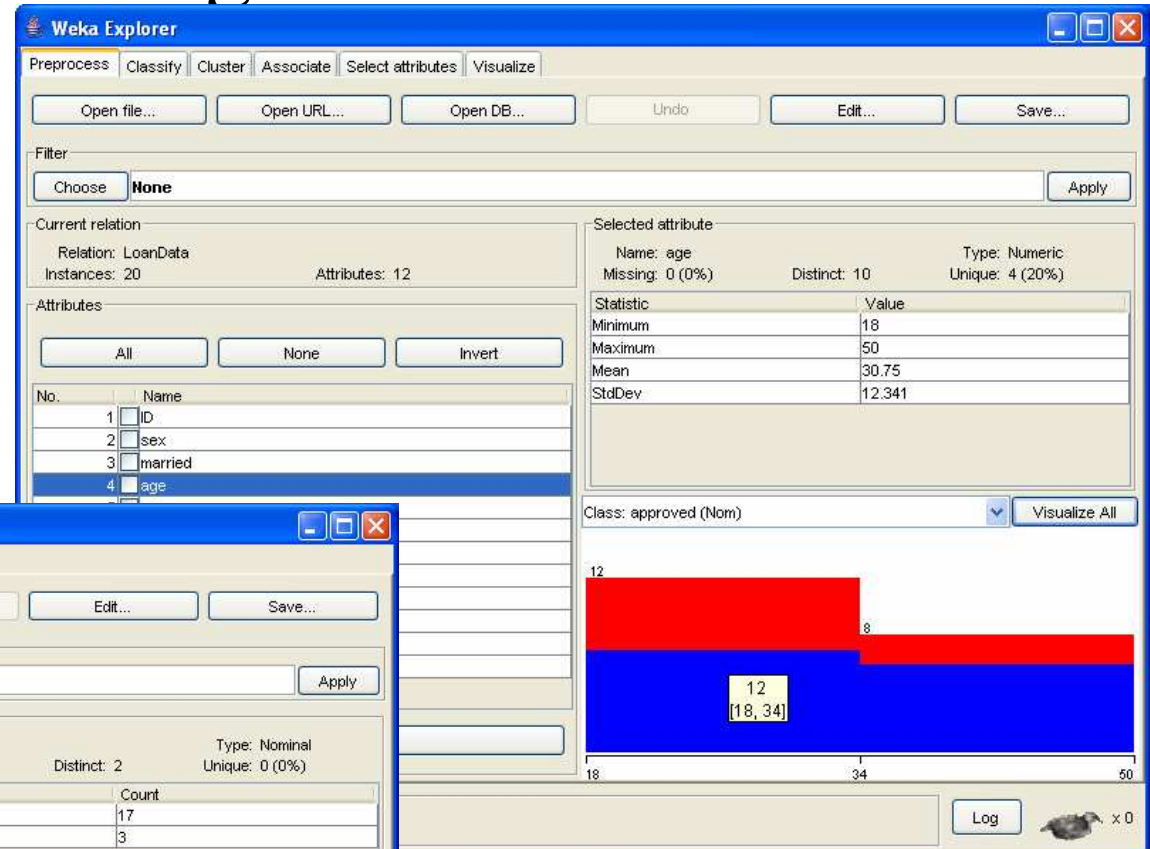
The image shows two overlapping windows from the Weka software. The background window is 'Weka Explorer', which has tabs for 'Preprocess', 'Classify', 'Cluster', 'Associate', 'Select attributes', and 'Visualize'. The 'Edit...' button is highlighted. The foreground window is 'Viewer', which displays a table of data for the 'LoanData' relation. The table has 13 columns: 'No.', 'ID', 'sex', 'married', 'age', 'money', 'pay', 'months', 'buy', 'emp', 'lastemp', 'area', and 'approved'. The data is organized into rows, with the first 17 rows visible. The 'approved' column contains values 'y' and 'n'. The 'Viewer' window has 'Undo', 'OK', and 'Cancel' buttons at the bottom.

No.	ID	sex	married	age	money	pay	months	buy	emp	lastemp	area	approved
	Numeric	Nominal	Nominal	Numeric	Numeric	Numeric	Numeric	Nominal	Nominal	Numeric	Nominal	Nominal
1	1.0	f	n	18.0	20.0	2.0	15.0	pc	y	1.0	good	y
2	2.0	f	n	20.0	10.0	2.0	20.0	pc	y	2.0	good	y
3	3.0	f	y	25.0	5.0	4.0	12.0	pc	n	0.0	bad	n
4	4.0	f	y	40.0	5.0	7.0	12.0	pc	y	2.0	good	y
5	5.0	f	n	50.0	5.0	4.0	12.0	pc	y	25.0	bad	y
6	6.0	m	n	18.0	10.0	5.0	8.0	pc	y	1.0	good	y
7	7.0	m	n	22.0	10.0	3.0	8.0	pc	y	4.0	good	y
8	8.0	m	y	28.0	15.0	4.0	10.0	pc	y	5.0	good	y
9	9.0	m	y	40.0	20.0	2.0	20.0	pc	y	15.0	good	y
10	10.0	m	y	50.0	5.0	4.0	12.0	pc	n	0.0	good	n
11	11.0	f	n	18.0	50.0	8.0	20.0	car	y	1.0	good	n
12	12.0	f	y	20.0	50.0	10.0	20.0	car	n	2.0	good	n
13	13.0	f	n	25.0	50.0	5.0	20.0	car	y	5.0	good	n
14	14.0	f	n	38.0	150.0	10.0	20.0	car	y	15.0	good	y
15	15.0	f	y	50.0	50.0	15.0	20.0	car	y	8.0	good	y
16	16.0	m	n	19.0	50.0	7.0	20.0	car	y	2.0	good	n
17	17.0	m	y	21.0	150.0	3.0	20.0	car	y	3.0	good	y

