An Introduction to the WEKA Data Mining System

Zdravko Markov
Central Connecticut State University
markovz@ccsu.edu

Ingrid Russell
University of Hartford
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) Poll**

Data 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, Equibits, 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
<table>
<thead>
<tr>
<th>Tool</th>
<th>Votes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPSS Clementine</td>
<td>135</td>
</tr>
<tr>
<td>SPSS</td>
<td>96</td>
</tr>
<tr>
<td>Excel</td>
<td>78</td>
</tr>
<tr>
<td>CART/MARS/TreeNet/RF</td>
<td>69</td>
</tr>
<tr>
<td>SAS</td>
<td>53</td>
</tr>
<tr>
<td>SAS Enterprise Miner</td>
<td>49</td>
</tr>
<tr>
<td>Your own code</td>
<td>39</td>
</tr>
<tr>
<td>Other free tools</td>
<td>34</td>
</tr>
<tr>
<td>Insightful Miner/ S-Plus</td>
<td>32</td>
</tr>
<tr>
<td>Statsoft Statistica</td>
<td>30</td>
</tr>
<tr>
<td>Weka</td>
<td>30</td>
</tr>
<tr>
<td>ThinkAnalytics</td>
<td>26</td>
</tr>
<tr>
<td>C4.5/C5.0/See5</td>
<td>25</td>
</tr>
<tr>
<td>R</td>
<td>25</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>23</td>
</tr>
<tr>
<td>Other commercial tools</td>
<td>23</td>
</tr>
<tr>
<td>MATLAB</td>
<td>16</td>
</tr>
<tr>
<td>Mineset (PurpleInsight)</td>
<td>16</td>
</tr>
<tr>
<td>Xelopes</td>
<td>16</td>
</tr>
<tr>
<td>Oracle Data Mining</td>
<td>10</td>
</tr>
<tr>
<td>Gornik</td>
<td>8</td>
</tr>
<tr>
<td>KXEN</td>
<td>7</td>
</tr>
<tr>
<td>IBM Iminer</td>
<td>5</td>
</tr>
<tr>
<td>Angoss</td>
<td>3</td>
</tr>
<tr>
<td>Equubits</td>
<td>3</td>
</tr>
<tr>
<td>Fair Isaac</td>
<td>3</td>
</tr>
<tr>
<td>GhostMiner</td>
<td>3</td>
</tr>
<tr>
<td>Megaputer</td>
<td>3</td>
</tr>
</tbody>
</table>
Weka Data Mining Software

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
– is 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.**
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 analyze 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 too large to be analyzed 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 publicly 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 inquisitive nature. (How should you pronounce WEKA? What does the weka sound like?)
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.

Google

Ian H. Witten, Eibe Frank

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

Comments

"If you have data that you want to analyze and understand, this book and the associated Weka toolset 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/

Machine Learning Experiences in Artificial Intelligence: A Multi-Institutional Project
NSF DUE 0716338

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 "site 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
This Jeopardy style 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.
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

Relations, attributes, tuples (instances)

Loan data – CVS format
(LoanData.cvs)
Data preprocessing and visualization


Overview

ARFF files have two distinct sections. The first section is the *Header* information, which is followed by the *Data* information.

