Data warehouse& Data Mining
UNIT-3

Syllabus

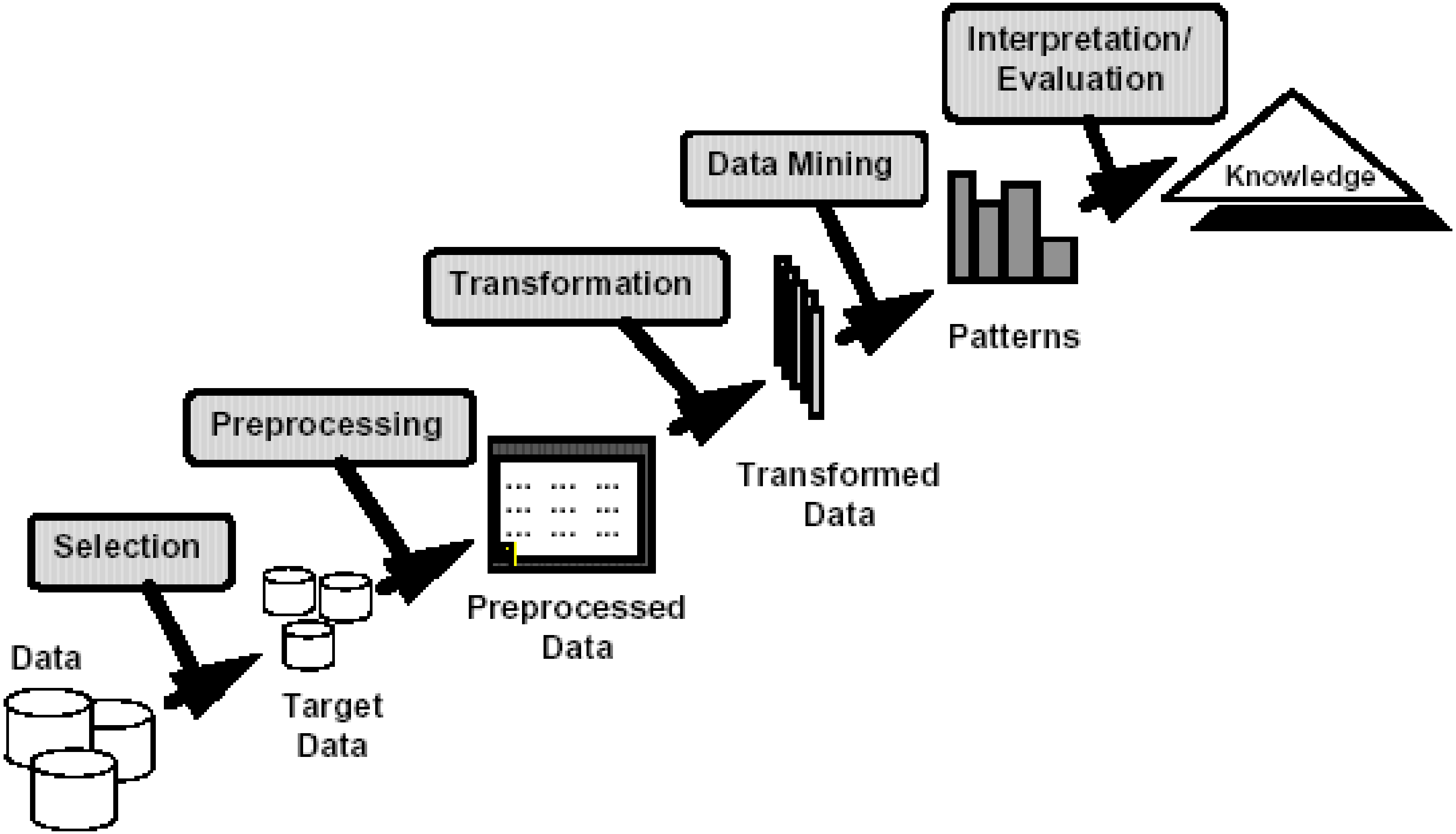
- **UNIT 3**
- **Classification:** Introduction, decision tree, tree induction algorithm – split algorithm based on information theory, split algorithm based on Gini index; naïve Bayes method; estimating predictive accuracy of classification method; classification software, software for association rule mining; case study; KDD Insurance Risk Assessment

What is Data Mining?

- Data Mining is:
  1. (1) The efficient discovery of previously unknown, valid, potentially useful, understandable patterns in large datasets
  2. **(2) Data mining is the analysis step of the "knowledge discovery in databases" process, or KDD**

(2) The analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner

Knowledge Discovery



Examples of Large Datasets

- Government: IRS, NGA, …
- Large corporations
  - WALMART: 20M transactions per day
  - MOBIL: 100 TB geological databases
  - AT&T 300 M calls per day
  - Credit card companies
- Scientific
  - NASA, EOS project: 50 GB per hour
  - Environmental datasets

KDD

The Knowledge Discovery in Databases (KDD) process is commonly defined with the stages:

(1) Selection
(2) Pre-processing
(3) Transformation
(4) Data Mining
(5) Interpretation/Evaluation

Data Mining Methods

1. Decision Tree Classifiers:

   Used for modeling, classification

2. Association Rules:

   Used to find associations between sets of attributes

3. Sequential patterns:

   Used to find temporal associations in time series

4. Hierarchical clustering:

   used to group customers, web users, etc

Why Data Preprocessing?

- Data in the real world is dirty
  - incomplete: lacking *attribute values*, lacking certain *attributes of interest*, or containing only aggregate data
  - noisy: containing errors or outliers
  - inconsistent: containing discrepancies in codes or names
- No quality data, no quality mining results!
  - Quality decisions must be based on quality data
  - Data warehouse needs consistent integration of quality data
  - Required for both OLAP and Data Mining!