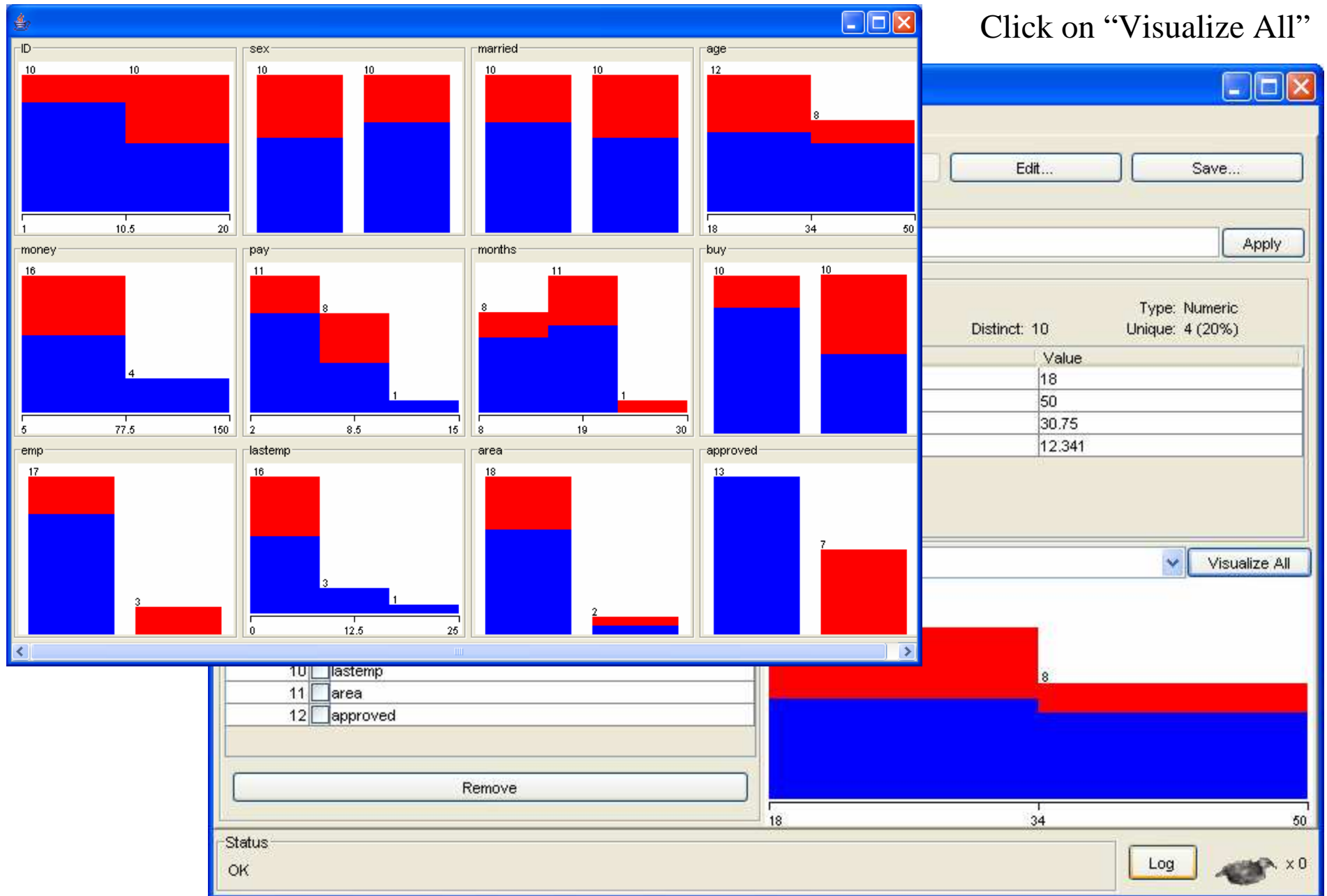
# Data preprocessing and visualization

## Examining data

- Attribute type and properties
- Class (last attribute) distribution



# Data preprocessing and visualization



# Data preprocessing and visualization

Web/Text documents - Department data

<http://www.cs.ccsu.edu/~markov/>

- Download Ch1, DMW Book
- Download datasets

School of Arts & Sciences Departments - Microsoft Internet Explorer

Address: <http://www.artsci.ccsu.edu/Departments.htm>

Central Connecticut State University

## Departments

[Department Chairs, Locations, Phone Numbers](#)

<a href="#">Anthropology</a>	<a href="#">History</a>
<a href="#">Art</a>	<a href="#">Mathematical Sciences</a>
<a href="#">Biological Sciences</a>	<a href="#">Modern Language</a>
<a href="#">Chemistry</a>	<a href="#">Music</a>
<a href="#">Communication</a>	<a href="#">Philosophy</a>
<a href="#">Computer Science</a>	<a href="#">Physics/Earth Sciences</a>
<a href="#">Criminal Justice</a>	<a href="#">Political Science</a>
<a href="#">Design</a>	<a href="#">Psychology</a>
<a href="#">Economics</a>	<a href="#">Sociology</a>
<a href="#">English</a>	<a href="#">Theatre</a>
<a href="#">Geography</a>	

[ [A&S Home](#) ] [ [A-Z Directory](#) ] [ [Departments](#) ] [ [About](#) ]

page last updated: 10/27/04  
Comments, suggestions: [aswebmaster@ccsu.edu](mailto:aswebmaster@ccsu.edu)

Music - Microsoft Internet Explorer

Address: <http://www.artsci.ccsu.edu/Departments/Music.html>

## The School of Arts and Sciences

Central Connecticut State University

### Music

Students majoring in music may pursue either a BS in Music education degree, the professional degree that certifies them to teach music in the public schools, or a BA in music, with specializations in either performance, music history, theory/composition, or jazz studies. Full-time and associate faculty are active in the United States and abroad performing, conducting, and presenting scholarly papers. The department's computer lab is equipped with MIDI keyboards and the industry's leading music software. The Music Department is the New England center for Orff Schulwerk training and the host for Connecticut's middle school/high school music festival and the Summer Music Institute, a national in-service program for music educators.

PROGRAMS OF STUDY: BS, BA, MS

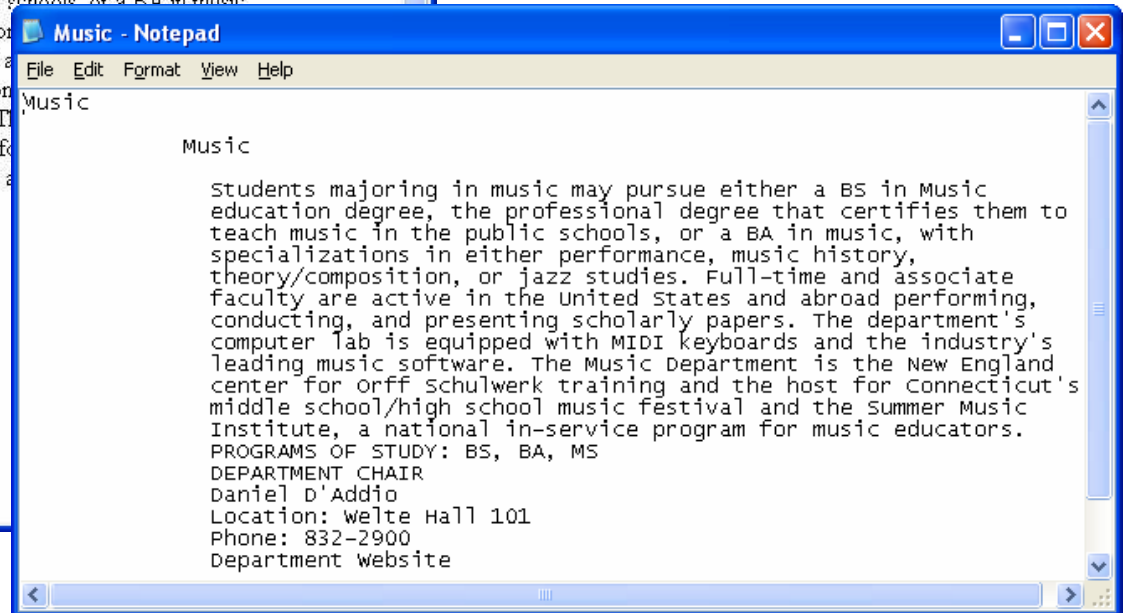
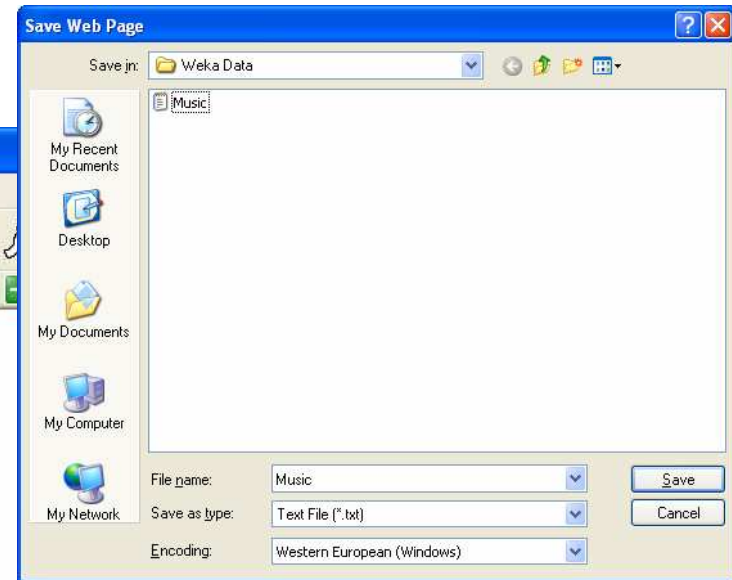
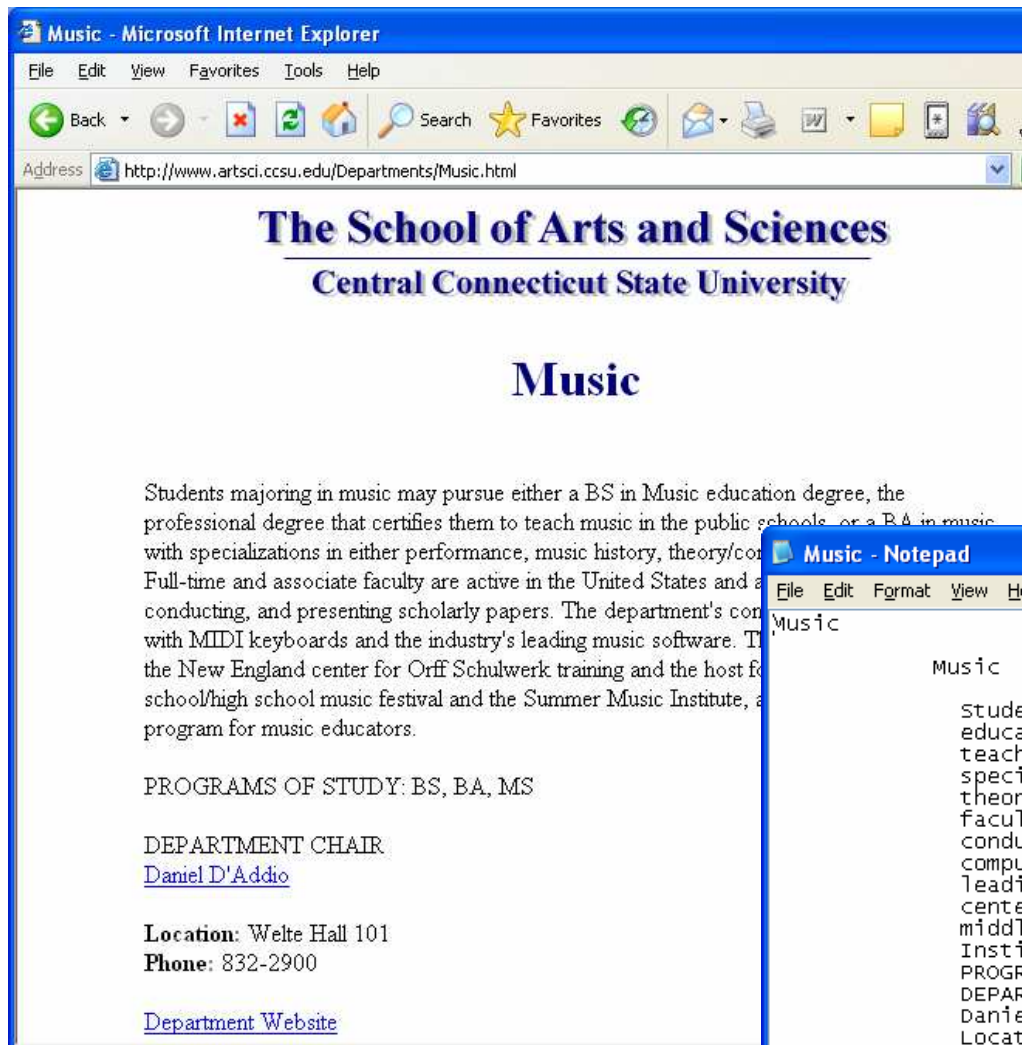
DEPARTMENT CHAIR  
[Daniel D'Addio](#)

