

BUSINESS INTELLIGENCE LABORATORY

Reminds on Data Mining

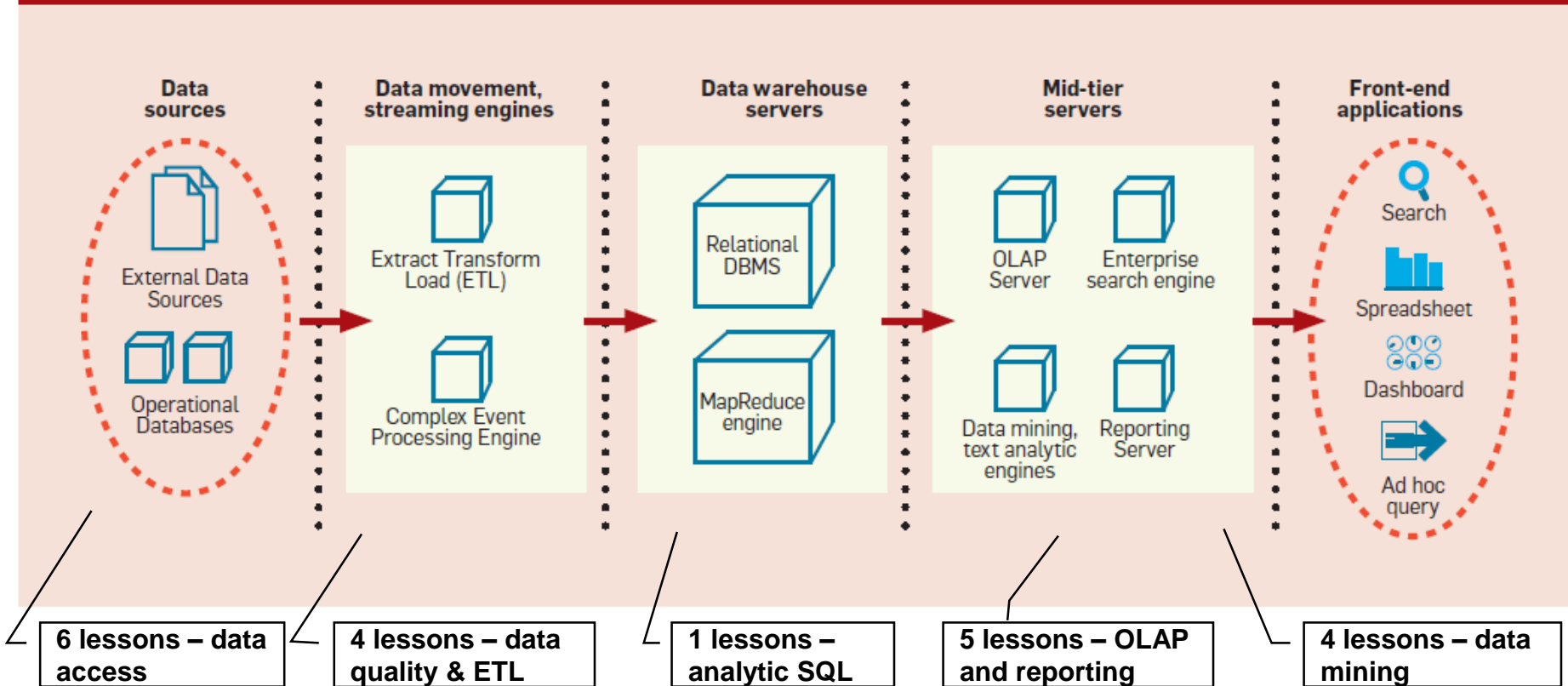
Salvatore Ruggieri & Anna Monreale

Computer Science Department, University of Pisa

BI Architecture

2

Figure 1. Typical business intelligence architecture.



Data Mining Techniques

3

Preprocessing and
visualization

- Classification/Regression
- Association Rule Discovery
- Clustering
- Sequential Pattern Discovery
- Deviation Detection
- Text Mining
- Web Mining
- Social Network Analysis
- ...

Tools for data mining

4

- From **DBMS**
 - SQL Server Analysis Services
 - Oracle Data Miner
 - IBM DB2 Intelligent Miner (discontinued)
- From **Statistical analysis**
 - IBM Modeler (formerly SPSS Clementine)
 - SAS Miner
- From **Machine-Learning**
 - Knime
 - Weka
- An updated list
 - <http://www.kdnuggets.com/software/index.html>

Standards

5

- XML representation of data mining models
 - ▣ Predictive Modelling Markup Language: [PMML](#)

- API for accessing data mining services
 - ▣ Microsoft [OLE DB for DM](#)
 - ▣ Java [JDM](#)

- SQL Extensions for data mining
 - ▣ Standard SQL/MM Part 6 Data Mining
 - ▣ Oracle, DB2 & SQL Server have non-standard extensions
 - SSAS [DMX](#) query language and [Data Mining queries](#)