Classification (Supervised learning): Form of data analysis

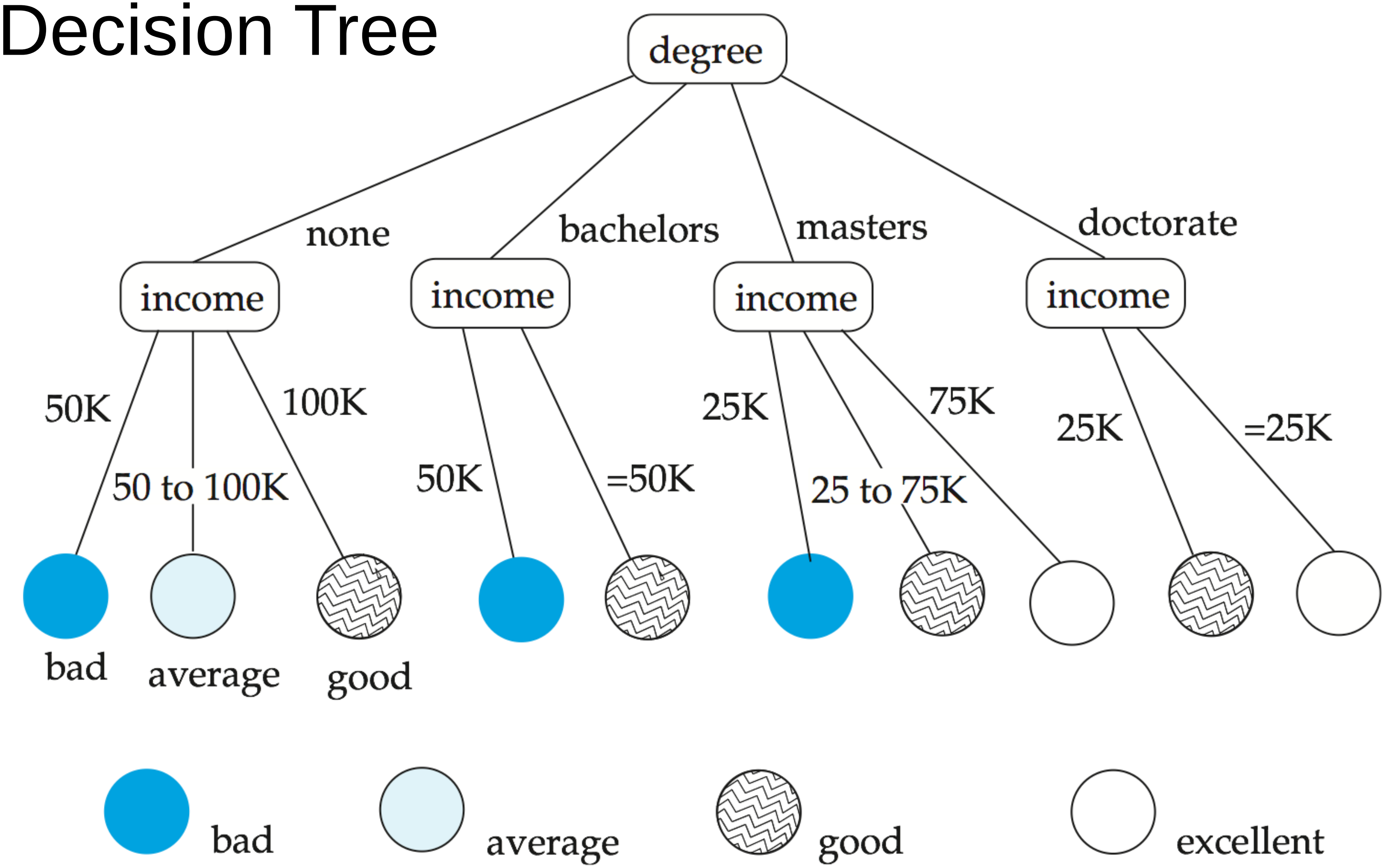
Classification: Definition

- Given a collection of records (*training set*)
  - Each record contains a set of *attributes*, one of the attributes is the *class*.
- Find a *model* for class attribute as a function of the values of other attributes.
- Goal: previously unseen records should be assigned a class as accurately as possible.
  - A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

Classification Rules

- Classification rules help assign new objects to classes.
- E.g. to categorize bank loan applications as either safe or risky.
  - E.g.2, **given a new automobile insurance applicant, should he or she be classified as low risk, medium risk or high risk?**
- Classification rules for above example could use a variety of data, such as educational level, salary, age, etc.
  - $\forall$ person P, P.degree = masters **and** P.income > 75,000
    $\Rightarrow$ P.credit = excellent
  - $\forall$ person P, P.degree = bachelors **and**
    (P.income $\geq$ 25,000 and P.income $\leq$ 75,000)
    $\Rightarrow$ P.credit = good
- Rules are not necessarily exact: there may be some misclassifications.
- Classification is a two step process:1. learning step (predetermined set).2 . Classification step
- Classification rules can be shown compactly as a decision tree**.**