**Location:** Welte Hall 101  
**Phone:** 832-2900

[Department Website](#)

# Data preprocessing and visualization

## Convert HTML to Text





# Data preprocessing and visualization

```
Departments-string - Notepad
File Edit Format View Help
@relation departments_string

@attribute document_name string
@attribute document_content string
@attribute document_class {A,B}

@data
Anthropology, "anthropology anthropology anthropology consists of four subfie
Art, "art art the art department s undergraduate degree program offers a wide
ent website", B
Biology, "biology biological sciences the undergraduate and graduate degrees
Chemistry, "chemistry chemistry the chemistry department has been approved by
Communication, "communication communication balancing theoretical practical a
Computer, "computer science computer science students majoring in computer sc
Justice, "criminal justice criminal justice although criminal justice is prin
Economics, "economics economics the bachelor s degree in economics is struct
English, "english english the english department offers courses and programs
Geography, "geography geography concerned with the science of location the ge
History, "history history given the diverse expertise of its faculty the depa
Math, "mathematical sciences mathematical sciences the department of mathemat
Languages, "modern languages modern languages the modern language department
Music, "music music students majoring in music may pursue either a bs in mus
Philosophy, "philosophy philosophy the department of philosophy offers underc
Physics, "physics earth sciences physics earth sciences the physics and eartf
Political, "political science political science the ba in political science
Psychology, "psychology psychology the psychology department offers courses
Sociology, "sociology sociology the sociology department offers degree progr
Theatre, "theatre theatre both the ba and bfa degrees with specialization in
```

## Loading text data in Weka

- String format for ID and content
- One document per line
- Add class (nominal) if needed

The screenshot shows the Weka GUI with the 'Visualize' tab selected. The 'Selected attribute' section shows 'document\_class' with a Type of 'Nominal'. Below this, a table displays the distribution of the attribute:

Label	Count
A	11
B	9

The visualization area shows two bars: a blue bar for class 'A' with a count of 11, and a red bar for class 'B' with a count of 9. The status bar at the bottom indicates 'OK'.

# Data preprocessing and visualization

## Converting a string attribute into nominal

Choose filters/unsupervised/attribute/StringToNominal and set the index to 1

The screenshot shows the Weka Explorer interface with the StringToNominal filter selected. The filter's configuration dialog is open, showing the attribute index set to 1. The dialog also displays the attribute's name, type, and statistics.

**Weka Explorer**

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... | Open URL... | Open DB... | Undo | Edit... | Save...

Filter: Choose **StringToNominal** -C 1 Apply

Current relation  
Relation: departments\_string  
Instances: 20      Attributes: 3

Attributes

All | None | Invert

No.	Name
1	<input checked="" type="checkbox"/> document_name
2	<input type="checkbox"/> document_content
3	<input type="checkbox"/> document_class

Remove

Status: OK

Log x 0

**weka.gui.GenericObjectEditor**

weka.filters.unsupervised.attribute.StringToNominal

About

Converts a string attribute (i. [More](#))

attributeIndex: 1

Open... | Save... | OK | Cancel

Attribute is neither numeric nor nominal.



# Data preprocessing and visualization

Converting text data into TFIDF (Term Frequency – Inverted Document Frequency) attribute format

- Choose filters/unsupervised/attribute/StringToWordVector
- Set the parameters as needed (see “More”)
- Click on “Apply”

The screenshot displays the Weka Explorer interface with the 'StringToWordVector' filter selected. The 'More' dialog is open, showing the filter's configuration options. The 'Information' dialog is also open, providing details about the filter's name, synopsis, and options.

**Weka Explorer - Filter Configuration:**

Filter: **StringToWordVector** -D ".,:>()?!" -W 1000 -L -A -S

Current relation: departments\_string-weka.filters.unsupervised.attribute.String...  
Instances: 20      Attributes: 612

Selected attribute: academic  
Missing: 0 (0%)      Distinct: 2

Statistic	Value
Minimum	0
Maximum	1
Mean	0.2
StdDev	0.41

Class: document\_class (Nom)

Bar chart showing the distribution of the 'academic' attribute:

Value	Count
0	16
1	4

**weka.gui.GenericObjectEditor - About:**

Converts String attributes into a set of attributes representing word occurrence information from the text contained in the strings.

Parameters:

- IDFTransform: False
- TFTransform: False
- attributeNamePrefix: (empty)
- delimiters: .:>()?!
- lowerCaseTokens: True
- normalizeDocLength: False
- onlyAlphabeticTokens: True
- outputWordCounts: False
- useStoplist: True

**Information - NAME:** weka.filters.unsupervised.attribute.StringToWordVector

**SYNOPSIS:** Converts String attributes into a set of attributes representing word occurrence information from the text contained in the strings. The set of words (attributes) is determined by the first batch filtered (typically training data).

**OPTIONS:**

IDFTransform -- Sets whether if the word frequencies in a document should be transformed into:

$$f_{ij} * \log(\text{num of Docs} / \text{num of Docs with word } i)$$

where  $f_{ij}$  is the frequency of word  $i$  in document (instance)  $j$ .



# Data preprocessing and visualization

- Change the attributes to nominal (use NumericToBinary filter)
- Save data on a file for further use

The screenshot shows the Weka Explorer interface with the 'Preprocess' tab selected. The 'Filter' section shows 'NumericToBinary' applied. The 'Current relation' is 'departments\_string-weka.filters.unsupervised.attribute.String...' with 20 instances and 612 attributes. The 'Attributes' list shows 'academic\_binarized' selected. The 'Selected attribute' section shows 'Name: academic\_binarized', 'Type: Nominal', 'Missing: 0 (0%)', 'Distinct: 2', and 'Unique: 0 (0%)'. A table below shows the distribution of the attribute:

Label	Count
0	16
1	4

The 'Class' is 'Copy of document\_class (Nom)'. A bar chart visualizes the distribution, showing a red bar for '0' with a count of 16 and a blue bar for '1' with a count of 4. The status bar shows 'OK' and a 'Log' button.