The *Header* of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types. An example standard IRIS dataset looks like this:

```
% 1. Title: Iris Plants Database
% 2. Sources:
%   (a) Creator: R.A. Fisher
%   (b) Donor: Michael Marshall (MARRAL@FUDIO.ac.ams.gov)
%   (c) Date: July, 1988

@relation iris

@attribute sepal-length NUMERICAL
@attribute sepal-width NUMERICAL
@attribute petal-length NUMERICAL
@attribute petal-width NUMERICAL
@attribute class {Iris-setosa, Iris-versicolor, Iris-virginica}
```

```
@data
1,m,y,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,m,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,bad,n
11,f,n,18,50,8,20,car,y,1,good,y
12,f,y,20,50,10,20,car,n,2,good,y
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/

---

**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](http://www.cs.waikato.ac.nz/ml/weka/).

- **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](http://www.cs.waikato.ac.nz/ml/weka/) to download a self-extracting executable that includes Java VM 1.4 (weka-3-4-12.exe: 24,415,809 bytes)

    Click [here](http://www.cs.waikato.ac.nz/ml/weka/) 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](http://www.cs.waikato.ac.nz/ml/weka/) to download a disk image for OS X (weka-3-4-12.dmg: 13,565,484 bytes)

  - **Other platforms (Linux, etc.)**
    
    Click [here](http://www.cs.waikato.ac.nz/ml/weka/) 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…”)

![Image of Weka Explorer showing file open dialog](image-url)
Data preprocessing and visualization

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

[Image of Weka Explorer interface]

- Current relation: LoanData
- Attributes: 12
- Selected attribute: ID
- Missing: 0.0% (0)
- Distinct: 20
- Statistic:
  - Minimum
  - Maximum
  - Mean
  - StdDev

[Image of Save dialog box]
Data preprocessing and visualization

Editing data in Weka (click on ”Edit…”)

The image shows a Weka Explorer window with a dataset labeled LoanData. The dataset contains attributes such as ID, sex, married, age, money, pay, months, buy, emp, istemp, area, and approved. Each row represents an instance with values for these attributes.
Data preprocessing and visualization

Examining data
- Attribute type and properties
- Class (last attribute) distribution
Data preprocessing and visualization

Click on “Visualize All”
Data preprocessing and visualization

Web/Text documents - Department data

http://www.cs.ccsu.edu/~markov/
- Download Ch1, DMW Book
- Download datasets
Data preprocessing and visualization

Convert HTML to Text
Data preprocessing and visualization

Loading text data in Weka
- String format for ID and content
- One document per line
- Add class (nominal) if needed
Data preprocessing and visualization

Converting a string attribute into nominal

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

Converting a string attribute into nominal

Click on Apply – document_name is now 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”
Data preprocessing and visualization

Make the class attribute last

- Choose filters/unsupervised/attribute/Copy
- Set the index to 2 and click on Apply
- Remove attribute 2
Data preprocessing and visualization

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

ARFF file representing the department data in binary format (NonSparse)

Note the format (see SparseToNonSparse instance filter)
Attribute Selection

Finding a minimal set of attributes that preserve the class distribution

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

IF $\text{accounting}=1$ THEN $\text{class}=A$ (Error=0, Coverage = 1 instance → overfitting)

IF $\text{accounting}=0$ THEN $\text{class}=B$ (Error=10/19, Coverage = 19 instances → low accuracy)
Attribute Selection

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

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)

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 (without document_name)
Attribute Selection (ranking)

The Weka Explorer software is used to select attributes through ranking. The screenshots show the process of selecting attributes using the GainRatioAttributeEval evaluator. The results are displayed with ranked attributes and selected attributes.

For example, the ranked attributes might include:
- 0.521146 - research
- 0.441742 - science
- 0.322391 - history
- 0.275999 - social

The selected attributes are:
- 307, 317, 483, 327, 37, 79, 548
Attribute Selection (explanation of ranking)
Attribute Selection (using filters)

- Choose filters/supervised/attribute/AttributeSelection
- Set parameters to InfoGainAttributeEval and Ranker
- Click on Apply and see the attribute ordering
Attribute Selection (using filters)
Classification – creating models (hypotheses)

**Mapping (independent attributes -> class)**

Inferring rudimentary rules - OneR

Weather data (weather.nominal.arff)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Rules</th>
<th>Errors</th>
<th>Total error</th>
</tr>
</thead>
<tbody>
<tr>
<td>outlook</td>
<td>sunny -&gt; no</td>
<td>2/5</td>
<td>4/14</td>
</tr>
<tr>
<td></td>
<td>overcast -&gt; yes</td>
<td>0/4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rainy -&gt; yes</td>
<td>2/5</td>
<td></td>
</tr>
<tr>
<td>temperature</td>
<td>hot -&gt; no</td>
<td>2/4</td>
<td>5/14</td>
</tr>
<tr>
<td></td>
<td>mild -&gt; yes</td>
<td>2/6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>cool -&gt; yes</td>
<td>1/4</td>
<td></td>
</tr>
<tr>
<td>humidity</td>
<td>high -&gt; no</td>
<td>3/7</td>
<td>4/14</td>
</tr>
<tr>
<td></td>
<td>normal -&gt; yes</td>
<td>1/7</td>
<td></td>
</tr>
<tr>
<td>windy</td>
<td>false -&gt; yes</td>
<td>2/8</td>
<td>5/14</td>
</tr>
<tr>
<td></td>
<td>true -&gt; no</td>
<td>3/5</td>
<td></td>
</tr>
</tbody>
</table>
Classification – OneR

Attibutes: 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 %
Classification – decision tree

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

