Assessment of quality in education - the way for reputation building

Dumitru Dan BURDESCU

Software Engineering Department
University of Craiova
Romania

Acknowledgement

My colleague Cristian Mihaescu has made an important contribution to this work regarding:

- web application development (Tesys e-Learning platform),
- machine learning algorithms adaptation and integration
- Experiments design and analysis

E-Learning Definition

Elliott Masie - internationally recognized futurist, analyst, researcher

(http://www.masie.com/elliott-masie.html)

"*The use of technology to design, deliver, select, administer, support and extend learning*"

Percepsys - http://www.percepsys.com/

(canadian company)

"*Using a technological means (Internet/Intranet/Extranet) to access and manage learning that supports and enhances the knowledge of an individual*"

E-Learning advantages

- No more expensive travel costs
- Less staff time wasted in travel
- Immediate availability
- Self-paced learning and increased confidence
- Instructional quality
- Instant feedback and scores
- Instant and less costly updates

Obstacles in E-Learning

Human

- Organizational
- Cultural Resistance
- Instructional

Technological

- Bandwidth
- Interactivity
- Technology support
- Development costs

E-Learning Challenges

<u>Content Development Bottleneck</u>

- –Long time to develop course
- –Existence of multi-dimensional skills - Web Team, Design team, Learning standards, Instructional design

<u>Infrastructure Problems</u>

- High cost of purchase, implementation and deployment
- Problematic and incompatible features between disparate systems
- Frequently -LMS, LCMS and Portal integration problems
- Difficult to measure activity, results and impact

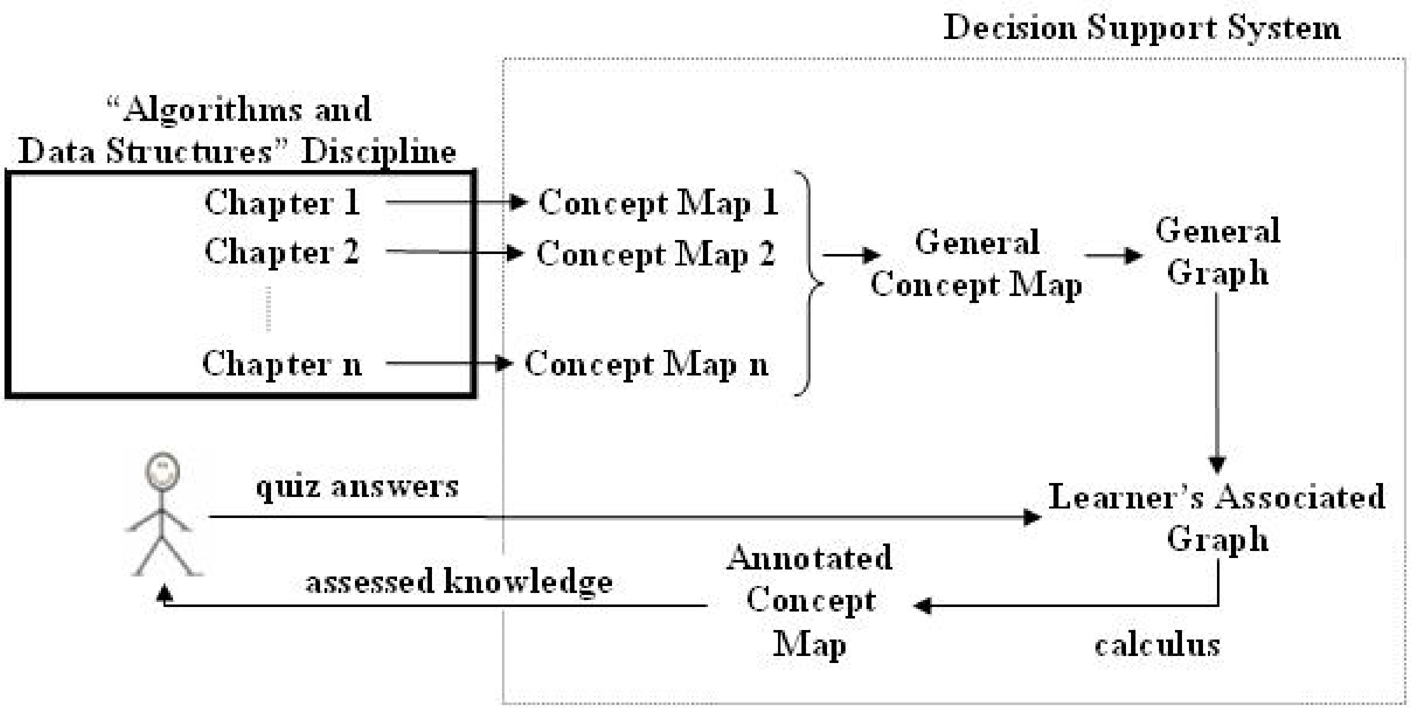
<u>QUALITY</u> - Quality assessment and reputation building