# Attribute Selection

*Finding a minimal set of attributes that preserve the class distribution*

Attribute relevance with respect to the class – not relevant attribute (*accounting*)

The screenshot shows the Weka Explorer interface. The 'Selected attribute' section displays the following information:

Name	Type
accounting	Nominal

Missing: 0 (0%)      Distinct: 2      Unique: 1 (5%)

Label	Count
0	19
1	1

Class: class (Nom)      Visualize All

The 'Attributes' list on the left shows the following attributes:

No.	Name
1	document_name
2	academic
3	accelerator
4	accounting
5	accreditation
6	accredited
7	activities
8	actuarial
9	addition
10	administration
11	advanced
12	advised
13	advisor

The 'Visualize All' button is active, and a bar chart is displayed below it, showing a red bar for label '0' with a count of 19 and a blue bar for label '1' with a count of 1.

IF accounting=1 THEN class=A (Error=0, Coverage = 1 instance → **overfitting** )

IF accounting=0 THEN class=B (Error=10/19, Coverage = 19 instances → **low accuracy**)



# Attribute Selection

Attribute relevance with respect to the class – relevant attribute (*science*)

The screenshot shows the Weka Explorer interface with the 'Select attributes' tab active. The 'Current relation' is 'departments\_string-weka.filters.unsupervised.attribute.String...' with 20 instances and 612 attributes. The 'Selected attribute' is 'science', which is a nominal attribute with 2 distinct values and 0 missing values. A table shows the distribution of the 'science' attribute:

Label	Count
0	13
1	7

The 'Attributes' list on the left shows 'science' selected. A bar chart at the bottom right visualizes the distribution, with a red bar for label 0 (count 13) and a blue bar for label 1 (count 7). The status bar at the bottom shows 'OK' and a 'Log' button.

IF accounting=1 THEN class=A (Error=0, Coverage = 7 instance)

IF accounting=0 THEN class=B (Error=4/13, Coverage = 13 instances)

# Attribute Selection (with document\_name)

The image shows two overlapping windows from the Weka software. The background window is the 'Weka Explorer' interface, and the foreground window is an 'Information' dialog box.

**Weka Explorer (Background Window):**

- Buttons: Preprocess, Classify, Cluster, Associate, **Select attributes**, Visualize
- Attribute Evaluator: Choose **CfsSubsetEval**
- Search Method: Choose **BestFirst -D 1 -N 5**
- Attribute Selection Mode:
  - Use full training set
  - Cross-validation (Folds: 10, Seed: 1)
- (Nom) class: [dropdown]
- Buttons: Start, Stop
- Result list (right-click for options): 01:49:21 - BestFirst + CfsSubsetEval
- Attributes list:

No.	Name
1	<input checked="" type="checkbox"/> document_name
2	<input type="checkbox"/> academic
3	<input type="checkbox"/> accelerator
4	<input type="checkbox"/> accounting
5	<input type="checkbox"/> accreditation
6	<input type="checkbox"/> accredited
7	<input type="checkbox"/> activities
8	<input type="checkbox"/> actuarial
9	<input type="checkbox"/> addition
10	<input type="checkbox"/> administration
11	<input type="checkbox"/> advanced
12	<input type="checkbox"/> advised
13	<input type="checkbox"/> advisor
- Status: OK

**Information Window (Foreground):**

- NAME: weka.attributeSelection.CfsSubsetEval
- SYNOPSIS: CfsSubsetEval :  
Evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them.  
Subsets of features that are highly correlated with the class while having low intercorrelation are preferred.
- OPTIONS: locallyPredictive -- Identify locally predictive attributes. Iteratively adds attributes with the highest correlation with the class as long as there is not already an attribute in the subset

**Attribute Selection Output (Weka Explorer):**

```
=== Attribute selection on all input data ===
Search Method:
  Best first.
  Start set: no attributes
  Search direction: forward
  Stale search after 5 node expansions
  Total number of subsets evaluated: 0.0
  Merit of best subset found: 0.0

Attribute Subset Evaluator (supervised, CfsSubsetEvaluator)
Including locally predictive attributes

Selected attributes: 1,307,317 : 3
  document_name
  research
  science
```

**Attribute Selection Bar Chart (Weka Explorer):**

Attribute	Value
Biology	1
Chemistry	1
Communication	1
Computer	1
Justice	1
Economics	1

A bar chart below the table shows 13 vertical bars, alternating in color between red and blue, representing the selection status of the attributes listed in the table above.

# Attribute Selection (without document\_name)

The image shows two overlapping windows of the Weka Explorer software. The background window is in the 'Select attributes' tab, displaying a list of 611 attributes. The 'academic' attribute is selected, and its statistics are shown: Name: academic, Missing: 0 (0%), Distinct: 2, Type: Nominal, Unique: 0 (0%). A bar chart shows the distribution of the 'academic' attribute with a count of 16 for each class.

The foreground window is also in the 'Select attributes' tab but shows the 'Attribute Evaluator' set to 'CfsSubsetEval' and the 'Search Method' set to 'BestFirst -D 1 -N 5'. The 'Attribute Selection Mode' is set to 'Use full training set'. The 'Attribute selection output' window displays the following text:

```
ATTRIBUTE SUBSET EVALUATOR (SUPERVISED, CLASS (NOMINAL): ALL CLASS):  
CFS Subset Evaluator  
Including locally predictive attributes  
Selected attributes: 27,36,49,170,217,306,316,385,386,389,395,417,482 : 13  
areas  
ba  
business  
include  
making  
research  
science  
acting  
active  
apply  
based  
concentration  
history
```

The 'Result list' shows the execution time for the attribute selection process: 01:49:21 - BestFirst + CfsSubsetEval and 01:55:14 - BestFirst + CfsSubsetEval.

# Attribute Selection (ranking)

The image displays two screenshots of the Weka Explorer interface, illustrating the results of an attribute selection process using the Ranker method with GainRatioAttributeEval.

**Left Screenshot: Ranked attributes**

Gain Ratio	Feature	Evaluated
0.5211466	307 research	
0.4431742	317 science	
0.3223913	483 history	
0.2795499	327 social	
0.2795499	37 ba	
0.2795499	79 concentrations	
0.2667478	546 polish	
0.2667478	589 technical	
0.2667478	492 interdisciplinary	
0.2667478	518 media	
0.2667478	515 maloney	
0.2667478	429 creative	
0.2667478	505 languages	
0.2667478	526 national	
0.2667478	396 based	
0.2667478	511 literature	
0.2667478	466 foreign	

**Right Screenshot: Attribute selection output**

Gain Ratio	Feature	Evaluated
0.0017186	322 select	
0.0017186	178 institute	
0.0017186	187 internship	
0.0017186	324 service	
0.0017186	348 suite	
0.0017186	315 school	
0.0017186	173 independent	
0.0017186	75 complete	
0.0017186	71 community	
0.0017186	316 schools	
0.0007727	216 majors	
0.0000741	188 internships	
0	262 phone	
0	346 study	
0	211 location	
0	98 department	
0	377 website	
0	59 chair	

**Selected attributes:** 307,317,483,327,37,79,546

# Attribute Selection (explanation of ranking)

The screenshot shows the Weka Explorer interface with the 'Visualize' tab selected. The 'Current relation' is 'departments\_string-weka.filters.unsupervised.attribute.String...' with 20 instances and 612 attributes. The 'Selected attribute' is 'research', which is a nominal attribute with 2 distinct values and 0 missing values. A table shows the distribution of the 'research' attribute:

Label	Count
0	12
1	8

A bar chart below the table visualizes this distribution, with a red bar for label 0 (count 12) and a blue bar for label 1 (count 8). The 'Attributes' list on the left shows 'research' selected at index 307. The status bar at the bottom indicates 'OK'.

The screenshot shows the Weka Explorer interface with the 'Visualize' tab selected. The 'Current relation' is 'departments\_string-weka.filters.unsupervised.attribute.String...' with 20 instances and 612 attributes. The 'Selected attribute' is 'chair', which is a nominal attribute with 1 distinct value and 0 missing values. A table shows the distribution of the 'chair' attribute:

Label	Count
0	0
1	20

A bar chart below the table visualizes this distribution, with a red bar for label 0 (count 0) and a blue bar for label 1 (count 20). The 'Attributes' list on the left shows 'chair' selected at index 59. The status bar at the bottom indicates 'OK'.