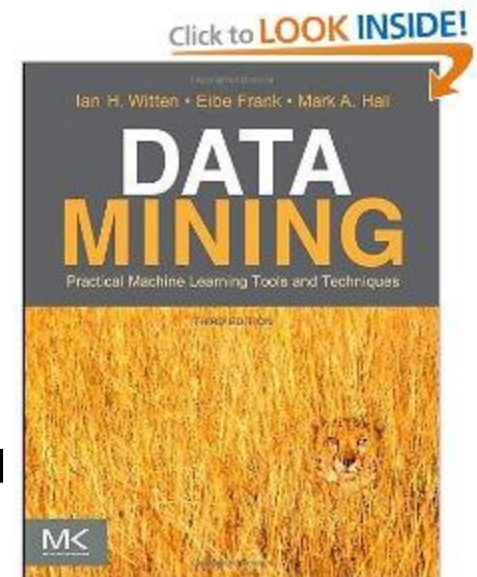
6

Weka

Weka

7

- Suite for machine learning / data mining
- Developed in Java
 - ▣ Distributed with a GNU GPL licence
 - ▣ Since 2006 it is part of the BI Pentaho suite
- References
 - ▣ “Data Mining” by Witten & Frank, 3rd ed., 2011
 - ▣ On line docs
<http://www.cs.waikato.ac.nz/ml/weka/index.html>
- Features / limits:
 - ▣ A complete set of tools for pre-processing, classification, clustering, association rules, visualization
 - ▣ Extensible (documented APIs)
 - ▣ Not very efficient / scalable (data are maintained in main memory)



Weka versions

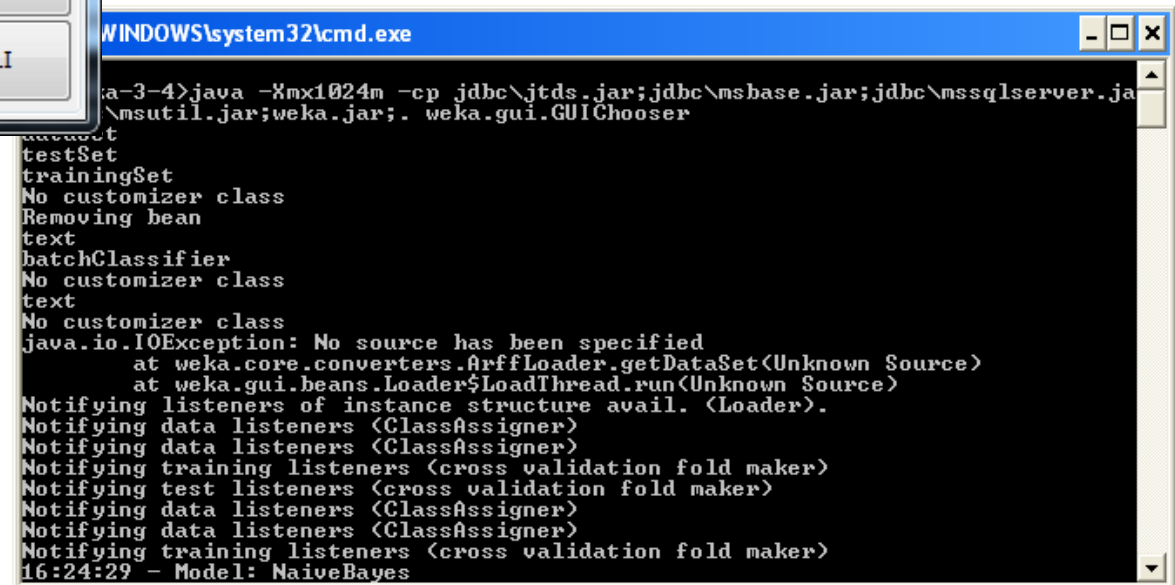
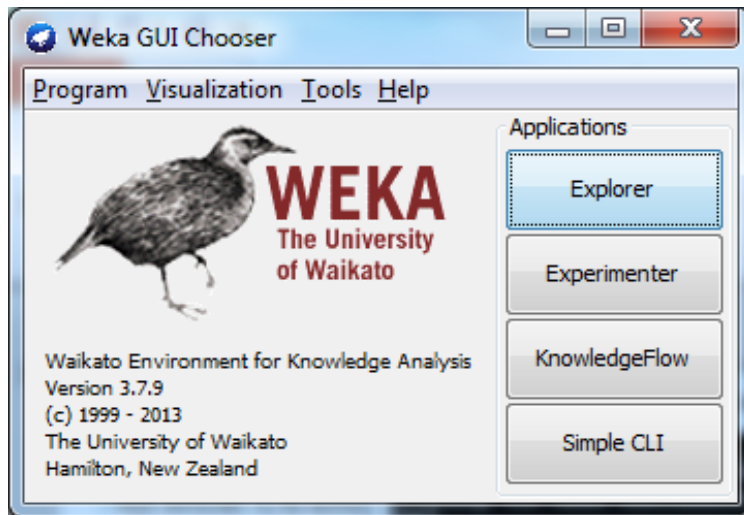
8

- Download: <http://www.cs.waikato.ac.nz/~ml/weka>
- May 2015, Weka 3.7.12 (developer version)
- Patch distributed by the teacher
 - ▣ To be copied in the Weka installation directory
 - ▣ It includes setting for:
 - Larger memory occupation (Java default is 80Mb)
 - Data types for SQL Server RDBMS
 - Driver JDBC
- Weka Light
 - ▣ Minimal version 3.7.12, patch already included