Decision Tree



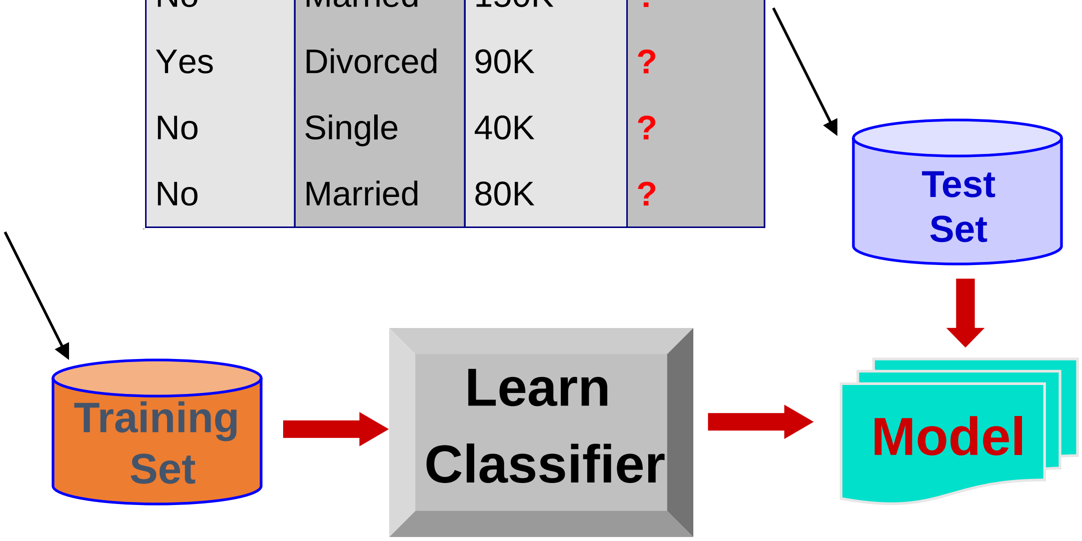
Construction of Decision Trees

- **Training set**: a data sample in which the classification is already known.
- **Greedy** top down generation of decision trees.
  - Each internal node of the tree partitions the data into groups based on a **partitioning attribute**, and a **partitioning condition** for the node
  - **Leaf** node:
    - all (or most) of the items at the node belong to the same class, or
    - all attributes have been considered, and no further partitioning is possible.

Classification Example

	categorical	categorical	continuous	class
Tid	**Home Owner**	**Marital Status**	**Taxable Income**	**Default**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

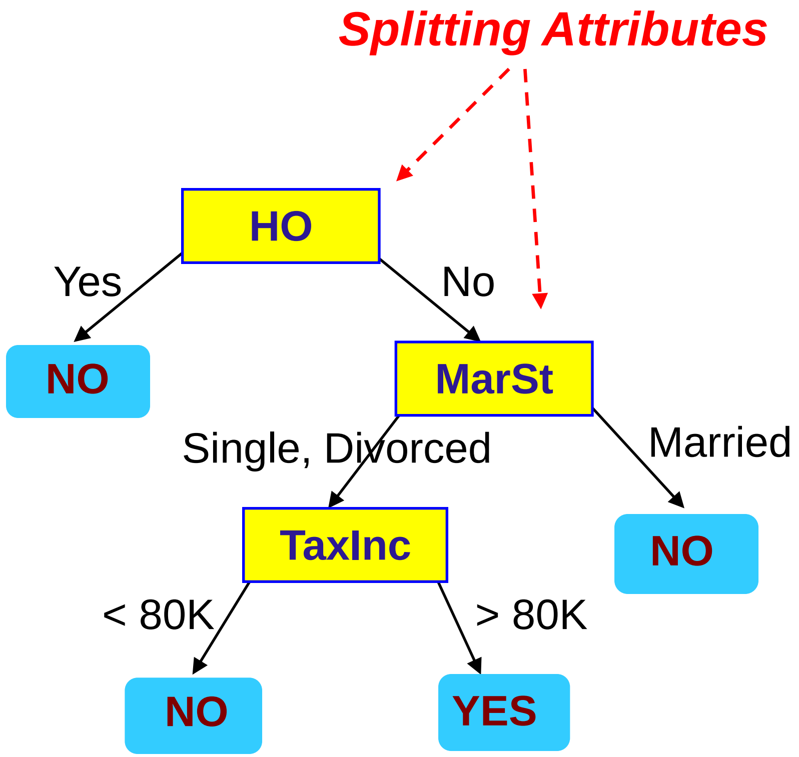
Home Owner	**Marital Status**	**Taxable Income**	**Default**
No	Single	75K	**?**
Yes	Married	50K	**?**
No	Married	150K	**?**
Yes	Divorced	90K	**?**
No	Single	40K	**?**
No	Married	80K	**?**



Example of a Decision Tree

	categorical	categorical	continuous	class
Tid	**Home Owner**	**Marital Status**	**Taxable Income**	**Default**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

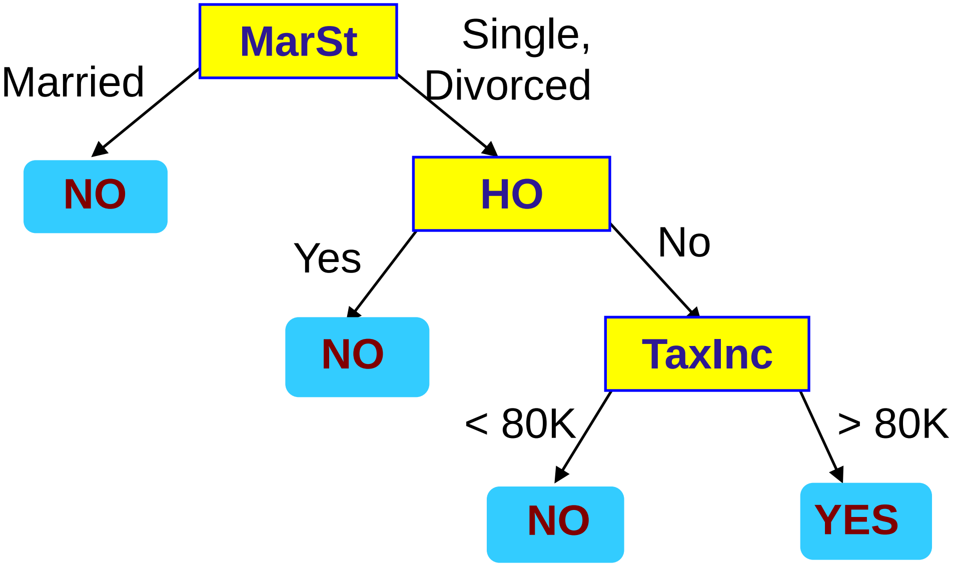
Training Data



Model: Decision Tree

Another Example of Decision Tree

	categorical	categorical	continuous	class
Tid	**Home Owner**	**Marital Status**	**Taxable Income**	**Default**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**



There could be more than one tree that fits the same data!