# Attribute Selection (using filters)

- Choose filters/supervised/attribute/AttributeSelection
- Set parameters to InfoGainAttributeEval and Ranker
- Click on Apply and see the attribute ordering

The screenshot shows the Weka Explorer interface with the 'AttributeSelection' filter applied. The 'research' attribute is selected, and its distribution is visualized in a bar chart. The 'InfoGainAttributeEval' evaluator and 'Ranker' search method are selected for the filter.

**Weka Explorer**

Preprocess | Classify | Cluster | Associate | **Select attributes** | Visualize

Open file... | Open URL... | Open DB... | Undo

Filter: **AttributeSelection** -E "weka.attributeSelection.InfoGainAttributeEval" -S "weka.attributeSelection.Ranker"

Current relation: departments\_string-weka.filters.unsupervised.attribute.String...  
Instances: 20 | Attributes: 612

Attributes: All | None | Invert

No.	Name
1	<input type="checkbox"/> document_name
2	<input checked="" type="checkbox"/> research
3	<input type="checkbox"/> science
4	<input type="checkbox"/> concentrations
5	<input type="checkbox"/> ba
6	<input type="checkbox"/> social
7	<input type="checkbox"/> history
8	<input type="checkbox"/> biological
9	<input type="checkbox"/> copernicus
10	<input type="checkbox"/> environmental
11	<input type="checkbox"/> include
12	<input type="checkbox"/> sciences
13	<input type="checkbox"/> business

Remove

Status: OK

**weka.gui.GenericObjectEditor**

weka.filters.supervised.attribute.AttributeSelection

About: A supervised attribute filter that can be used to select attributes. [More](#)

evaluator: **InfoGainAttributeEval**

search: **Ranker -T -1.7976931348623157E308 -N -1**

Open... | Save... | **OK** | Cancel

Selected attribute: Name: research | Missing: 0 (0%) | Distinct: 2 | Type: Nominal | Unique: 0 (0%)

Label	Count
0	12
1	8

Class: class (Nom) | Visualize All

# Attribute Selection (using filters)

The screenshot displays the Weka Explorer interface. The top menu bar includes 'Preprocess', 'Classify', 'Cluster', 'Associate', 'Select attributes', and 'Visualize'. Below the menu are buttons for 'Open file...', 'Open URL...', 'Open DB...', 'Undo', 'Edit...', and 'Save...'. The 'Filter' section shows the 'AttributeSelection' filter applied with the command: `-E "weka.attributeSelection.GainRatioAttributeEval" -S "weka.attributeSelection.Ranker -T -1.797693134862315"`. The 'Current relation' section shows 'Relation: departments\_string-weka.filters.unsupervised.attribute.String...' with 20 instances and 4 attributes. The 'Attributes' section lists 'research', 'science', 'history', and 'class', with 'research' selected. The 'Selected attribute' section shows a table with labels 0 and 1, and counts 12 and 8 respectively. The 'Class' section shows 'class (Nom)' and a 'Visualize All' button. The bottom window shows four bar charts for 'research', 'science', 'history', and 'class', each with two bars (red and blue) representing different classes. The 'research' chart has values 12 and 8, 'science' has 13 and 7, 'history' has 17 and 3, and 'class' has 11 and 9.

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... | Open URL... | Open DB... | Undo | Edit... | Save...

Filter

Choose **AttributeSelection** -E "weka.attributeSelection.GainRatioAttributeEval" -S "weka.attributeSelection.Ranker -T -1.797693134862315" Apply

Current relation

Relation: departments\_string-weka.filters.unsupervised.attribute.String...  
Instances: 20 | Attributes: 4

Attributes

All | None | Invert

No.	Name
1	<input checked="" type="checkbox"/> research
2	<input type="checkbox"/> science
3	<input type="checkbox"/> history
4	<input type="checkbox"/> class

Selected attribute

Name: research | Type: Nominal  
Missing: 0 (0%) | Distinct: 2 | Unique: 0 (0%)

Label	Count
0	12
1	8

Class: class (Nom) | Visualize All

research

Class	Count
0	12
1	8

science

Class	Count
0	13
1	7

history

Class	Count
0	17
1	3

class

Class	Count
0	11
1	9

Log | x 0

# Classification – creating models (hypotheses)

*Mapping (independent attributes -> class)*

## Inferring rudimentary rules - OneR

Weather data (weather.nominal.arff)

No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	play Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Attribute	Rules	Errors	Total error
outlook	sunny -> no overcast -> yes rainy -> yes	2/5 0/4 2/5	4/14
temperature	hot -> no mild -> yes cool -> yes	2/4 2/6 1/4	5/14
humidity	high -> no normal -> yes	3/7 1/7	4/14
windy	false -> yes true -> no	2/8 3/5	5/14



# Classification – OneR

The screenshot shows the Weka Explorer interface with the OneR classifier selected. The classifier output window displays the following information:

```
Attributes: 5
            outlook
            temperature
            humidity
            windy
            play
Test mode:  evaluate on training data

=== Classifier model (full training set) ===

outlook:
    sunny   -> no
    overcast -> yes
    rainy   -> yes
(10/14 instances correct)

Time taken to build model: 0 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances      10      71.4286 %
Incorrectly Classified Instances    4       28.5714 %
```

The interface also shows test options (Use training set selected, 10 folds, 66% split) and a result list containing the entry "16:04:47 - rules\_OneR".

# Classification – decision tree

Right click on the highlighted line in Result list and choose Visualize tree

The screenshot displays the Weka Explorer interface. The 'Classifier' tab is active, showing 'J48 -C 0.25 -M 2' selected. The 'Test options' section includes 'Use training set' selected, 'Cross-validation' with 'Folds' set to 10 and 'Percentage split' at 66%. The 'Result list' shows two entries: '16:04:47 - rules.OneR' and '16:09:06 - trees.J48', with the latter highlighted. The 'Classifier output' pane shows the following text:

```
J48 pruned tree
-----
outlook = sunny
|  humidity = high: no (3.0)
|  humidity = normal: yes (2.0)
outlook = overcast: yes (4.0)
outlook = rainy
|  windy = TRUE: no (2.0)
|  windy = FALSE: yes (3.0)

Number of Leaves :    5
Size of the tree :    8

Time taken to build model: 0.13 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances      14          100 %
Incorrectly Classified Instances     0           0 %
```

The 'Weka Classifier Tree Visualizer' window shows a tree diagram with the root node 'outlook'. The 'outlook' node branches into 'sunny', 'overcast', and 'rainy'. The 'overcast' branch leads to a leaf node 'yes (4.0)'. The 'sunny' branch leads to a 'humidity' node, which branches into 'high' (leaf 'no (3.0)') and 'normal' (leaf 'yes (2.0)'). The 'rainy' branch leads to a 'windy' node, which branches into 'TRUE' (leaf 'no (2.0)') and 'FALSE' (leaf 'yes (3.0)').

# Classification – decision tree

Top-down induction of decision trees (TDIDT, old approach know from pattern recognition):

- Select an attribute for root node and create a branch for each possible attribute value.
- Split the instances into subsets (one for each branch extending from the node).
- Repeat the procedure recursively for each branch, using only instances that reach the branch (those that satisfy the conditions along the path from the root to the branch).
- Stop if all instances have the same class.

ID3, C4.5, J48 (Weka): Select the attribute that minimizes the class entropy in the split.