Weka interfaces

9

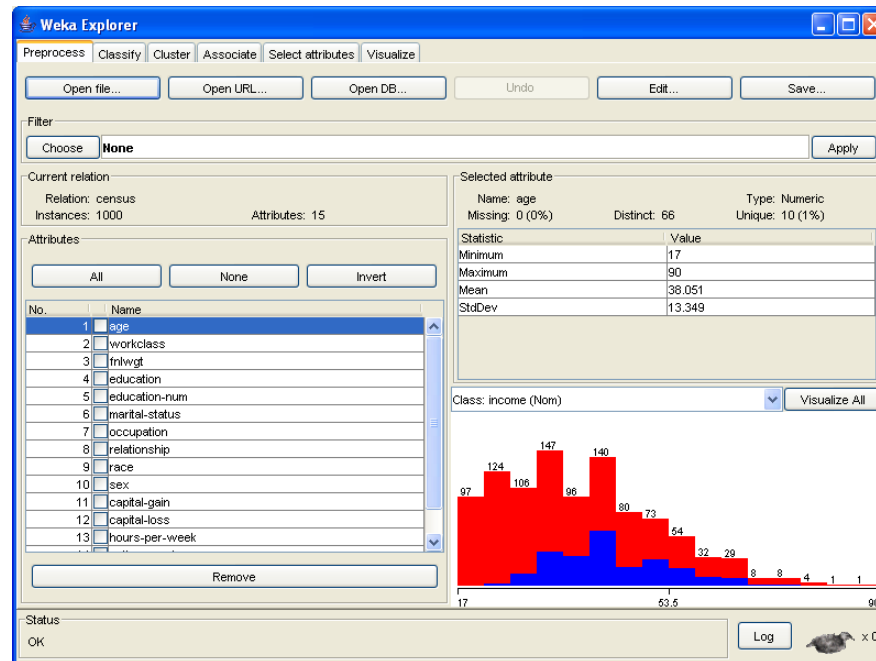
- GUI chooser and console with errors/warnings



The screenshot shows a Windows command prompt window titled 'WINDOWS\system32\cmd.exe'. The command entered is: `java -Xmx1024m -cp jdbc\jtds.jar;jdbc\msbase.jar;jdbc\mssqlserver.jar;msutil.jar;weka.jar;. weka.gui.GUIChooser`. The output shows various progress messages and a final error: `java.io.IOException: No source has been specified`. The error stack trace includes: `at weka.core.converters.ArffLoader.getDataSet(Unknown Source)` and `at weka.gui.beans.Loader$LoadThread.run(Unknown Source)`. The output ends with: `16:24:29 - Model: NaiveBayes`.

Weka interfaces: Explorer

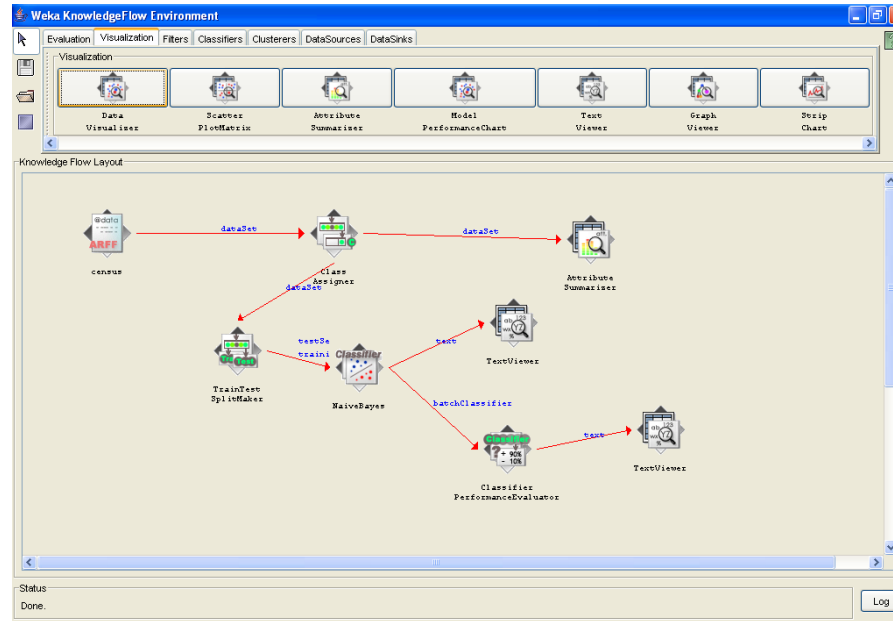
10



Explorer: GUI with distinct panels for pre-processing, classification, clustering, ...

