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
Agenda

• Data Mining

• Weka Project

• Basic functionality of Weka by example

• Weka for document classification and clustering
# Database management systems (DBMS), Online Analytical Processing (OLAP) and Data Mining

<table>
<thead>
<tr>
<th>Area</th>
<th>DBMS</th>
<th>OLAP</th>
<th>Data Mining</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Extraction of detailed and summary data</td>
<td>Summaries, trends and forecasts</td>
<td>Knowledge discovery of hidden patterns and insights</td>
</tr>
<tr>
<td>Type of result</td>
<td>Information</td>
<td>Analysis</td>
<td>Insight and Prediction</td>
</tr>
<tr>
<td>Method</td>
<td>Deduction (Ask the question, verify with data)</td>
<td>Multidimensional data modeling, Aggregation, Statistics</td>
<td>Induction (Build the model, apply it to new data, get the result)</td>
</tr>
<tr>
<td>Example question</td>
<td>Who purchased mutual funds in the last 3 years?</td>
<td>What is the average income of mutual fund buyers by region by year?</td>
<td>Who will buy a mutual fund in the next 6 months and why?</td>
</tr>
</tbody>
</table>
Example of DBMS, OLAP and Data Mining: Weather data

Assume we have made a record of the weather conditions during a two-week period, along with the decisions of a tennis player whether or not to play tennis on each particular day. Thus we have generated tuples (or examples, instances) consisting of values of four independent variables (outlook, temperature, humidity, windy) and one dependent variable (play).
• What was the temperature in the sunny days?  \{85, 80, 72, 69, 75\}
• Which days the humidity was less than 75?  \{6, 7, 9, 11\}
• Which days the temperature was greater than 70?  \{1, 2, 3, 8, 10, 11, 12, 13, 14\}
• Which days the temperature was greater than 70 and the humidity was less than 75?  
The intersection of the above two:  \{11\}
OLAP: Multidimensional Model (Data Cube)

Dimensions:
- Time: Week 1={1, 2, 3, 4, 5, 6, 7}, Week 2={8, 9, 10, 11, 12, 13, 14}
- Outlook: {sunny, rainy, overcast}

Unit: play (yes/no)

<table>
<thead>
<tr>
<th></th>
<th>sunny</th>
<th>rainy</th>
<th>overcast</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 / 5</td>
<td>0 / 2</td>
<td>2 / 1</td>
<td>2 / 0</td>
</tr>
<tr>
<td>Week 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 2</td>
<td>2 / 1</td>
<td>1 / 1</td>
<td>2 / 0</td>
</tr>
</tbody>
</table>

⇒ if outlook = overcast then play = yes
Discretize numeric attributes (data pre-processing stage in data mining). Group the temperature values in three intervals (hot, mild, cool) and humidity values in two (high, normal).

1. humidity=normal windy=false 4 ==> play=yes (4, 1)
2. temperature=cool 4 ==> humidity=normal (4, 1)
3. outlook=overcast 4 ==> play=yes (4, 1)
4. temperature=cool play=yes 3 ==> humidity=normal (3, 1)
5. outlook=rainy windy=false 3 ==> play=yes (3, 1)
6. outlook=rainy play=yes 3 ==> windy=false (3, 1)
7. outlook=sunny humidity=high 3 ==> play=no (3, 1)
8. outlook=sunny play=no 3 ==> humidity=high (3, 1)
9. temperature=cool windy=false 2 ==> humidity=normal play=yes (2, 1)
10. temperature=cool humidity=normal windy=false 2 ==> play=yes (2, 1)
Data Mining: Decision Tree and Rules

If outlook = overcast then yes
If humidity = normal and windy = false then yes
If temperature = mild and humidity = normal then yes
If outlook = rainy and windy = false then yes
If outlook = sunny and humidity = high then no
If outlook = rainy and windy = true then no
Data Mining: Prediction

\[ P(\text{play=yes} \mid \text{outlook=sunny, temperature=mild, humidity=normal, windy=false}) = 0.8 \]

\[ P(\text{play=no} \mid \text{outlook=sunny, temperature=mild, humidity=normal, windy=false}) = 0.2 \]
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.