Measuring Quality

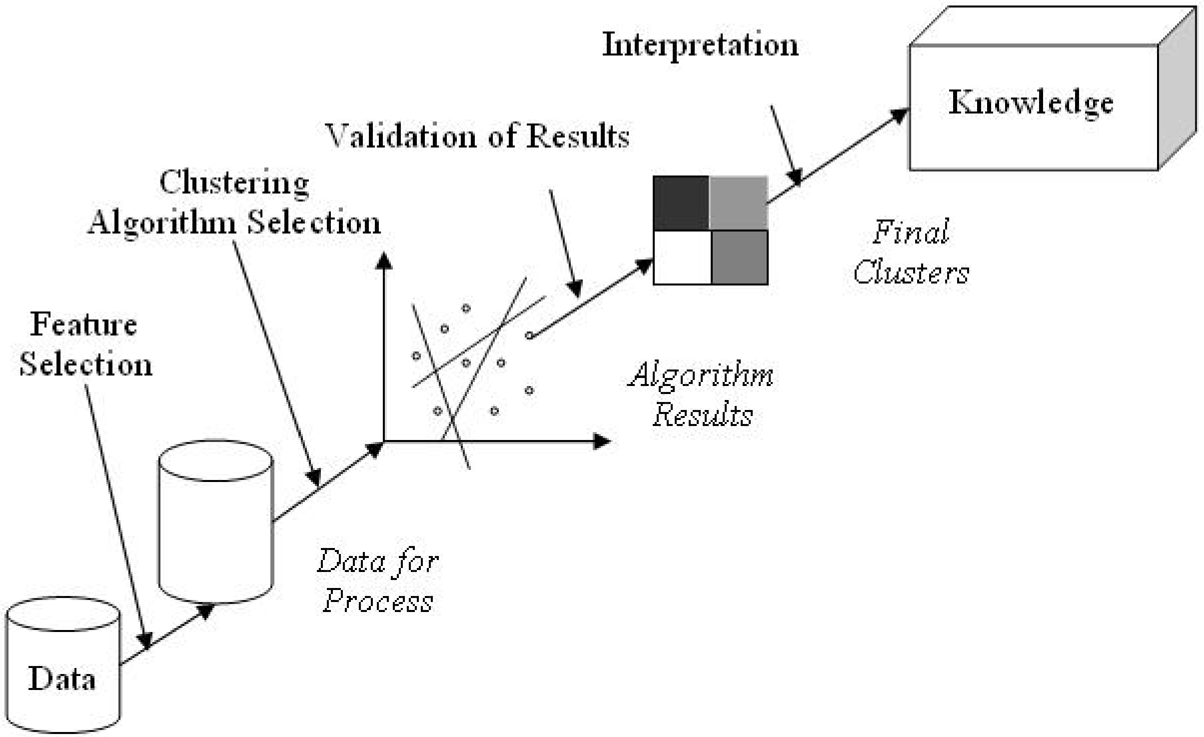
Road map:

- **structure** the materials (e.g. concept maps)
- **data** - log performed activities
- **algorithms** - analyze performed activities
- **process** - produce valuable feedback and conclusion regarding quality issues