Weka interfaces: KnowledgeFlow

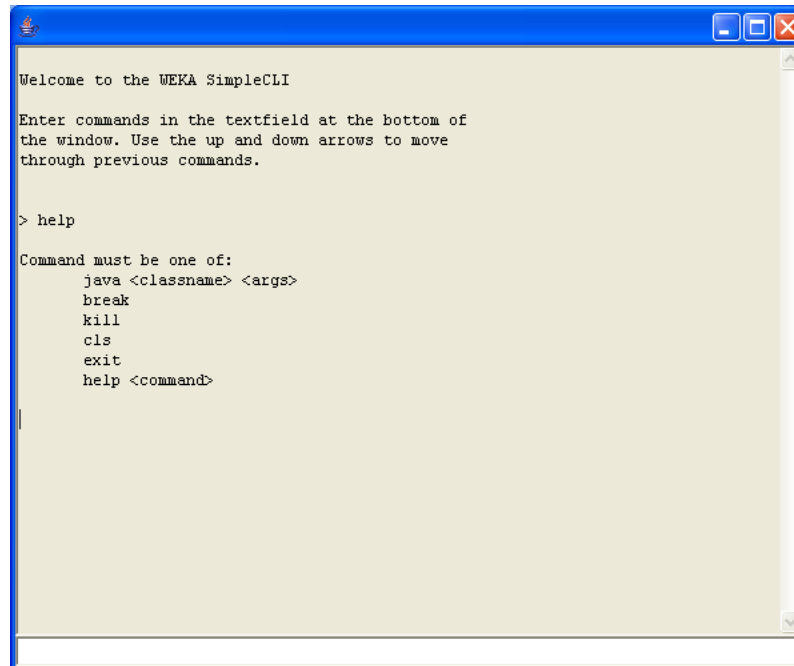
11



KnowledgeFlow: GUI with data flow

Weka interfaces: Simple CLI

12



```
Welcome to the WEKA SimpleCLI

Enter commands in the textfield at the bottom of
the window. Use the up and down arrows to move
through previous commands.

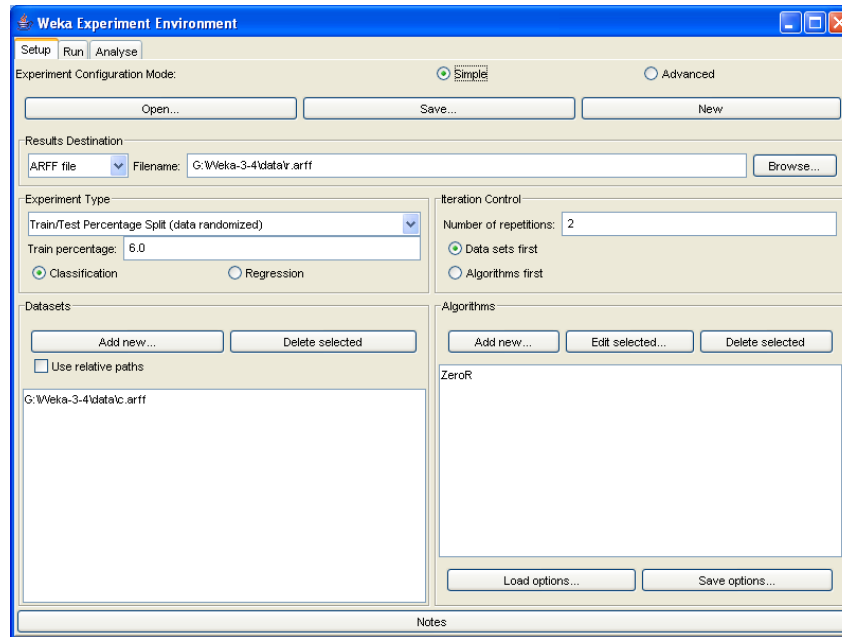
> help

Command must be one of:
  java <classname> <args>
  break
  kill
  cls
  exit
  help <command>
```

Simple CLI (Call Level Interface):
command line interface

Weka interfaces: Experimenter

13



Experimenter: automation of large experiments by varying datasets, algorithms, parameters, ..

Details

14

- Weka manual
 - ▣ Installation directory, or at the weka website



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

WEKA Manual
for Version 3-7-12

Remco R. Bouckaert
Eibe Frank
Mark Hall
Richard Kirkby
Peter Reutemann
Alex Seewald
David Scuse

December 16, 2014

Filters

15

- Conversions
 - MakeIndicator, NominalToBinary, NumericToBinary, NumericToNominal
- Selections
 - RemovePercentage, RemoveRange, RemoveWithValues, SubSetByExpression
- Sampling
 - Resample, SpreadSubSample, StratifiedRemoveFolds
- Transformation
 - Add, AddExpression, AddNoise, AddValues
- Normalization
 - Center, Normalize, Standardize
- Discretization
 - Discretize
- Cleaning
 - NumericCleaner, Remove, RemoveByType, RemoveUseless
- Missing Values
 - ReplaceMissingValues