Found only on the islands of New Zealand, the Weka is a flightless bird with an inquisitive nature. The name is pronounced like this, and the bird sounds like this.

Weka is open source software issued under the GNU General Public License.

We have put together several free online courses that teach machine learning and data mining using Weka. Check out the website for the courses for details on when and how to enrol. The videos for the courses are available on Youtube.

Yes, it is possible to apply Weka to big data.
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 usable 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.
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.**
Now …
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)
- 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 – CSV format
(LoanData.csv)
Data preprocessing and visualization


Attribute-Relation File Format (ARFF)

April 4th, 2006

This documentation is superseded by the WekaDoc Wiki. Version specific documentation is available there:

- 34k
- 35k

1st April 2002

An ARFF (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes. ARFF files were used in the Machine Learning Project at the Department of Computer Science of The University of Waikato for use with the Weka machine learning software. This document describes the version of ARFF used with Weka versions 3.2 to 3.3; it is an extension of the ARFF format as described in the document written by Ian H. Witten and Eibe Frank (the new additions are string attributes, date attributes, and sparse instances).

This explanation was cobbled together by Gordon Paynter (gordon.paynter@ucr.edu) from the Weka 2.1 ARFF description, email from 1myreadbox.com and Eibe Frank (eibe@cs.waikato.ac.nz), and some datasets. It has been edited by Richard Kirkby (rkirkby@cs.waikato.ac.nz) if you're interested in seeing the ARFF 3 proposal.