Structuring a dicipline

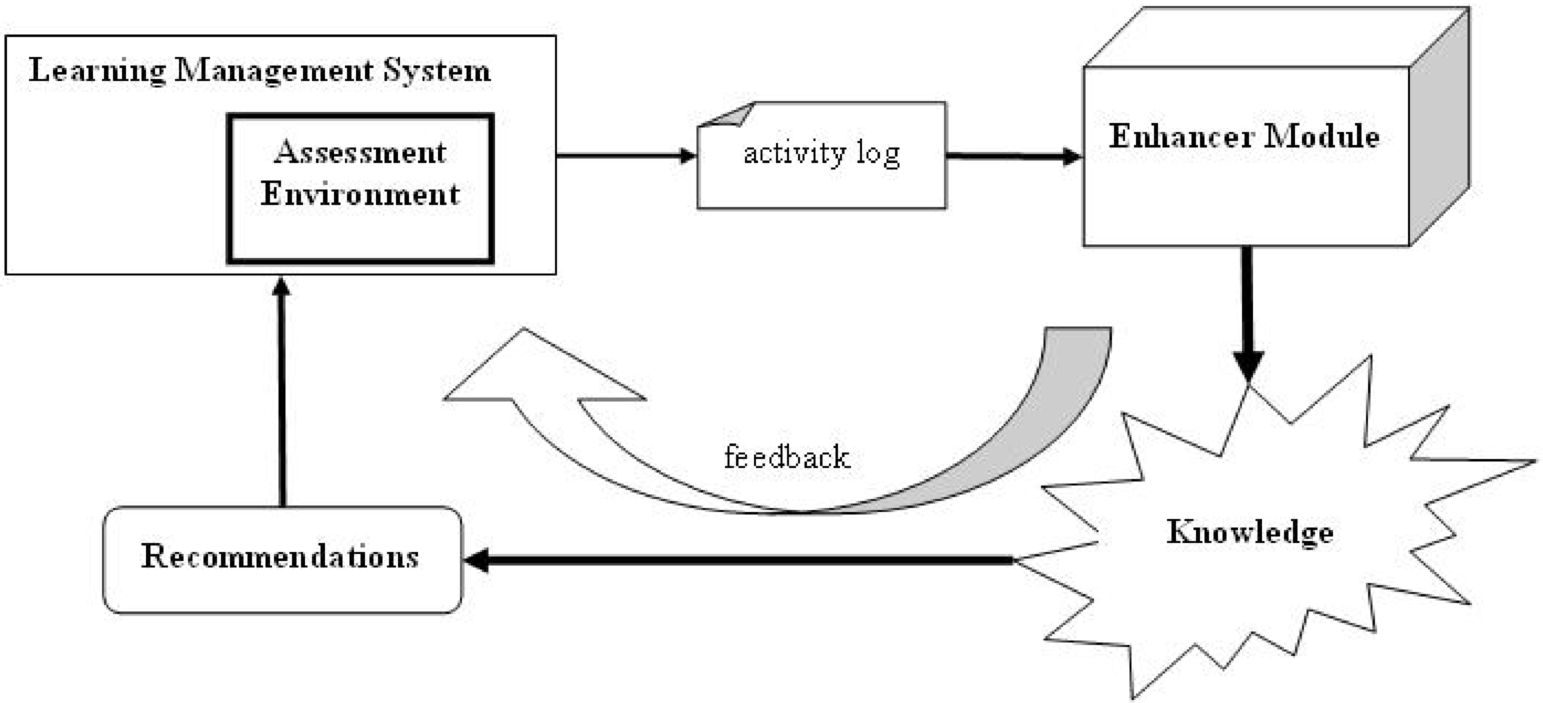


Steps of a learning process



Steps of learning process

Integrating Machine Learning



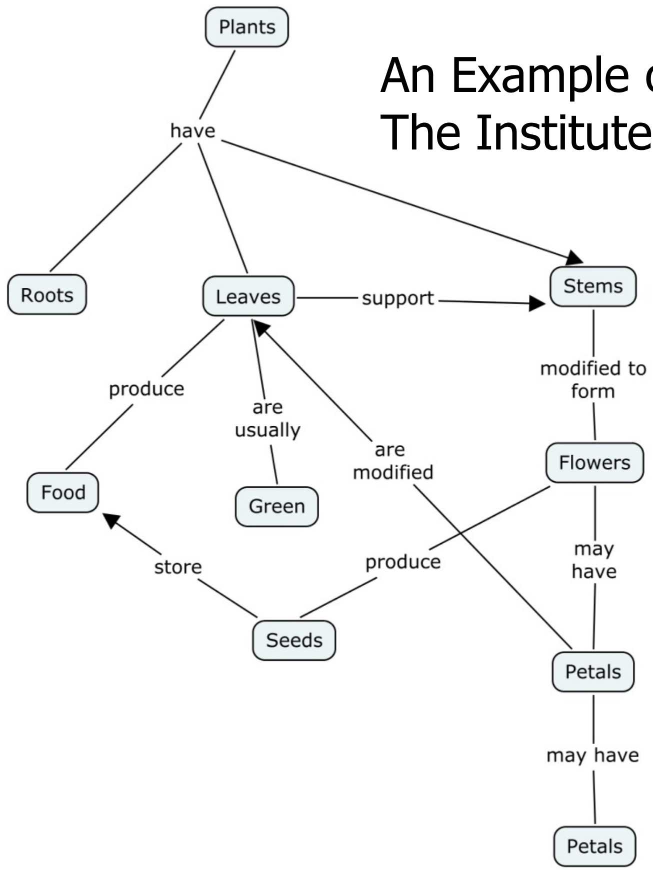
Concept Maps

- Two-dimensional, hierarchical diagrams that show the structure of knowledge within a discipline
- Composed of concept labels, each enclosed in a box or oval, a series of labeled linking lines and general-to-specific organization.
- Concept map – diagrams indicating interrelationships among concepts and representing conceptual frameworks within a specific domain of knowledge

Concept Maps (cont.)

- CMap tools (IHMC) that we will use today
- C-TOOLS – Luckie (PI), University of Michigan NSF grant available: http://ctools.msu.edu/ctools/index.html
- TPL-KATS – University of Central Florida (e.g., Hoeft, Jentsch, Harper, Evans, Bowers, & Salas, 1990). TPL-KATS: concept map: a computerized knowledge assessment tool. Computers in Human Behavior, 19 (6), 653-657.
- SEMNET – http://www.semanticresearch.com/about/
- CMAT – Arneson & Lagowski, University of Texas, http://chemed.cm.utexas.edu

Concept Maps (cont.)



An Example of a Concept Map (Novak, The Institute for Human and Machine Cognition)

Activity monitoring

- Log4j utility
- Log4j.properties file