Decision tree classifiers

- Widely used learning method
- Easy to interpret: can be re-represented as if-then-else rules
- Approximates function by piece wise constant regions
- Does not require any prior knowledge of data distribution, works well on noisy data.
- Has been applied to:
  - classify medical patients based on the disease,
  - equipment malfunction by cause,
  - loan applicant by likelihood of payment.

Tree induction

- Greedy strategy
  - Split the records at each node based on an attribute test that optimizes some chosen criterion.

- Issues
  - Determine how to split the records
    - How to specify structure of split?
    - What is best attribute / attribute value for splitting?
  - Determine when to stop splitting

Tree induction

- Greedy strategy
  - Split the records at each node based on an attribute test that optimizes some chosen criterion.

- Issues
  - Determine how to split the records
    - How to specify structure of split?
    - What is best attribute / attribute value for splitting?
  - Determine when to stop splitting

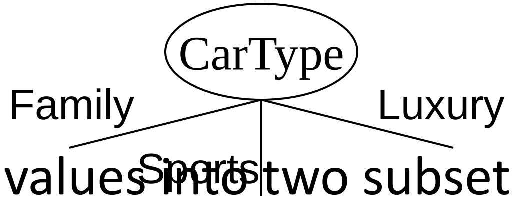
Specifying structure of split

- Depends on attribute type
  - Nominal
  - Ordinal

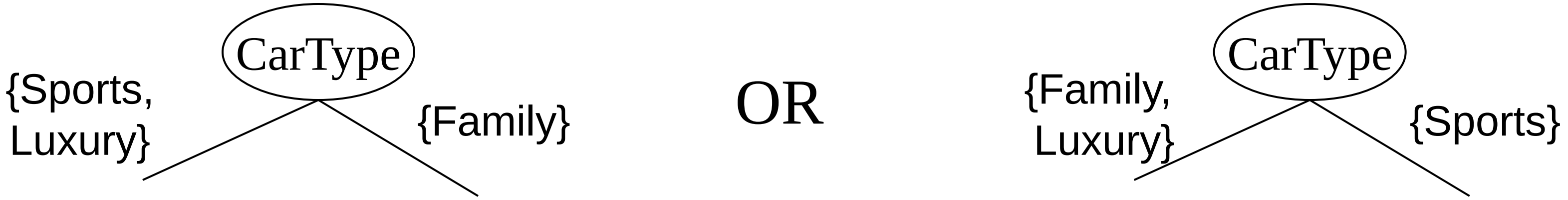
- Depends on number of ways to split
  - Binary (two-way) split
  - Multi-way split

Splitting based on nominal attributes

- Multi-way split: Use as many partitions as distinct values.



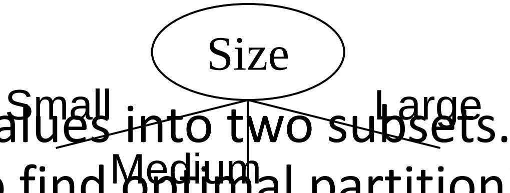
- Binary split: Divides values into two subsets. Need to find optimal partitioning.



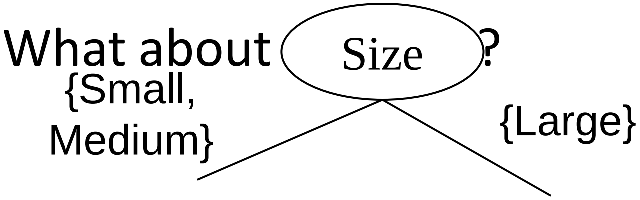
Splitting based on ordinal attributes

- Multi-way split: Use as many partitions as distinct values.

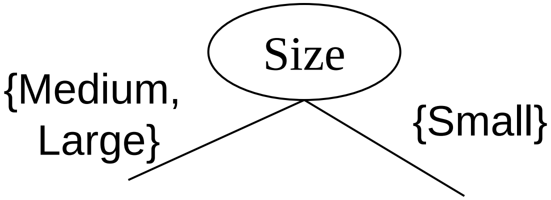
- Binary split: Divides values into two subsets.
  Need to find optimal partitioning.

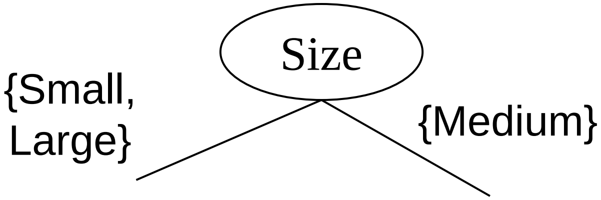


- What about ?



OR





Splitting Criteria: Gini Index

- If a data set T contains examples from n classes, gini index, gini(T) is defined as

$$gini(T) = 1 - \sum_{j=1}^{n} p_j^2$$

where p_j is the relative frequency of class j in T.

gini(T) is minimized if the classes in T are skewed.

The Gini coefficient measures the inequality among values of a frequency distribution (for example levels of income).

*Gini Index

After splitting T into two subsets T1 and T2 with sizes N1 and N2, the gini index of the split data is defined as

$$gini_{split}(T) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)$$

- The attribute providing smallest $\text{gini}_{\text{split}}(T)$ is chosen to split the node.

Pros and Cons of decision trees

· **Pros**

+ Reasonable training time

+ Fast application

+ Easy to interpret

+ Easy to implement

+ Can handle large number of features

· **Cons**

- Cannot handle complicated relationship between features

- simple decision boundaries

- problems with lots of missing data

Association rule learning

- **It is a method for discovering interesting relations between variables in large databases**. It is intended to identify strong rules discovered in databases using some measures of interestingness. Based on the concept of strong rules, Rakesh Agrawal et al. introduced association rules for discovering regularities between products in large-scale transaction data recorded by point-of-sale (POS) systems in supermarkets.
- For example, {onion, potato}….> Burger found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, they are likely to also buy hamburger meat. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements. In addition to the above example from market basket analysis association rules are employed today in many application areas including Web usage mining, intrusion detection, Continuous production, and bioinformatics..