# Classification – numeric attributes

weather.arff

The image shows a Weka Explorer window with the following components:

- Classifier:** J48 -C 0.25 -M 2
- Test options:** Use training set (selected), Folds: 10, Percentage split: 66.
- Classifier output:**

```
J48 pruned tree
-----
outlook = sunny
| humidity <= 75: yes (2.0)
| humidity > 75: no (3.0)
outlook = overcast: yes (4.0)
outlook = rainy
| windy = TRUE: no (2.0)
| windy = FALSE: yes (3.0)
```
- Statistics:** Number of Leaves: 5, Size of the tree: 8, Time taken to build model: 0 seconds.
- Evaluation:** Evaluation on training set: 14/100 (100%) correctly classified, 0/0 (0%) incorrectly classified.
- Viewer:** Relation: weather. Table with 14 rows and 6 columns: No., outlook (Nominal), temperature (Numeric), humidity (Numeric), windy (Nominal), play (Nominal).
- Weka Classifier Tree Visualizer:** A tree diagram showing the structure of the J48 classifier. The root node is 'outlook', which branches into 'sunny', 'overcast', and 'rainy'. 'sunny' leads to a 'humidity' node, which branches into '<= 75' (yes 2.0) and '> 75' (no 3.0). 'overcast' leads to a leaf node 'yes (4.0)'. 'rainy' leads to a 'windy' node, which branches into '= TRUE' (no 2.0) and '= FALSE' (yes 3.0).

# Classification – predicting class

Click on Set...

Click on Open file...

The screenshot displays the Weka Explorer interface with several windows open. The main window shows the 'Test Instances' dialog with 'weather.nominal.test' selected. The 'Open' dialog shows the file 'weather.nominal.test' selected in the 'Weka Data' folder. The 'weather.nominal.test - WordPad' window shows the file's contents, including the relation name, attributes, and data. The main window shows the classifier output for a J48 tree, including the model structure and evaluation results.

**Weka Explorer - Test Instances**  
Relation: weather.nominal.test  
Instances: 1    Attributes: 5  
Buttons: Open file...    Open URL...

**Weka Explorer - Classifier**  
Classifier: J48 -C 0.25 -M 2

**Weka Explorer - Test options**  
 Use training set  
 Supplied test set    Set...  
 Cross-validation    Folds: 10  
 Percentage split    %: 66  
More options...

**Weka Explorer - Classifier output**  
Test mode: user supplied test set: 1 instances  
=== Classifier model (full training set) ===  
J48 pruned tree  
-----  
outlook = sunny  
| humidity = high: no (3.0)  
| humidity = normal: yes (2.0)  
outlook = overcast: yes (4.0)  
outlook = rainy  
| windy = TRUE: no (2.0)  
| windy = FALSE: yes (3.0)  
Number of Leaves : 5  
Size of the tree : 8  
Time taken to build model: 0 seconds  
=== Evaluation on test set ===  
=== Summary ===  
Correctly Classified Instances    0            0    %  
Incorrectly Classified Instances    1            100    %

**Weka Explorer - Result list**  
16:48:34 - trees.J48  
16:53:08 - trees.J48

**Open**  
Look in: Weka Data  
Files of type: Arff data files  
File name: weather.nominal.test  
Buttons: Open    Cancel

**weather.nominal.test - WordPad**  
@relation weather.nominal.test  
  
@attribute outlook {sunny, overcast, rainy}  
@attribute temperature {hot, mild, cool}  
@attribute humidity {high, normal}  
@attribute windy {TRUE, FALSE}  
@attribute play {yes, no}  
  
@data  
sunny,mild,normal,FALSE,no

**Weka Explorer - Status**  
OK    Log    x 0

# Classification – predicting class

Right click on the highlighted line in Result list and choose Visualize classifier errors

The screenshot displays the Weka Explorer interface. The 'Classifier' tab is active, showing the 'J48 -C 0.25 -M 2' classifier. The 'Test options' section is set to 'Supplied test set'. The 'Result list' on the left shows two entries: '16:48:34 - trees.J48' and '16:53:08 - trees.J48', with the latter highlighted in purple. The 'Classifier output' pane shows the following text:

```
Test mode: user supplied test set
=== Classifier model (full training) ===
J48 pruned tree
-----
outlook = sunny
| humidity = high: no (3.0)
| humidity = normal: yes (2.0)
outlook = overcast: yes (4.0)
outlook = rainy
| windy = TRUE: no (2.0)
| windy = FALSE: yes (3.0)

Number of Leaves : 5
Size of the tree : 8
Time taken to build model: 0 seconds

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances 0 0 %
Incorrectly Classified Instances 1 100 %
```

The 'Weka Classifier Visualize' dialog box is open, showing the following settings:

- X: outlook (Nom)
- Y: temperature (Nom)
- Colour: play (Nom)
- Select Instance: (empty)

The plot area shows a 2D scatter plot with 'outlook' on the x-axis (values: sunny, overcast) and 'temperature' on the y-axis (values: cool, mild, hot). A red square is plotted at the intersection of 'sunny' and 'mild'. The 'Class colour' legend shows 'yes' in blue and 'no' in red.

The 'Weka : Instance info' dialog box is also open, displaying the following information:

```
Plot : 16:53:08 - trees.J48
Instance: 0
Instance_number : 0.0
outlook : sunny
temperature : mild
humidity : normal
windy : FALSE
predictedplay : yes
play : no
```

The 'Click on the square' text is positioned above the 'Weka Classifier Visualize' dialog box, indicating the action to be taken on the highlighted instance in the result list.

# Classification – predicting class

Click on Save

The image shows two overlapping windows. The background window is 'Weka Classifier Visualize: 16:53:08 - trees.J48 (weather.sym)'. It has a top bar with the title and a menu bar. Below the menu bar are two dropdown menus: 'X: outlook (Nom)' and 'Y: temperature (Nom)'. Below these are two more dropdown menus: 'Colour: play (Nom)' and 'Select Instance'. There are three buttons: 'Reset', 'Clear', and 'Save'. Below the buttons is a plot area titled 'Plot: weather.symbolic\_predicted'. The plot shows a 2D space with 'outlook' on the x-axis (sunny, overcast, rainy) and 'temperature' on the y-axis (cool, mild, hot). A single red square is plotted at the intersection of 'sunny' and 'mild'. Below the plot is a 'Class colour' section with 'yes' in blue and 'no' in red. The foreground window is 'weather.nominal.test.error - WordPad'. It has a menu bar (File, Edit, View, Insert, Format, Help) and a toolbar. The text area contains the following text:

```
@relation weather.symbolic_predicted  
  
@attribute Instance_number numeric  
@attribute outlook {sunny,overcast,rainy}  
@attribute temperature {hot,mild,cool}  
@attribute humidity {high,normal}  
@attribute windy {TRUE,FALSE}  
@attribute predictedplay {yes,no}  
@attribute play {yes,no}  
  
@data  
0,sunny,mild,normal,FALSE,yes,no
```

# Prediction (no model, lazy learning)

Relation: weather.symbolic

No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	play Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Buttons: Undo, OK, Cancel

test: (sunny, cool, high, TRUE, ?)

- K-nearest neighbor (KNN, IBk)  
*Take the class of the nearest neighbor or the majority class among K neighbors*  
 K=1 -> no  
 K=3 -> no  
 K=5 -> yes  
 K=14 -> yes (Majority predictor, ZeroR)

- Weighted K-nearest neighbor  
 K=5 -> undecided  
 $no = 1/1 + 1/2 = 1.5$   
 $yes = 1/2 + 1/2 + 1/2 = 1.5$

X	2	8	9	11	12	...	10
Distance(test,X)	1	2	2	2	2	...	4
play	no	no	yes	yes	yes	...	yes

- Distance is calculated as the number of different attribute values
- Euclidean distance for numeric attributes



# Prediction (no model, lazy learning)

The image shows the Weka Explorer interface with the 'IBk -K 1 -W 0' classifier selected. The 'Test options' section is set to 'Supplied test set'. The 'Classifier output' pane shows the following text:

```
Instances: 14
Attributes: 5
           outlook
           temperatur
           humidity
           windy
           play
Test mode: user supp
=== Classifier model (f
IB1 instance-based clas
using 1 nearest neighbour(s) for classification

Time taken to build model: 0 seconds

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances      0          0      %
Incorrectly Classified Instances    1          100     %
```

The 'weka.gui.GenericObjectEditor' dialog box is open, showing the configuration for the 'weka.classifiers.lazy.IBk' classifier. The 'About' tab is active, displaying 'K-nearest neighbours classifier.' and a 'More' button. The configuration parameters are:

- KNN: 1
- crossValidate: False
- debug: False
- distanceWeighting: No distance weighting
- meanSquared: False
- noNormalization: False
- windowSize: 0

The dialog box has 'Open...', 'Save...', 'OK', and 'Cancel' buttons at the bottom.