log4j.appender.R.File=D:/devel/Tomcat/idd.log

log4j.appender.R.MaxFileSize=1000KB

log4j.appender.R.MaxBackupIndex=5

Activity monitoring (cont)

Field	Description
id	primary key
userid	identifies the user who performed the action
date	stores the date when the action was performed
action	stores a tag that identifies the action
details	stores details about performed action
level	specifies the importance of the action

Structure of activity table

Naive Bayes classifier

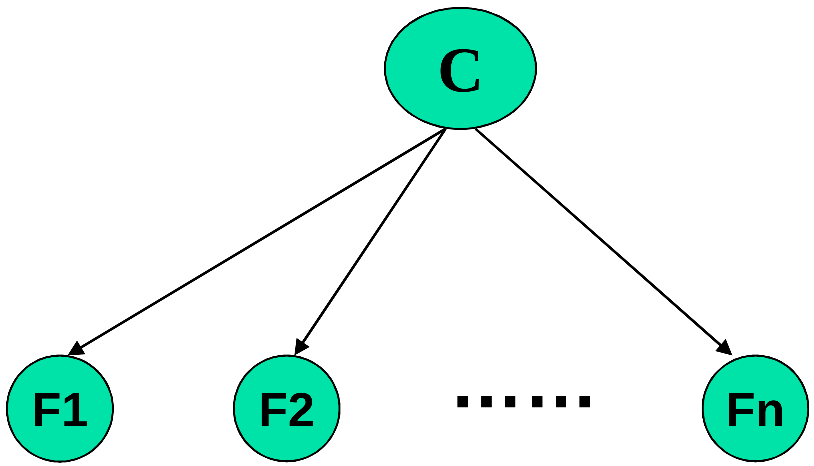
A **naive Bayes classifier** is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions, or more specifically, *independent feature model*.

$$p(C|F_1,\dots,F_n) = \frac{p(C)\ p(F_1,\dots,F_n|C)}{p(F_1,\dots,F_n)}. \qquad \text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}.$$

Naive Bayes probability model

Graphical illustration

- a class node C at root, want P(C|F1,…,Fn)
- evidence nodes F - observed features as leaves
- conditional independence between all evidence



Naive Bayes probability model

The classifier is a conditional model

$$p(C|F_1,\dots,F_n) = \frac{p(C)\ p(F_1,\dots,F_n|C)}{p(F_1,\dots,F_n)}.$$

Following the Bayes's rule strictly, we have

$$p(C,F_1,\dots,F_n)$$
$$= p(C)\ p(F_1|C)\ p(F_2|C,F_1)\ p(F_3|C,F_1,F_2)\ p(F_4,\dots,F_n|C,F_1,F_2,F_3)$$ .....