Association rules

- Given set T of groups of items
- Example: set of item sets purchased
- Goal: find all rules on itemsets of the form a-->b such that
  - support of a and b > user threshold s
  - conditional probability (confidence) of b given a > user threshold c
- Example: Milk --> bread
- Purchase of product A --> service B

T
Milk, cereal
Tea, milk
Tea, rice, bread
cereal

Association Rule Discovery: Application 1

- Marketing and Sales Promotion:
  - Let the rule discovered be

    {Bagels, … } --> {Potato Chips}
  - Potato Chips as consequent => Can be used to determine what should be done to boost its sales.
  - Bagels in the antecedent => Can be used to see which products would be affected if the store discontinues selling bagels.
  - Bagels in antecedent *and* Potato chips in consequent => Can be used to see what products should be sold with Bagels to promote sale of Potato chips!

Association Rules (Cont.)

- Rules have an associated support, as well as an associated confidence.
- **Support** is a measure of what fraction of the population satisfies both the antecedent and the consequent of the rule.
  - E.g., suppose only 0.001 percent of all purchases include milk and screwdrivers. The support for the rule is *milk* ⇒ *screwdrivers* is low.
- **Confidence** is a measure of how often the consequent is true when the antecedent is true.
  - E.g., the rule *bread* ⇒ *milk* has a confidence of 80 percent if 80 percent of the purchases that include bread also include milk.

- CLASSIFICATION SOFTWARE

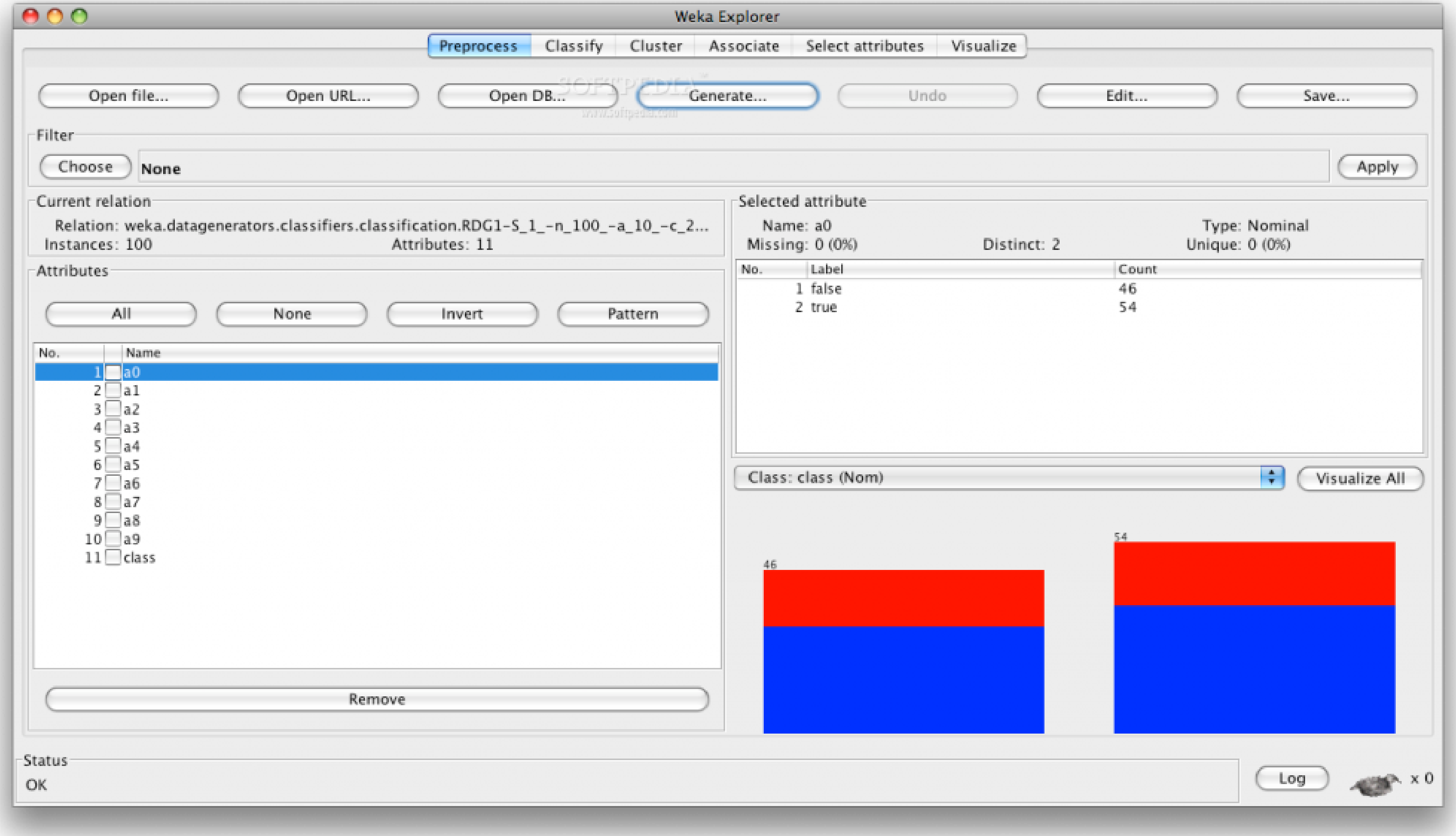
RapidMiner

- **Written in the Java Programming language, this tool offers advanced analytics through template-based** frameworks. A bonus: Users hardly have to write any code. Offered as a service, rather than a piece of local software, this tool holds top position on the list of data mining tools.
- In addition to data mining, RapidMiner also provides functionality like data preprocessing and visualization, predictive analytics and statistical modeling, evaluation, and deployment. What makes it even more powerful is that it provides learning schemes, models and algorithms from WEKA and R scripts.