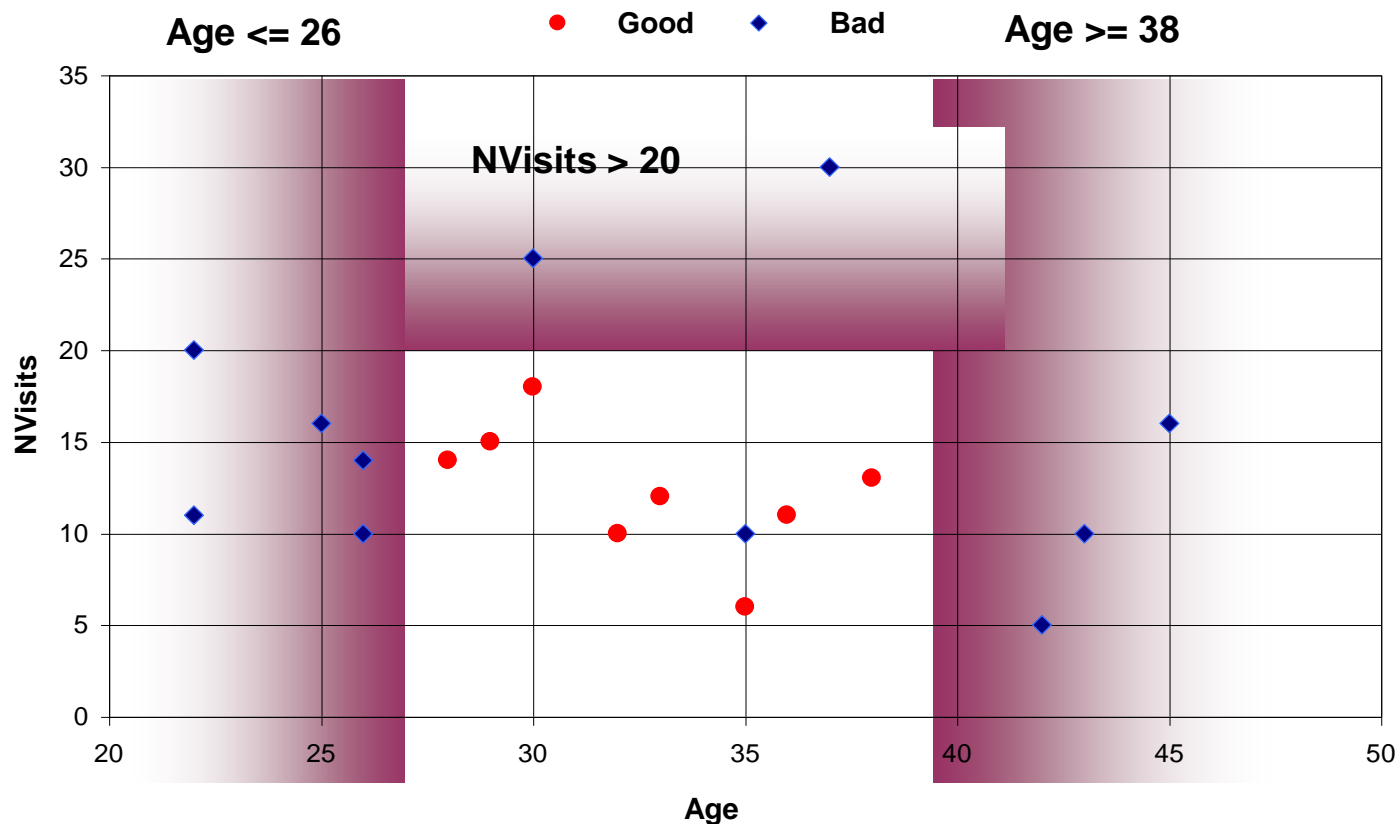
16

Reminds on classification

Who are my best customers?

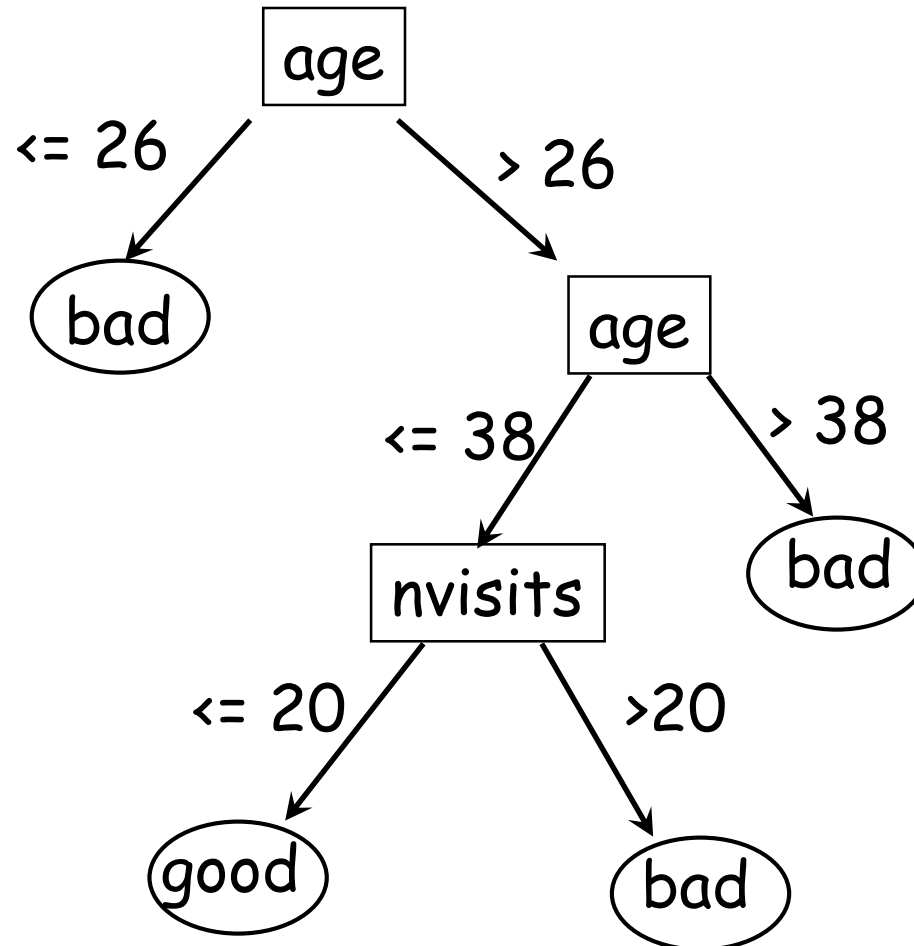
17

- ... given their age and frequency of visit !
- Good customers = top buyers, buy more than X, ...



... described with a decision tree!