Simplify this through conditional independence - $p(F_i|C,F_j) = p(F_i|C)$

$$p(C,F_1,\dots,F_n) = p(C)\ p(F_1|C)\ p(F_2|C)\ p(F_3|C)\ \cdots$$

So the conditional distribution over the class *C* is

$$p(C|F_1,\dots,F_n) = \frac{1}{Z}p(C)\prod_{i=1}^{n}p(F_i|C)$$

Z is constant given features

Building Clusters from Data

- clustering - process of grouping a set of physical or abstract objects into classes of similar objects
- we create clusters of users based on their activity.
- We have n objects and k clusters to form
- clusters are formed to optimize an objective partitioning criterion (similarity function or distance)
- Step 1: Define a list of attributes
- Step 2: Compute attribute values for each student
- Step 3: Run iterative-based clustering algorithm

Cluster's parameters

- $\mu = \frac{x_1 + x_2 + \ldots + x_n}{n}$, the means
- $\sigma = \frac{(x_1 - \mu)^2 + (x_2 - \mu)^2 + \ldots + (x_n - \mu)^2}{n-1}$,the standard deviation
- p, the probability
  - Sum of all probabilities for all clusters is 1.
  - Knowing instances distribution in clusters we can determine parameters
  - Knowing parameters, we can determine the probability that a given instance comes from a cluster

EM algorithm

- We do not know the distribution that each training instance came from, nor the parameters μ, σ or the probability.
- Start with initial guess for the five parameters, use them to calculate the cluster probabilities for each instance
- Use these probabilities to re-estimate the parameters, and repeat ("expectation-maximization")
- "expectation" - The first step which computes cluster probabilities;
- "maximization" - The second step, calculation of the distribution parameters is of the likelihood of the distributions given the data.

EM algorithm (cont.)

- k-means algorithm stops when the classes of instances don’t change from one iteration to the next – a “fixed point” has been reached.
- The EM algorithm converges toward a fixed point
- We can see how close it is by calculating the overall likelihood
- Overall likelihood is a measure of the “goodness” of clustering and increases at each iteration of the EM algorithm.
- Conclusion: iterate until the increase in log-likelihood becomes negligible

Decision Tree Induction Algorithm

- Decision trees “divide-and-conquer” approach
- The nodes from a decision tree imply testing a certain attribute
- Creating a decision tree can be expressed in a recursive way
- The most important thing is the order in which the attributes are taken into consideration

Decision Tree Induction Algorithm

Algoritm: Generate_Decision_Tree: Creates a decision tree
Input : Instances with discrete attribute values
Output : Decision tree
Method :
(1) Create a node N ;
(2) **if** *instances* are all in same class C **than**
(3) **return** N – leaf node labeled with class C;
(4) **if** *attribute list* is empty **than**
(5) **return** N as leaf node labeled with most appropriate class;// majority voting;
(6) Select a *test attribute* , attribute with largest information gain
(7) Label node N with *test attribute*
(8) For each possible value of *test attribute*
//instance partitioning
(9) Create a branch for each *test attribute*;
(10) Let Si be the set of instances for which *test attribute* = ai //a partition
(11) **if** Si is empty **than**
(12) Create a leaf labeled with the most representative class.
(13) **else** create a node returned by Generate_Decision_Tree (Si, attribute list, test attribute)

Decision trees

- Attributes:

  AT = the **a**verage grade of taken **t**ests for a student

  NT = the **n**umber of taken **t**ests for a student

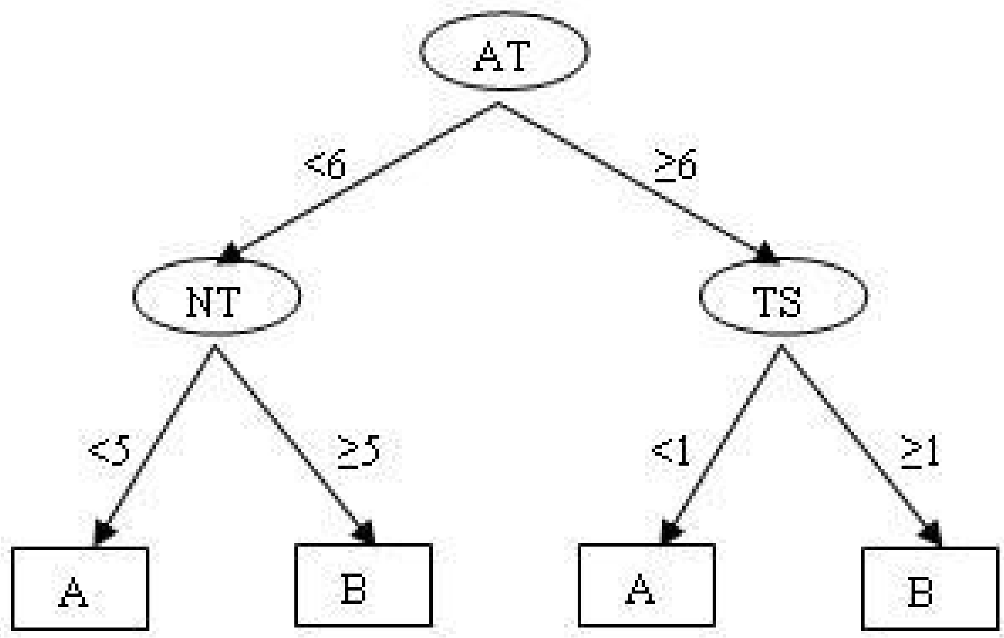
  TS = **t**ime **s**pent for taking tests

- Classes:

  A = the class of students that did not pass the exam

  B = the class of students that passed the exam

Sample decision tree



Experiments

- Define goal
- Define input data
- Choose algorithm and run experiment
- Obtain and interpret results
- Conclusions

Experiment 1 – Naïve Bayes

Define goals:

- Improving learner's proficiency
- Advise the learner regarding the resources he should access and study

Experiment 1 – Naïve Bayes

Define input data

For each attribute there is defined the set of nominal values it may have

@relation activity

@attribute *chapterId* {1, 2, 3, 4}

@attribute *noOfTests* {1, 2, 3, 4, 5}

@attribute *avgTests* {1, 2, 3, 4, 5}

@attribute *finalResult* {1, 2, 3, 4, 5}

@attribute *recommend* {yes,no}

Experiment 1 – Naïve Bayes

Define input data

The second section of the *activity.arff* file is represented by the data itself.

@data
1, 1, 2, 3, no
1, 2, 3, 2, no
2, 3, 4, 3, yes
…..

Experiment 1 – Naïve Bayes

Detailed results

=== Run information ===
Scheme:
weka.classifiers.bayes.NaiveBayes
Relation: activity
Instances: 500
Attributes: 5
chapterId, noOfTests
avgTests, finalResult
recomend
Test mode: evaluate on training data
=== Classifier model (full training set) ===
Naive Bayes Classifier
Class yes: Prior probability = 0.38
chapterId: Counts = 35 62 92 85 (Total = 310)
noOfTests: Counts = 50 68 84 70 40 (Total = 312)
avgTests: Counts = 60 55 80 74 43 (Total = 315)
finalResult: Counts = 40 72 82 80 36 (Total = 310)
Class no: Prior probability = 0.62
chapterId: Counts = 55 139 86 60 90 (Total = 420)
noOfTests: Counts = 50 141 81 85 88 (Total = 445)
avgTests: Counts = 52 132 90 79 94 (Total = 447)
finalResult: Counts = 52 102 100 89 104 (Total = 447)

Experiment 1 – Naïve Bayes

Detailed results

```
=== Evaluation on training set ===
=== Summary ===
Correctly Classified Instances        394           84.2105 %
Incorrectly Classified Instances      106           15.7895 %
Kappa statistic                     0.6503
Mean absolute error                   0.2078
Root mean squared error                 0.3462
Relative absolute error             44.3346 %
Root relative squared error           71.7354 %
Total Number of Instances             500
=== Confusion Matrix ===
  a  b   <-- classified as
 92  81 |  a = yes
 25 302 |  b = no
```

Experiment 1 – Naïve Bayes

Conclusions

- The accuracy of over 84% is good
- This accuracy proves the concept: Naïve Bayes may be used to recommend resources
- Further improvements need to be considered:
  - Improve accuracy
  - Take other attributes into consideration
  - Use other granularities for considered attributes
  - Personalize the recommendation system

Experiment 2 – EM Clustering

Define goals:

- Classify students
- Prove the classification power of an e-Learning platform (assessment setup)