WEKA

- The original non-Java version of WEKA primarily was developed for analyzing data from the agricultural domain. With the Java-based version, the tool is very sophisticated and used in many different applications including visualization and algorithms for data analysis and predictive modeling. Its free under the GNU General Public License, which is a big plus compared to RapidMiner, because users can customize it however they please.
- **WEKA supports several standard data mining tasks, including data preprocessing, clustering, classification, regression, visualization and feature selection**.
  WEKA would be more powerful with the addition of sequence modeling, which currently is not included.

WEKA



Filter: None

Relation: weka.datagenerators.classifiers.classification.RDG1-S_1_-n_100_-a_10_-c_2...
Instances: 100 Attributes: 11

Name: a0 Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count
1	false	46
2	true	54

No.		Name
1	□	a0
2	□	a1
3	□	a2
4	□	a3
5	□	a4
6	□	a5
7	□	a6
8	□	a7
9	□	a8
10	□	a9
11	□	class

Class: class (Nom)

OK

R-Programming

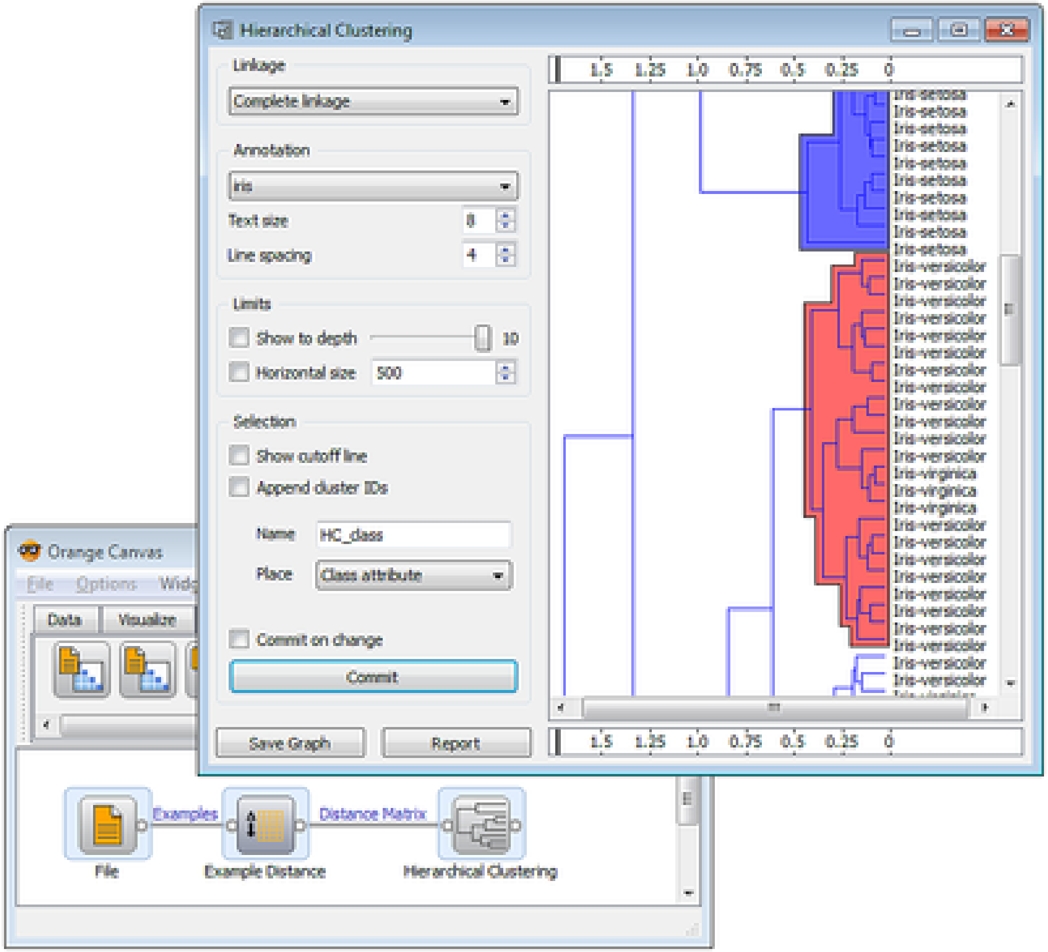
- What if I tell you that Project R, a GNU project, is written in R itself? It’s primarily written in C and Fortran. And a lot of its modules are written in R itself. It’s a free software programming language and software environment for statistical computing and graphics. The R language is widely used among data miners for developing statistical software and data analysis. Ease of use and extensibility has raised R’s popularity substantially in recent years.

- Besides data mining it provides statistical and graphical techniques, including linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, and others.

Orange

- **Python is picking up in popularity because it’s simple and easy to learn yet powerful**. Hence, when it comes to looking for a tool for your work and you are a Python developer, look no further than Orange, a Python-based, powerful and open source tool for both novices and experts.
-