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:

```
@relation Iris

@attribute sepal-length NUMERIC
@attribute sepal-width NUMERIC
@attribute petal-length NUMERIC
@attribute petal-width NUMERIC
@attribute class (Iris-setosa, Iris-versicolor, Iris-virginica)

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, 1.12, pc, y, 2, good, y
5, f, n, 50, 5, 4.12, pc, y, 2, good, 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, 2.15, 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

Run Weka and select the Explorer
Data preprocessing and visualization

Load data into Weka – ARFF format or CSV format (click on “Open file…”)
Data preprocessing and visualization

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

[Image of Weka Explorer interface with a file being saved]

[Image of file save dialog box with file path and save button]
Data preprocessing and visualization

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

[Image of Weka software interface showing data preprocessing and visualization options]
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

Click on “Visualize” tab, double-click on a plot to see the 2D projection of the instance space.
Data preprocessing and visualization

Using filters: click on “Choose” in the “Filter” window, select “Discretize”
Data preprocessing and visualization

Click in the “Discretize” in the Filter window and choose parameters, then click on “Apply”

Note how the plot of “lastemp” changed.
Data preprocessing and visualization

Web/Text documents - Department data

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 Off 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: Wehe Hall 101
Phone: 832-2900

Department Website
Data preprocessing and visualization
Department data document collection

http://www.cs.ccsu.edu/~markov/MDLclustering

http://www.cs.ccsu.edu/~markov/MDLclustering/data.zip

MDL Clustering

Algorithms for unsupervised attribute ranking, discretization and clustering available as Java classes through a command-line interface. All Weka classes are also included.

Manual
Executable JAR file
Data
Lab Project

Zdravko Markov
http://www.cs.ccsu.edu/~markov/
Data preprocessing and visualization

Department data: Create ARFF file

http://www.cs.ccsu.edu/~markov/MDLclustering/MDL.jar

MDL Clustering

Algorithms for unsupervised attribute ranking, discretization and clustering available as Java classes through a command-line interface. All Weka classes are also included.

Manual
Executable JAR file
Data
Lab Project

Zdravko Markov
http://www.cs.ccsu.edu/~markov/

Welcome to the WEKA SimpleCLI

This interface provides access to the MDL classes and all WEKA classes. Enter commands in the textfield at the bottom of the window. Use the up and down arrows to move through previous commands. Command must be one of:

- java <classname> <args> [ > file]
- break
- kill
- cls
- history
- exit
- help [<command>]

MDL classes (see http://www.cs.ccsu.edu/~markov/MDLclustering/MDLmanual.pdf):

MDLcluster <input file>.arff [compression cutoff] [de] [<output file>, {arff|csv}]
MDLranker <input file>.arff [num of attributes] [output file], {arff|csv} [de]
MDLdiscretize <input file>.arff <output file>, {arff|csv} [de]
ARFFstring <input directory> <class label> <output file name>

WEKA classes: see http://www.cs.waikato.ac.nz/ml/weka/documentation.html

> java ARFFstring data/departments/A A deptA

java ARFFstring data/departments/A A deptA
Data preprocessing and visualization
Department data: Create ARFF file in string format (using SimpleCLI)

1. Create file deptA with the files in folder data/departments/A with class label A:
   
   ```java
   java ARFFstring data/departments/A A deptA
   ```

2. Create file deptB with the files in folder data/departments/B with class label B:
   
   ```java
   java ARFFstring data/departments/B B deptB
   ```

3. Merge deptA and deptB into one file departments-string.arff

4. Add the following ARFF file header in the beginning of departments-string.arff:
   
   ```arff
   @relation departments_string
   @attribute document_name string
   @attribute document_content string
   @attribute document_class {A,B}
   @data
   ```
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, set attributeRange to 1, click on Apply
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 document_class last attribute
• 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 NumericToNominal filter)
- Save data on a file for further use
Data preprocessing and visualization

ARFF department data in binary format (NonSparse and Sparse format, see SparseToNonSparse filter)
Data preprocessing and visualization

ARFF department data in TF and TFIDF format
Data preprocessing and visualization

Student Projects

- Preprocess.html
- Visualization.html
Attribute Selection

Finding a minimal set of attributes that preserve the class distribution

Attribute relevance with respect to the class – irrelevant attribute (accounting)

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 (\textit{science})

IF science\,=\,1 THEN class\,=\,A (Error\,=\,0, Coverage \,=\, 7 instance)
IF science\,=\,0 THEN class\,=\,B (Error\,=\,4/13, Coverage \,=\, 13 instances)
Attribute Selection (with document_name)
Attribute Selection (without document_name)

Select document_name and click on Remove
Attribute Selection (ranking)
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)

Choose filters/supervised/attribute/AttributeSelection and use CfsSubsetEval and BestFirst search. Then click on Visualize All.
Attribute Selection

Student Projects

• Attribute Selection.html