```
Classifier output:
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
```

Correctly Classified Instances 14 100 %
Incorrectly Classified Instances 0 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

J48 pruned tree
---------
outlook = sunny
| humidity <= 75: yes (2.0)
| humidity > 75: no (3.0)
outlook = overcast
| 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

Correctly Classified Instances 14 100%
Incorrectly Classified Instances 0 0%
Classification – predicting class

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 ===

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

@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,TRUE,TRUE
sunny,mild,normal,FALSE,yes
sunny,hot,high,TRUE,no
sunny,hot,high,FALSE,yes
overcast,mild,normal,TRUE,yes
overcast,mild,normal,FALSE,no
rainy,hot,high,TRUE,yes
rainy,hot,high,FALSE,no
rainy,mild,high,TRUE,yes
rainy,mild,high,FALSE,yes

Classification – predicting class

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

Click on the square
Classification – predicting class

Click on Save
Prediction (no model, lazy learning)

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

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

- **Weighted K-nearest neighbor**
  $K=5 \rightarrow \text{undecided}$  
  $\text{no}=\frac{1}{1}+\frac{1}{2}=1.5$  
  $\text{yes}=\frac{1}{2}+\frac{1}{2}+\frac{1}{2}=1.5$

<table>
<thead>
<tr>
<th>X</th>
<th>2</th>
<th>8</th>
<th>9</th>
<th>11</th>
<th>12</th>
<th>…</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance(test,X)</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>…</td>
<td>4</td>
</tr>
<tr>
<td>play</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>…</td>
<td>yes</td>
</tr>
</tbody>
</table>

- Distance is calculated as the number of different attribute values
- Euclidean distance for numeric attributes
Prediction (no model, lazy learning)
Prediction (no model, lazy learning)

Departments-binary-test.arff

Departments-binary-training
Prediction (no model, lazy learning)

Test mode: user supplied test set: 5 instances

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

Time taken to build model: 0 seconds

Predictions on test set:

<table>
<thead>
<tr>
<th>inst#, actual, predicted, error, probability distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1   2:B 2:B   0.059 ±0.941</td>
</tr>
<tr>
<td>2   2:B 2:B   0.059 ±0.941</td>
</tr>
<tr>
<td>3   2:B 2:B   0.059 ±0.941</td>
</tr>
<tr>
<td>4   2:B 2:B   0.059 ±0.941</td>
</tr>
<tr>
<td>5   2:B 2:B   0.059 ±0.941</td>
</tr>
</tbody>
</table>

Evaluation on test set:

- Correctly Classified Instances: 5 100 %
- Incorrectly Classified Instances: 0 0 %
Model evaluation – holdout (percentage split)

Click on More options…

Test mode: split 66% train, remainder test

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

IBk 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 ===

Correctly Classified Instances  2  40  %
Incorrectly Classified Instances  3  60  %
Model evaluation – cross validation

--- Predictions on test data ---

<table>
<thead>
<tr>
<th>inst#,</th>
<th>actual, predicted, error, probability distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2:no 1:yes + *0.962 0.038</td>
</tr>
<tr>
<td>2</td>
<td>1:yes 1:yes *0.5 0.5</td>
</tr>
<tr>
<td>1</td>
<td>2:no 1:yes + *0.962 0.038</td>
</tr>
<tr>
<td>2</td>
<td>1:yes 1:yes *0.962 0.038</td>
</tr>
<tr>
<td>1</td>
<td>2:no 2:no 0.071 *0.929</td>
</tr>
<tr>
<td>2</td>
<td>1:yes 1:yes *0.658 0.342</td>
</tr>
<tr>
<td>1</td>
<td>2:no 1:yes + *0.5 0.5</td>
</tr>
<tr>
<td>2</td>
<td>1:yes 1:yes *0.929 0.071</td>
</tr>
<tr>
<td>1</td>
<td>2:no 2:no 0.342 *0.658</td>
</tr>
<tr>
<td>2</td>
<td>1:yes 1:yes *0.929 0.071</td>
</tr>
<tr>
<td>1</td>
<td>1:yes 1:yes *0.5 0.5</td>
</tr>
<tr>
<td>2</td>
<td>1:yes 2:no + 0.342 *0.658</td>
</tr>
<tr>
<td>1</td>
<td>1:yes 2:no + 0.071 *0.929</td>
</tr>
<tr>
<td>2</td>
<td>1:yes 2:no + 0.071 *0.929</td>
</tr>
</tbody>
</table>

--- Stratified cross-validation ---

--- Summary ---

Correctly Classified Instances 8 57.1429 %
Incorrectly Classified Instances 6 42.8571 %
Model evaluation – leave one out cross validation
Model evaluation – confusion (contingency) matrix

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>a</strong></td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td><strong>b</strong></td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>
Clustering – k-means

Click on Ignore attributes
Hierarchical Clustering – Cobweb
Association Rules (A => B)

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

- Rule 180: humidity=high, windy=FALSE; play=no
- Rule 181: humidity=high, windy=FALSE; play=yes
- Rule 182: humidity=high, play=no; windy=TRUE
- Rule 183: temperature=cool, windy=FALSE; play=TRUE
- Rule 184: temperature=cool, humidity=normal; windy
- Rule 185: temperature=cool, humidity=normal; windy
- Rule 186: humidity=normal, windy=FALSE; temperature
- Rule 187: temperature=cool, humidity=normal; windy
- Rule 188: temperature=mild, humidity=normal; windy
- Rule 189: temperature=mild, play=yes; windy=FALSE
- Rule 190: temperature=mild, play=yes; windy=TRUE; conf:0.5
- Rule 191: temperature=mild, play=yes; humidity=normal; conf:0.5
- Rule 192: temperature=mild, humidity=high; play=no; conf:0.5
- Rule 193: humidity=high, play=no; temperature=mild; conf:0.5
- Rule 194: temperature=mild, humidity=high; play=yes; conf:0.5
- Rule 195: temperature=mild, play=yes; humidity=high; conf:0.5
- Rule 196: temperature=mild, humidity=high, windy=FALSE; conf:0.5
- Rule 197: humidity=high, windy=FALSE; temperature=mild; conf:0.5
- Rule 198: temperature=mild, humidity=high, windy=TRUE; conf:0.5
- Rule 199: temperature=hot, windy=FALSE; play=yes; conf:0.5
- Rule 200: temperature=hot, humidity=high; play=no; conf:0.5
And many more …

Thank you!