Orange



Hierarchical Clustering

Linkage

Complete linkage

Annotation

iris

Text size 8

Line spacing 4

Limits

☐ Show to depth 10

☐ Horizontal size 500

Selection

☐ Show cutoff line

☐ Append cluster IDs

Name HC_class

Place Class attribute

☐ Commit on change

Commit

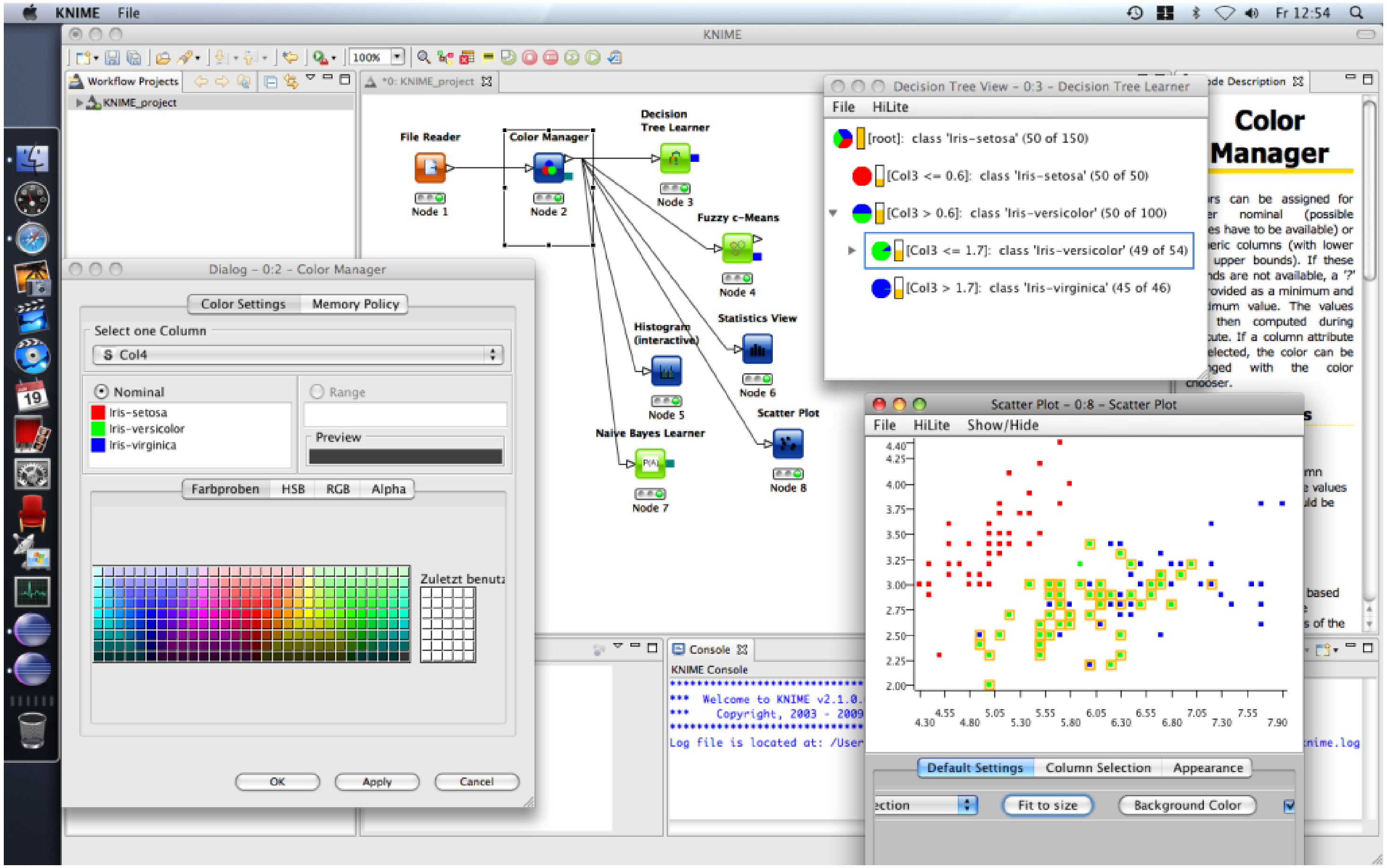
Save Graph

Report

KNIME

- Data preprocessing has three main components: extraction, transformation and loading. KNIME does all three. It gives you a graphical user interface to allow for the assembly of nodes for data processing. It is an open source data analytics, reporting and integration platform. KNIME also integrates various components for machine learning and data mining through its modular data pipelining concept and has caught the eye of business intelligence and financial data analysis.
- Written in Java and based on Eclipse, KNIME is easy to extend and to add plugins. Additional functionalities can be added on the go. Plenty of data integration modules are already included in the core version.

KNIME



Select one Column: S Col4

● Nominal

- 🟥 Iris-setosa
- 🟩 Iris-versicolor
- 🟦 Iris-virginica

○ Range

Preview

[root]: class 'Iris-setosa' (50 of 150)

[Col3 <= 0.6]: class 'Iris-setosa' (50 of 50)

▼ [Col3 > 0.6]: class 'Iris-versicolor' (50 of 100)

▶ [Col3 <= 1.7]: class 'Iris-versicolor' (49 of 54)

[Col3 > 1.7]: class 'Iris-virginica' (45 of 46)

rs can be assigned for
er nominal (possible
es have to be available) or
eric columns (with lower
upper bounds). If these
nds are not available, a '?'
rovided as a minimum and
imum value. The values
then computed during
ute. If a column attribute
elected, the color can be
nged with the color
chooser.