18



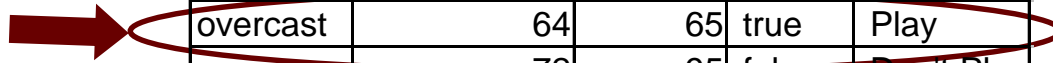
Classification: input

19

A set of examples (or instances or cases) which described a concept or event (**class**) given predictive **attributes** (or **features**)

- ▣ **Attributes** can be either continuous or discrete (maybe discretized)
- ▣ The **class** is discrete

outlook	temperature	humidity	windy	class
sunny	85	85	false	Don't Play
sunny	80	90	true	Don't Play
overcast	83	78	false	Play
rain	70	96	false	Play
rain	68	80	false	Play
rain	65	70	true	Don't Play
overcast	64	65	true	Play
sunny	72	95	false	Don't Play
sunny	69	70	false	Play
rain	75	80	false	Play
sunny	75	70	true	Play
overcast	72	90	true	Play
overcast	81	75	false	Play
rain	71	80	true	Don't Play



Classification: output

20

A function $f(\text{sample}) = \text{class}$, called a **classification model**, that describes/predict the class value given the feature values of a sample obtained by generalizing the samples of the training set

- Usage of a classification model:
 - ▣ **descriptively**
 - Which customers have abandoned?
 - ▣ **predictively**
 - Over a **score set** of samples with unknown class value
 - Which customers will respond to this offer?

How to evaluate a class. model?

21

□ Holdout method

- Split the available data into two sets
- Training set is used to build the model
- Test set is used to evaluate the interestingness of the model
 - Typically, training is 2/3 of data and test is 1/3



outlook	temperature	humidity	windy	class
sunny	85	85	false	Don't Play
sunny	80	90	true	Don't Play
overcast	83	78	false	Play
rain	70	96	false	Play
rain	68	80	false	Play
rain	65	70	true	Don't Play
overcast	64	65	true	Play
sunny	72	95	false	Don't Play
sunny	69	70	false	Play

outlook	temperature	humidity	windy	class
rain	75	80	false	Play
sunny	75	70	true	Play
overcast	72	90	true	Play
overcast	81	75	false	Play
rain	71	80	true	Don't Play

How good is a classification model?

22

- Stratified holdout
 - ▣ Available data is divided by stratified sampling wrt class distribution

- (Stratified) n-fold cross-validation
 - ▣ Available data divided into n parts of equal size
 - ▣ For $i=1..n$, the i-th part is used as test set and the rest as training set for building a classifier
 - ▣ The average quality measure of the n classifiers is statistically more significant than the holdout method
 - ▣ The FINAL classifier is the one training from all the available data
 - Cross-validation is useful when data is scarce or attribute distributions are skewed

Quality measures: accuracy

23

- **Accuracy**: percentage of cases in the test set that is correctly predicted by the model
 - E.g., accuracy of 80% means that in 8 cases out of 10 in the test set the predicted class is the same of the actual class

- Misclassification % = $(100 - \text{accuracy})$

- Lower bound on accuracy: **majority classifier**
 - A trivial classifier for which $f(\text{case}) = \text{majority class value}$
 - Its accuracy is the percentage of the majority class
 - E.g., two classes: fraud 2% legal 98%
 - Its hard to beat the 98% accuracy


Quality measures: confusion matrix


24

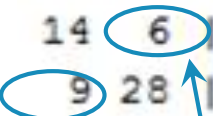
```
Correctly Classified Instances      42           73.6842 %
Incorrectly Classified Instances    15           26.3158 %
Total Number of Instances          57
```

```
=== Confusion Matrix ===
```

```
  a  b  <-- classified as
14  6  |  a = bad
 9 28  |  b = good
```

 Predicted class

 Actual class

 Misclassified cases

Quality measures: precision

25

Precision: accuracy of predicting “C”

$$\frac{\# \text{ Cases predicted Class=C and with real Class=C}}{\# \text{ Cases predicted Class=C}}$$

```
Root mean squared error          0.3213
Relative absolute error          53.8362 %
Root relative squared error      75.46 %
Coverage of cases (0.95 level)  97.8923 %
Mean rel. region size (0.95 level) 73.3048 %
Total Number of Instances        16606
```

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,594	0,058	0,760	0,594	0,667	0,586	0,881	0,748	>50K
	0,942	0,406	0,881	0,942	0,910	0,586	0,881	0,943	<=50K
Weighted Avg.	0,859	0,323	0,852	0,859	0,852	0,586	0,881	0,896	

76% of times predictions >50K are correct

=== Confusion Matrix ===

```
  a    b  <-- classified as
2346 1605 |   a = >50K
 740 11915 |  b = <=50K
```

Quality measures: recall

26

Recall: coverage of predicting “C”

$$\frac{\# \text{ Cases predicted Class=C and with real Class=C}}{\# \text{ Cases with real Class=C}}$$

```
Root mean squared error          0.3213
Relative absolute error          53.8362 %
Root relative squared error      75.46 %
Coverage of cases (0.95 level)   97.8923 %
Mean rel. region size (0.95 level) 73.3048 %
Total Number of Instances       16606
```

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,594	0,058	0,760	0,594	0,667	0,586	0,881	0,748	>50K
	0,942	0,406	0,881	0,942	0,910	0,586	0,881	0,943	<=50K
Weighted Avg.	0,859	0,323	0,852	0,859	0,852	0,586	0,881	0,896	

59,4% of real class >50K are found by predictions



=== Confusion Matrix ===

```
      a      b  <-- classified as
2346 1605 |      a = >50K
 740 11915 |     b = <=50K
```

Measures: lift chart

27

- Classifier: $f(\text{sample}, \text{class}) = \text{confidence}$
 - ▣ and then $f(\text{sample}) = \text{argmax}_{\text{class}} f(\text{sample}, \text{class})$
 - ▣ E.g., $f(\text{sample}, \text{play}) = 0.3$ $f(\text{sample}, \text{don't play}) = 0.7$
- Samples in the test set can be ranked according to a fixed class
 - ▣ Rank customers on the basis of the classifier confidence **they will respond** to an offer
- **Lift chart**
 - ▣ X-axis: ranked sample of the test set
 - ▣ Y-axis: percentage of the total cases in the test set with the actual class value included in the ranked sample of the test set (i.e., recall)
 - ▣ Plots: performance of a classifier vs random ranking
 - ▣ Useful when resources (e.g., budget) are limited