# Prediction (no model, lazy learning)

Departments-binary-test.arff

**Selected attribute**

Label	Count
Anthropology	0
Art	0
Biology	0
Chemistry	0
Communication	0
Computer	0
Justice	0

**Class: class (Nom)**

Label	Count
0	0
1	1
2	1
3	1
4	1
5	1
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0

Departments-binary-training

**Selected attribute**

Label	Count
Music	1
Philosophy	1
Physics	0
Political	0
Psychology	0
Sociology	0
Theatre	0

**Class: class (Nom)**

Label	Count
0	0
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1

# Prediction (no model, lazy learning)

**Weka Explorer**

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Classifier: Choose **IBk -K 1 -W 0**

Test options:

- Use training set
- Supplied test set
- Cross-validation Folds:
- Percentage split %:

(Nom) class:

Result list (right-click for options):

- 03:15:24 - lazy IBk

Classifier output:

```
Test mode: user supplied test set: 5 instances

=== Classifier model (full training set) ===

IB1 instance-based classifier
using 1 nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Predictions on test set ===

inst#, actual, predicted, error, probability distribution
1 2:B 2:B 0.059 *0.941
2 2:B 2:B 0.059 *0.941
3 2:B 2:B 0.059 *0.941
4 2:B 2:B 0.059 *0.941
5 2:B 2:B 0.059 *0.941

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances 5 100 %
Incorrectly Classified Instances 0 0 %
```

Status: OK   x 0

# Model evaluation – holdout (percentage split)

Click on More options...

The screenshot shows the Weka Explorer interface with the 'Classifier' tab selected. The 'IBk -K 1 -W 0' classifier is chosen. Under 'Test options', 'Percentage split' is selected with a value of 66%. A 'More options...' button is visible. The 'Classifier output' pane displays the following text:

```
Test mode: split 66% train, remainder test

=== Classifier model (full training set) ===

IB1 instance-based classifier
using 1 nearest neighbour(s) for classification

Time taken to build model: 0 seconds

=== Predictions on test split ===

inst#, actual, predicted, error, probability distribution
1 1:yes 1:yes *0.5 0.5
2 1:yes 1:yes *0.787 0.213
3 2:no 1:yes + *0.909 0.091
4 2:no 1:yes + *0.5 0.5
5 1:yes 2:no + 0.091 *0.909

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

Correctly Classified Instances 2 40 %
Incorrectly Classified Instances 3 60 %
```

The 'Classifier evaluation options' dialog box is open, showing the following settings:

- Output model
- Output per-class stats
- Output entropy evaluation measures
- Output confusion matrix
- Store predictions for visualization
- Output predictions
- Cost-sensitive evaluation
- Random seed for XVal / % Split: 1

The 'More options...' button in the dialog is labeled 'Set...'. The 'OK' button is at the bottom of the dialog.

# Model evaluation – cross validation

The screenshot shows the Weka Explorer interface with the following components:

- Classifier:** IBk -K 1 -W 0
- Test options:** Cross-validation is selected with 7 folds and 66% split.
- Classifier output:** Displays predictions on test data and stratified cross-validation summary.
- Result list:** Shows three recent lazy.IBk operations.
- Status:** OK

**Classifier output details:**

=== Predictions on test data ===

inst#	actual	predicted	error	probability distribution
1	2:no	1:yes	+	*0.962 0.038
2	1:yes	1:yes		*0.5 0.5
1	2:no	1:yes	+	*0.962 0.038
2	1:yes	1:yes		*0.962 0.038
1	2:no	2:no		0.071 *0.929
2	1:yes	1:yes		*0.658 0.342
1	2:no	1:yes	+	*0.5 0.5
2	1:yes	1:yes		*0.929 0.071
1	2:no	2:no		0.342 *0.658
2	1:yes	1:yes		*0.929 0.071
1	1:yes	1:yes		*0.5 0.5
2	1:yes	2:no	+	0.342 *0.658
1	1:yes	2:no	+	0.071 *0.929
2	1:yes	2:no	+	0.071 *0.929

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	8	57.1429 %
Incorrectly Classified Instances	6	42.8571 %

# Model evaluation – leave one out cross validation

The screenshot shows the Weka Explorer interface. The 'Classifier' tab is selected, and the classifier is set to 'IBk -K 1 -W 0'. The 'Test options' section is configured for 'Cross-validation' with 'Folds' set to 14. The 'Classifier output' window displays the results of the cross-validation process.

**Test options:**

- Use training set
- Supplied test set (Set...)
- Cross-validation (Folds: 14)
- Percentage split (%: 66)

**Classifier output:**

```
=== Predictions on test data ===

inst#,   actual, predicted, error, probability distribution
  1      2:no      1:yes      + *0.964 0.036
  1      2:no      1:yes      + *0.964 0.036
  1      2:no      2:no                0.067 *0.933
  1      2:no      1:yes      + *0.5    0.5
  1      2:no      2:no                0.341 *0.659
  1      1:yes     1:yes      *0.5    0.5
  1      1:yes     2:no      + 0.067 *0.933
  1      1:yes     1:yes      *0.5    0.5
  1      1:yes     1:yes      *0.964 0.036
  1      1:yes     1:yes      *0.659 0.341
  1      1:yes     1:yes      *0.933 0.067
  1      1:yes     1:yes      *0.933 0.067
  1      1:yes     2:no      + 0.341 *0.659
  1      1:yes     2:no      + 0.067 *0.933

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      8          57.1429 %
Incorrectly Classified Instances    6          42.8571 %
```

**Result list (right-click for options):**

- 18:14:47 - lazy.IBk
- 18:18:57 - lazy.IBk
- 18:21:32 - lazy.IBk
- 18:25:07 - lazy.IBk
- 18:26:49 - lazy.IBk**

**Status:** OK

# Model evaluation – confusion (contingency) matrix

	predicted	
	a	b
actual	a	2
	b	1
	2	0

**Weka Explorer**

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Classifier: Choose **J48 -C 0.25 -M 2**

Test options:
 

- Use training set
- Supplied test set
- Cross-validation Folds: 10
- Percentage split %: 66

Classifier output:

```

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: weather.symbolic
Instances: 14
Attributes: 5
  outlook
  temperature
  humidity
  windy
  play
Test mode: split 66% train, remainder test

=== Classifier model (full training set) ===

J48 pruned tree

outlook = sunny
| humidity = high: no (3.0)
| humidity = normal: yes (2.0)
outlook = overcast: yes (4.0)
outlook = rainy
| windy = TRUE: no (2.0)
| windy = FALSE: yes (3.0)
    
```

Result list (right-click for options):

- 03:15:24 - lazy.IBk
- 03:18:05 - trees.J48
- 03:18:36 - trees.J48

Status: OK

**Weka Explorer**

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Classifier: Choose **J48 -C 0.25 -M 2**

Test options:
 

- Use training set
- Supplied test set
- Cross-validation Folds: 10
- Percentage split %: 66

Classifier output:

```

=== Summary ===

Correctly Classified Instances      2      40
Incorrectly Classified Instances    3      60
Kappa statistic                     -0.3636
Mean absolute error                  0.6
Root mean squared error              0.7746
Relative absolute error             126.9231 %
Root relative squared error         157.6801 %
Total Number of Instances           5

=== Confusion Matrix ===

 a b  <-- classified as
 2 1 | a = yes
 2 0 | b = no
    
```

Result list (right-click for options):