Classification – creating models (hypotheses)

Mapping (independent attributes -> class)
Classification – creating models (hypotheses)

Inferring one-attribute 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, overcast -&gt; yes, rainy -&gt; yes</td>
<td>2/5, 0/4, 2/5</td>
<td>4/14</td>
</tr>
<tr>
<td>temperature</td>
<td>hot -&gt; no, mild -&gt; yes, cool -&gt; yes</td>
<td>2/4, 2/6, 1/4</td>
<td>5/14</td>
</tr>
<tr>
<td>humidity</td>
<td>high -&gt; no, normal -&gt; yes</td>
<td>3/7, 1/7</td>
<td>4/14</td>
</tr>
<tr>
<td>windy</td>
<td>false -&gt; yes, true -&gt; no</td>
<td>2/8, 3/5</td>
<td>5/14</td>
</tr>
</tbody>
</table>
Classification – OneR
Classification – decision tree

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

Top-down induction of decision trees (TDIDT, old approach known 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
Classification – predicting class

Click on Set…

Click on Open file…

Test Instances

Relation: weather...
Attributes: 5
Instances: ?
Sum of weights: ?

Open file…
Open URL…

Preprocess Classify
Classifier
Choose J48 - C 0.25 -

Test options
Use training set
Supplied test set
Cross-validation
Percentage split

Class (Nom) play

More split
%
66

Classifier output

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

=== Predictions on test set ===

inst# actual predicted error prediction
1 1:yes 1 1

weather.nominal.test.arff - Notepad

@relation weather.nominal.test

@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, normal}
@attribute play {yes, no}

@data
sunny,mild,normal,FALSE,?
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
Classification

Student Projects

• Classification.html
Prediction (no model, lazy learning)

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

- **K-nearest neighbor (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

<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

Student Projects

• Prediction.html
Model evaluation – holdout (percentage split)

<table>
<thead>
<tr>
<th>inst#</th>
<th>actual</th>
<th>predicted</th>
<th>error</th>
<th>prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1:y</td>
<td>2:n</td>
<td>+ 1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1:y</td>
<td>2:n</td>
<td>+ 1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2:n</td>
<td>2:n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1:y</td>
<td>2:n</td>
<td>+ 1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1:y</td>
<td>2:n</td>
<td>+ 1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1:y</td>
<td>2:n</td>
<td>+ 1</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2:n</td>
<td>2:n</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

--- Evaluation on test split ---

Time taken to test model on test split: 0 seconds

--- Summary ---

Correctly Classified Instances 2 28.5714 %
Incorrectly Classified Instances 5 71.4286 %
Total Number of Instances 7

--- Detailed Accuracy By Class ---

<table>
<thead>
<tr>
<th></th>
<th>TP Rate</th>
<th>FP Rate</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>emp = y</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>y</td>
</tr>
<tr>
<td>buy = pc: y</td>
<td>1.000</td>
<td>1.000</td>
<td>0.286</td>
<td>1.000</td>
<td>0.444</td>
<td>n</td>
</tr>
<tr>
<td>money &lt; 50: n</td>
<td>0.286</td>
<td>0.286</td>
<td>0.082</td>
<td>0.286</td>
<td>0.127</td>
<td></td>
</tr>
</tbody>
</table>

--- Confusion Matrix ---

a b
0 5 | a = y
0 2 | b = n
Model evaluation – cross validation
Model evaluation – leave one out cross validation
Model evaluation – confusion (contingency) matrix

<table>
<thead>
<tr>
<th>predicted</th>
<th>actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>yes</td>
<td>3</td>
</tr>
<tr>
<td>no</td>
<td>1</td>
</tr>
</tbody>
</table>

**Confusion Matrix**

- TP (True Positive): 3
- FP (False Positive): 1
- TN (True Negative): 0
- FN (False Negative): 1

**Evaluation metrics**

- **Precision** = TP / (TP + FP)
- **Recall** = TP / (TP + FN)

**Classifier Output**

- **Correctly Classified Instances**: 3 (60%)
- **Incorrectly Classified Instances**: 2 (40%)
- **Total Number of Instances**: 5
Model evaluation

Student Projects

• Evaluation.html
Clustering – k-means

Click on Ignore attributes
Clustering – classes to clusters evaluation

Right click on Result list, select Visualize cluster assignments

Click on Save
Clustering

Student Projects

• Clustering.html
Association Rules (A => B)

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

Student Projects

• Association.html
Document classification and clustering

Predict the class of the Theatre document

1. Create a training set – all departments excluding Theatre (data collection)
2. Use Binary, Term Frequency or TFIDF representation (data preprocessing)
3. Select a relevant subset of attributes (attribute selection)
4. Use J48, IBk, and Naïve Bayes (classification)
5. Evaluate all models by cross validation (model evaluation)
6. Choose the best model and predict the class of Theatre (prediction)
7. Cluster the training set with K-means compare the cluster centroids with Theatre
Document classification and clustering

Teaching resources and student projects based on Weka

- Zdravko Markov and Daniel T. Larose, *Data Mining the Web: Uncovering Patterns in Web Content, Structure, and Usage*, *Wiley* 2007 (free excerpts: Chapter 1, TOC, Index)
- Data sets: [http://www.cs.ccsu.edu/~markov/dmwdata.zip](http://www.cs.ccsu.edu/~markov/dmwdata.zip)
- [Clustering.html](http://www.cs.ccsu.edu/~markov/MDLclustering/)

[http://www.cs.ccsu.edu/~markov/DMWprojects](http://www.cs.ccsu.edu/~markov/DMWprojects)