Contacting only 50% of customer will reach 80% of those who respond. Lift = $80/50 = 1.6$



Lift Chart - variants

29

- $\text{Lift}(X) = \text{recall}(X)$
 - ▣ Estimation of random classifier lift
 - ▣ Previous example, $\text{Lift}(50\%) = 80\%$

- $\text{LiftRatio}(X) = \text{recall}(X) / X$
 - ▣ Ratio of lift over random order
 - ▣ Previous example, $\text{LiftRatio}(50\%) = 80\% / 50\% = 1.6$

- Profit chart
 - ▣ Given a cost/benefit model, the Y axis represent the total cost/gain when contacting X and not contacting $\text{TestSet} \setminus X$

The unbalancing problem

30

- For unbalanced class values, it is difficult to obtain a good model
 - Fraud = 2% Normal = 98%
 - The majority classifier is accurate at 98% but it is not useful

- Oversampling and Undersampling
 - Select a **training** set with a more balanced distribution of class values A and B
 - 60-70% for class A and 30-40% for class B
 - By increasing the number of cases with class B (oversampling) or by reducing those with class A (undersampling)
 - The training algorithm has more chances of distinguishing characteristics of A VS B
 - The test set **MUST** have the original distribution of values

- Cost Sensitive Classifier, Ensembles (bagging, boosting, stacking)
 - Weights errors, build several classifiers and average their predictions

31

Rule based classification

Rule-Based Classifier

- Classify records by using a collection of “if...then...” rules
- Rule: $(Condition) \rightarrow y$
 - where
 - *Condition* is a conjunctions of attributes
 - y is the class label
 - *LHS*: rule antecedent or condition
 - *RHS*: rule consequent
 - Examples of classification rules:
 - $(Blood\ Type=Warm) \wedge (Lay\ Eggs=Yes) \rightarrow Birds$
 - $(Taxable\ Income < 50K) \wedge (Refund=Yes) \rightarrow Evade=No$

Rule-based Classifier (Example)

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
human	warm	yes	no	no	mammals
python	cold	no	no	no	reptiles
salmon	cold	no	no	yes	fishes
whale	warm	yes	no	yes	mammals
frog	cold	no	no	sometimes	amphibians
komodo	cold	no	no	no	reptiles
bat	warm	yes	yes	no	mammals
pigeon	warm	no	yes	no	birds
cat	warm	yes	no	no	mammals
leopard shark	cold	yes	no	yes	fishes
turtle	cold	no	no	sometimes	reptiles
penguin	warm	no	no	sometimes	birds
porcupine	warm	yes	no	no	mammals
eel	cold	no	no	yes	fishes
salamander	cold	no	no	sometimes	amphibians
gila monster	cold	no	no	no	reptiles
platypus	warm	no	no	no	mammals
owl	warm	no	yes	no	birds
dolphin	warm	yes	no	yes	mammals
eagle	warm	no	yes	no	birds

R1: (Give Birth = no) \wedge (Can Fly = yes) \rightarrow Birds

R2: (Give Birth = no) \wedge (Live in Water = yes) \rightarrow Fishes

R3: (Give Birth = yes) \wedge (Blood Type = warm) \rightarrow Mammals

R4: (Give Birth = no) \wedge (Can Fly = no) \rightarrow Reptiles

R5: (Live in Water = sometimes) \rightarrow Amphibians

Application of Rule-Based Classifier

- A rule r **covers** an instance x if the attributes of the instance satisfy the condition of the rule

R1: (Give Birth = no) \wedge (Can Fly = yes) \rightarrow Birds

R2: (Give Birth = no) \wedge (Live in Water = yes) \rightarrow Fishes

R3: (Give Birth = yes) \wedge (Blood Type = warm) \rightarrow Mammals

R4: (Give Birth = no) \wedge (Can Fly = no) \rightarrow Reptiles

R5: (Live in Water = sometimes) \rightarrow Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
hawk	warm	no	yes	no	?
grizzly bear	warm	yes	no	no	?

The rule R1 covers a hawk \Rightarrow Bird

The rule R3 covers the grizzly bear \Rightarrow Mammal

Rule Coverage and Accuracy

- **Coverage of a rule:**
 - Fraction of records that satisfy the antecedent of a rule
- **Accuracy of a rule:**
 - Fraction of records that satisfy both the antecedent and consequent of a rule

<i>Tid</i>	Refund	Marital Status	Taxable Income	Class
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

(Status=Single) → No

Coverage = 40%, Accuracy = 50%

How does Rule-based Classifier Work?

R1: (Give Birth = no) \wedge (Can Fly = yes) \rightarrow Birds

R2: (Give Birth = no) \wedge (Live in Water = yes) \rightarrow Fishes

R3: (Give Birth = yes) \wedge (Blood Type = warm) \rightarrow Mammals

R4: (Give Birth = no) \wedge (Can Fly = no) \rightarrow Reptiles

R5: (Live in Water = sometimes) \rightarrow Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
lemur	warm	yes	no	no	?
turtle	cold	no	no	sometimes	?
dogfish shark	cold	yes	no	yes	?