- 03:15:24 - lazy.IBk
- 03:18:05 - trees.J48
- 03:18:36 - trees.J48

Status: OK

Log x 0

# Clustering – k-means

Click on Ignore attributes

The screenshot shows the Weka Explorer interface with the 'Cluster' tab selected. The 'Clusterer' is set to 'SimpleKMeans -N 2 -S 10'. The 'Cluster mode' is 'Use training set'. The 'Ignore attributes' field is empty. The 'Result list' shows a job titled '04:07:45 - SimpleKMeans'. The 'Clusterer output' pane displays the following text:

```
kMeans
=====
Number of iterations: 2
Within cluster sum of squares criterion:
Cluster centroids:

Cluster 0
  Mean/Mode: Chemistry 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  Std Devs:  N/A  N/A  N/A  N/A  N/A  N/A  N/A  N/A  N/A  N/A  N/A  N/A  N/A  N/A  N/A
Cluster 1
  Mean/Mode: Anthropology 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  Std Devs:  N/A  N/A  N/A  N/A  N/A  N/A  N/A  N/A  N/A  N/A  N/A  N/A  N/A  N/A  N/A

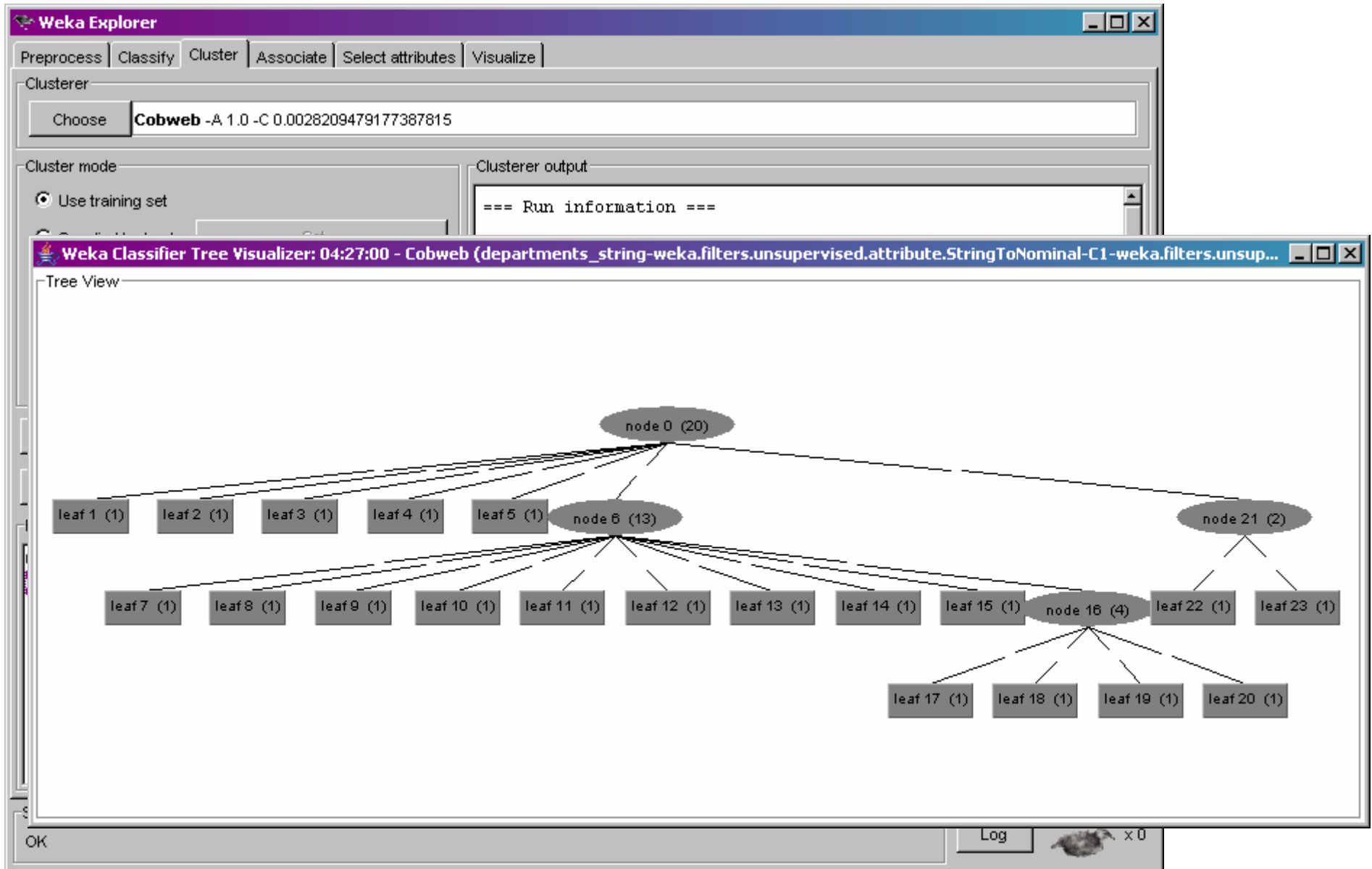
Clustered Instances

 0      3 ( 15%)
 1     17 ( 85%)
```

The 'Weka Clusterer Visualize' window shows a 2D plot with X-axis 'research (Nom)' and Y-axis 'science (Nom)'. The plot contains several data points (represented by 'x' marks) colored according to two clusters: 'cluster0' (blue) and 'cluster1' (red). The 'Class colour' section below the plot shows 'cluster0' in blue and 'cluster1' in red.



# Hierarchical Clustering – Cobweb



# Association Rules (A => B)

- *Confidence* (accuracy):  $P(B|A) = (\# \text{ of tuples containing both } A \text{ and } B) / (\# \text{ of tuples containing } A)$ .
- *Support* (coverage):  $P(A,B) = (\# \text{ of tuples containing both } A \text{ and } B) / (\text{total } \# \text{ of tuples})$

The screenshot shows the Weka Explorer interface with the Apriori algorithm executed. The 'Associator output' window displays the following results:

```
Generated sets of large itemsets:  
Size of set of large itemsets L(1): 12  
Size of set of large itemsets L(2): 47  
Size of set of large itemsets L(3): 39  
Size of set of large itemsets L(4): 6  
  
Best rules found:  
  
1. humidity=normal windy=FALSE 4 ==> play=yes 4    conf:(1)  
2. temperature=cool 4 ==> humidity=normal 4    conf:(1)  
3. outlook=overcast 4 ==> play=yes 4    conf:(1)  
4. temperature=cool play=yes 3 ==> humidity=normal 3    conf:(1)  
5. outlook=rainy windy=FALSE 3 ==> play=yes 3    conf:(1)  
6. outlook=rainy play=yes 3 ==> windy=FALSE 3    conf:(1)  
7. outlook=sunny humidity=high 3 ==> play=no 3    conf:(1)  
8. outlook=sunny play=no 3 ==> humidity=high 3    conf:(1)  
9. temperature=cool windy=FALSE 2 ==> humidity=normal play=yes 2    conf:(1)  
10. temperature=cool humidity=normal windy=FALSE 2 ==> play=yes 2    conf:(1)
```

The 'Viewer' window displays the 'weather.symbolic' relation with the following data:

No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	play Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

# Association Rules

**Weka Explorer**

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Associator

Choose **Apriori** -N 200 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0

Start Stop

Associator output

Result list (right-click for ...)

- 04:32:57 - Apriori
- 04:34:09 - Apriori
- 04:39:14 - Apriori


```

180. humidity=high play=no 4 ==> windy=FALSE 2
181. humidity=high windy=FALSE 4 ==> play=yes 2
182. humidity=high play=no 4 ==> windy=TRUE 2
183. temperature=cool 4 ==> windy=FALSE play=yes 2
184. temperature=cool 4 ==> humidity=normal windy=FALSE 2
185. temperature=cool humidity=normal 4 ==> windy=FALSE 2
186. humidity=normal windy=FALSE 4 ==> temperature=cool 2
187. temperature=cool 4 ==> humidity=normal windy=FALSE 2
188. temperature=cool humidity=normal 4 ==> windy=FALSE 2
189. temperature=mild play=yes 4 ==> windy=FALSE 2
190. temperature=mild play=yes 4 ==> windy=TRUE 2   conf: (0.5)
191. temperature=mild play=yes 4 ==> humidity=normal 2   conf: (0.5)
192. temperature=mild humidity=high 4 ==> play=no 2   conf: (0.5)
193. humidity=high play=no 4 ==> temperature=mild 2   conf: (0.5)
194. temperature=mild humidity=high 4 ==> play=yes 2   conf: (0.5)
195. temperature=mild play=yes 4 ==> humidity=high 2   conf: (0.5)
196. temperature=mild humidity=high 4 ==> windy=FALSE 2   conf: (0.5)
197. humidity=high windy=FALSE 4 ==> temperature=mild 2   conf: (0.5)
198. temperature=mild humidity=high 4 ==> windy=TRUE 2   conf: (0.5)
199. temperature=hot 4 ==> windy=FALSE play=yes 2   conf: (0.5)
200. temperature=hot 4 ==> humidity=high play=no 2   conf: (0.5)

```

Status

OK

Log  x 0

**Viewer**

Relation: weather.symbolic

No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	play Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Undo OK Cancel

And many more ...

Thank you!