mn
e values
uld be

based
e
s of the

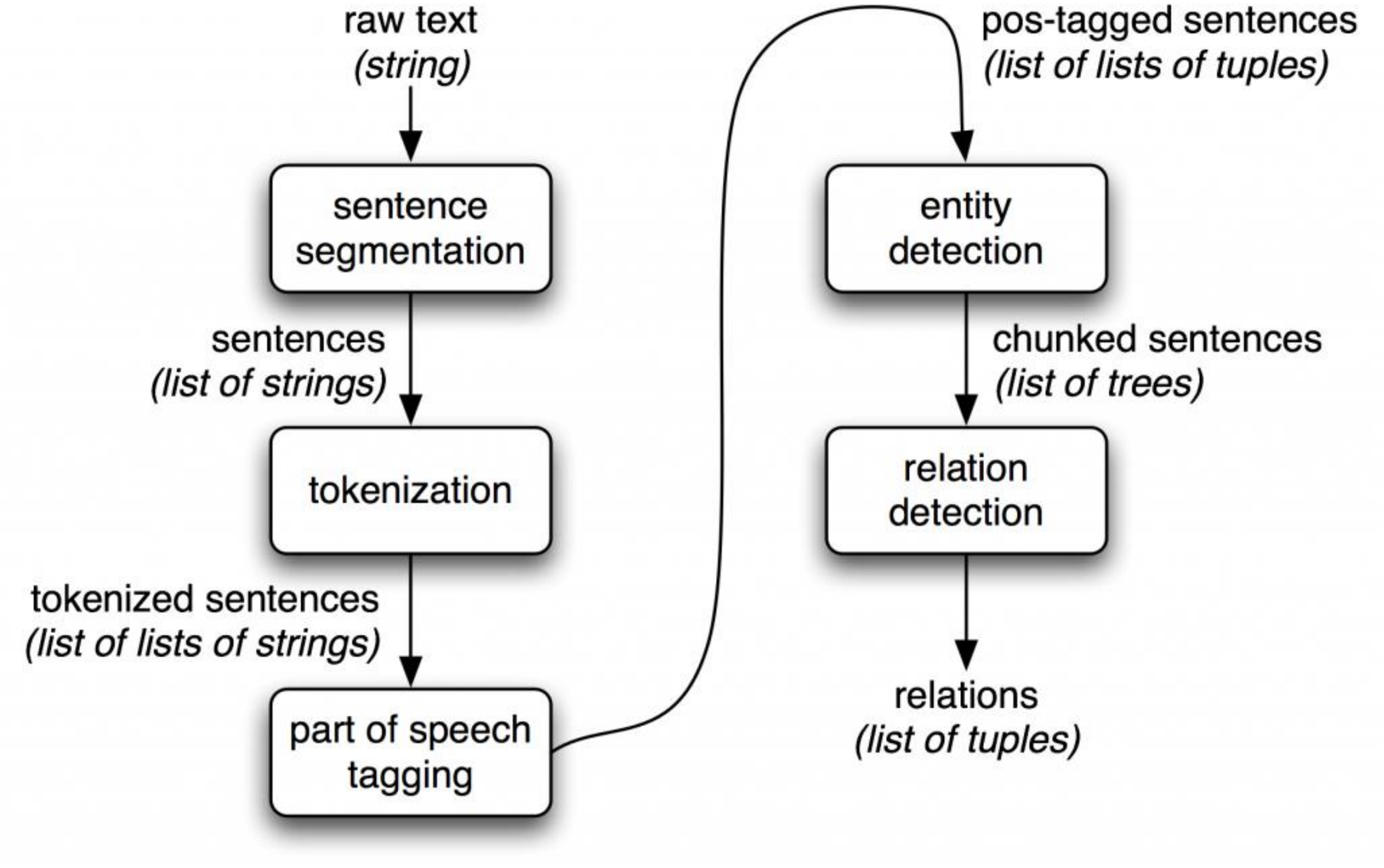
```
KNIME Console
******************************
***  Welcome to KNIME v2.1.0.
***    Copyright, 2003 - 2009
******************************
Log file is located at: /User
```

```
nime.log
```

NLTK

- When it comes to language processing tasks, nothing can beat NLTK. NLTK provides a pool of language processing tools including data mining, machine learning, data scraping, sentiment analysis and other various language processing tasks. All you need to do is install NLTK, pull a package for your favorite task and you are ready to go. Because it’s written in Python, you can build applications on top if it, customizing it for small tasks.
- .

NLTK


raw text
(string)
sentence segmentation
sentences
(list of strings)
tokenization
tokenized sentences
(list of lists of strings)
part of speech tagging
pos-tagged sentences
(list of lists of tuples)
entity detection
chunked sentences
(list of trees)
relation detection
relations
(list of tuples)

The following applications are available under free/open source licenses.

- Carrot2: Text and search results clustering framework.
- Chemicalize.org: A chemical structure miner and web search engine.
- ELKI: A university research project with advanced cluster analysis and outlier detection methods written in the Java language.
- GATE: a natural language processing and language engineering tool.

- KNIME: The Konstanz Information Miner, a user friendly and comprehensive data analytics framework.
- Massive Online Analysis (MOA): a real-time big data stream mining with concept drift tool in the Java programming language.

- ML-Flex: A software package that enables users to integrate with third-party machine-learning packages written in any programming language, execute classification analyses in parallel across multiple computing nodes, and produce HTML reports of classification results.
- MLPACK library: a collection of ready-to-use machine learning algorithms written in the C++ language.
- NLTK (Natural Language Toolkit): A suite of libraries and programs for symbolic and statistical natural language processing (NLP) for the Python language.
- OpenNN: Open neural networks library.
- Orange: A component-based data mining and machine learning software suite written in the Python language.
- R: A programming language and software environment for statistical computing, data mining, and graphics. It is part of the GNU Project.
- scikit-learn is an open source machine learning library for the Python programming language

- Torch: An open source deep learning library for the Lua programming language and scientific computing framework with wide support for machine learning algorithms.
- UIMA: The UIMA (Unstructured Information Management Architecture) is a component framework for analyzing unstructured content such as text, audio and video – originally developed by IBM.
- Weka: A suite of machine learning software applications written in the Java programming language.

- Software for association rule mining

Associations Software: commercial

- Azmy SuperQuery, includes association rule finder.
- IBM SPSS Modeler Suite, includes market basket analysis.
- LPA Data Mining Toolkit supports the discovery of association rules within relational database.
- Magnum Opus, flexible tool for finding associations in data, including statistical support for avoiding spurious discoveries.
- Megaputer Polyanalyst Suite, includes market basket analysis engine
- SmartBundle, a market basket analysis tool for develop profitable retail product bundles and promotions. (30-day free trial)
- Wizsoft WizRule: finds association rules and potential data errors; WizWhy uses association rules for data mining.
- Xpertrule Miner 4.0
- XAffinity(TM), for identifying affinities or patterns in transaction and click stream data

Associations Software: free

arules, a free R extension package which provides the infrastructure for representing, manipulating and analyzing transaction data association rules.

- Apriori, a program to find association rules with the apriori algorithm (Agrawal et al.). Fast implementation that uses prefix trees.
- Apriori, FP-growth, Eclat and DIC implementations by Bart Goethals.
- ARtool, collection of algorithms and tools for the mining of association rules in binary databases. Distributed under the GNU General Public License.
- DM-II system, includes CBA for classification based on associations, and many more features.
- FIMI, Frequent Itemset Mining Implementations repository, including software and datasets.
- Magnum Opus Demo, highly-functional demo software for finding associations in data, including statistical support for avoiding spurious discoveries.