A lemur triggers rule R3, so it is classified as a mammal

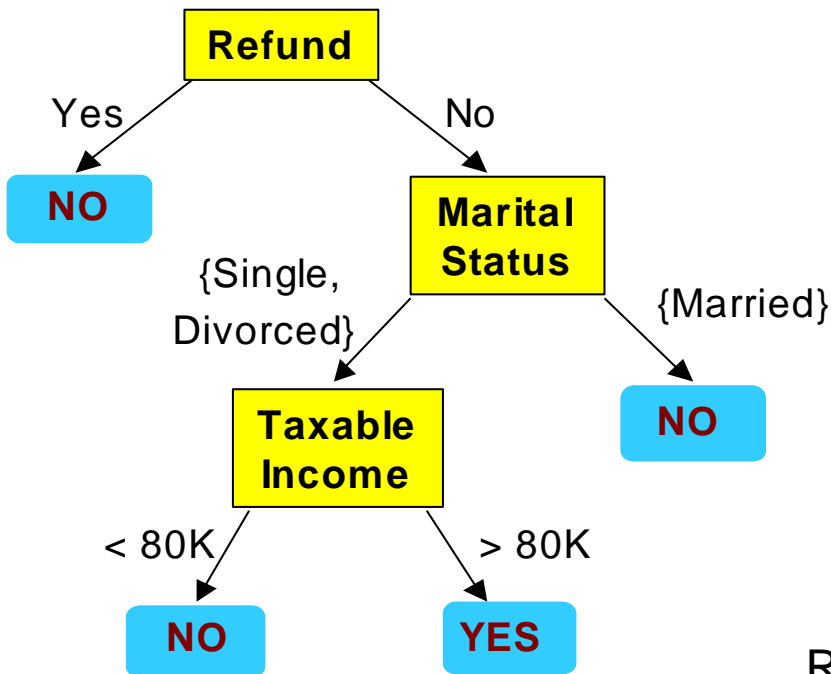
A turtle triggers both R4 and R5

A dogfish shark triggers none of the rules

Characteristics of Rule-Based Classifier

- Mutually exclusive rules
 - Classifier contains mutually exclusive rules if the rules are independent of each other
 - Every record is covered by **at most one rule**
- Exhaustive rules
 - Classifier has exhaustive coverage if it accounts for every possible combination of attribute values
 - Each record is covered by **at least one rule**

From Decision Trees To Rules



Classification Rules

(Refund=Yes) ==> No

(Refund=No, Marital Status={Single,Divorced}, Taxable Income<80K) ==> No

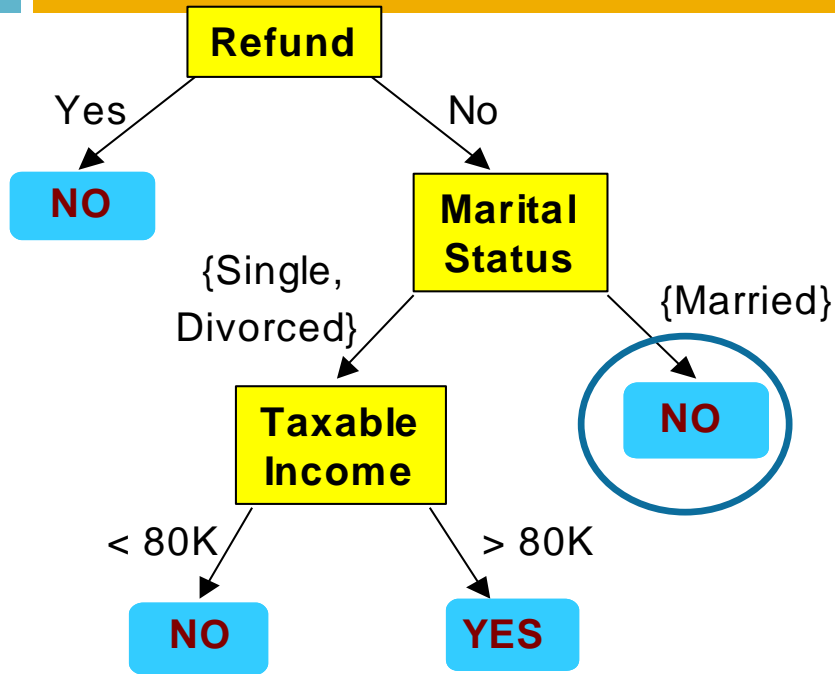
(Refund=No, Marital Status={Single,Divorced}, Taxable Income>80K) ==> Yes

(Refund=No, Marital Status={Married}) ==> No

Rules are mutually exclusive and exhaustive

Rule set contains as much information as the tree

Rules Can Be Simplified



Tid	Refund	Marital Status	Taxable Income	Cheat
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

Initial Rule: $(\text{Refund}=\text{No}) \wedge (\text{Status}=\text{Married}) \rightarrow \text{No}$

Simplified Rule: $(\text{Status}=\text{Married}) \rightarrow \text{No}$

Effect of Rule Simplification

- Rules are no longer mutually exclusive
 - ▣ A record may trigger more than one rule
 - ▣ Solution?
 - Ordered rule set
 - Unordered rule set – use voting schemes

- Rules are no longer exhaustive
 - ▣ A record may not trigger any rules
 - ▣ Solution?
 - Use a default class

Building Classification Rules

□ Direct Method:

- Extract rules directly from data
- e.g.: RIPPER, CN2, Holte's 1R

□ Indirect Method:

- Extract rules from other classification models (e.g. decision trees, neural networks, etc).
- e.g: C4.5rules