Experiment 2 – EM Clustering

Define input data

//Sample *activity.arff* file

@relation activity

@attribute nLogings {<10,<50,<70,<100,>100}

@attribute nTests {<10,<20,<30,<50,>50}

@attribute nSentMessages {<10,<20,<30,<50,>50}

Experiment 2 – EM Clustering

Define input data

//Sample *activity.arff* file

@data

<50,<20,<10,

<50,>50, <20,

<10,<20, <10,

<50,<10, <10,

<100,<50,<50,

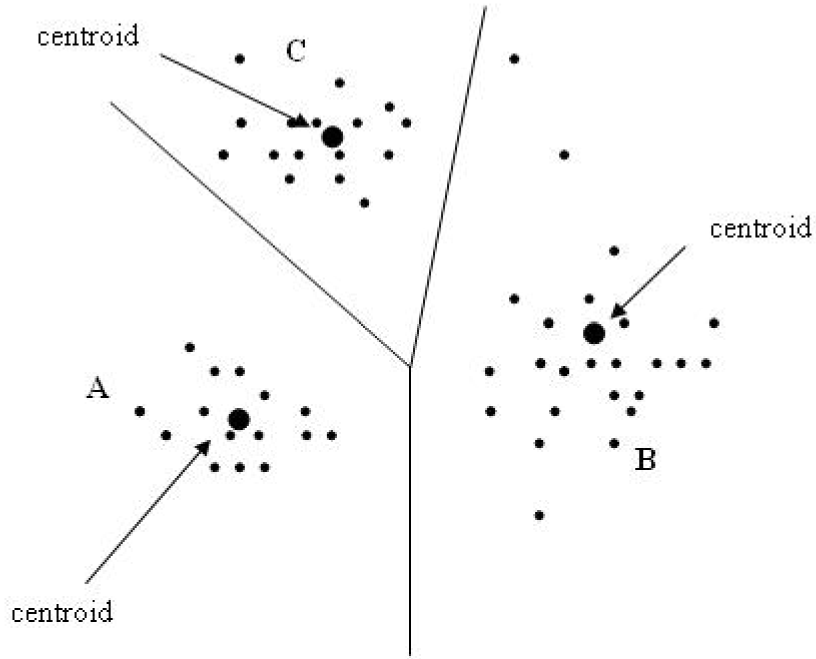
Experiment 2 – EM Clustering

- 91 instances (34%) in cluster A,
- 42 instances (16%) in cluster B and
- 135 instances (50%) in cluster C.

Instance	Cluster A	Cluster B	Cluster C
1	1	0	0
2	1	0	0
3	0	1	0
…….	…	….	…
268	0	0	1

Distribution of instances after EM algorithm

Experiment 2 – EM Clustering



Distribution of instances after EM algorithm with centroids

Experiment 2 – EM Clustering

Conclusions

- Quality of clustering: log-likelihood is -2.61092
- This accuracy proves the concept: clustering is a good method for students classification
- Further improvements need to be considered:
  - Improve accuracy
  - Take/Add other attributes into consideration
  - Use other granularities for considered attributes
  - Define a procedure for giving advice such that a learner may jump from one cluster to another

Experiment 3 – Decision Trees

Define goals:

- Create classes students
- Advice students such that they may pass from one class to another.

Experiment 3 – Decision Trees

Define input data

//Sample *activity.arff* file

@relation activity
@attribute no_of_sessions {1,2,3,4,5}
@attribute mean_delay{1,2,3,4,5}
@attribute mean_session_lenth {1,2,3,4,5}
@attribute mean_no_of_actions {1,2,3,4,5}
@attribute no_of_tests {1,2,3,4,5}
@attribute no_of_messages {1,2,3,4,5}
@data
1,2,4,3,2,3,
2,1,3,1,3,4,
4,1,2,4,1,2,

Experiment 3 – Decision Trees

Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2

Relation: activity
Instances: 375
Attributes: 6
no_of_sessions, mean_delay, mean_session_lenth, mean_no_of_actions, no_of_tests, no_of_messages

Test mode: 10-fold cross-validation

Number of Leaves : 13
Size of the tree : 16
Time taken to build model: 0.13 seconds

=== Stratified cross-validation ===
Correctly Classified Instances 333 (88.8 %)
Incorrectly Classified Instances 42 (11.2 %)

J48 pruned tree

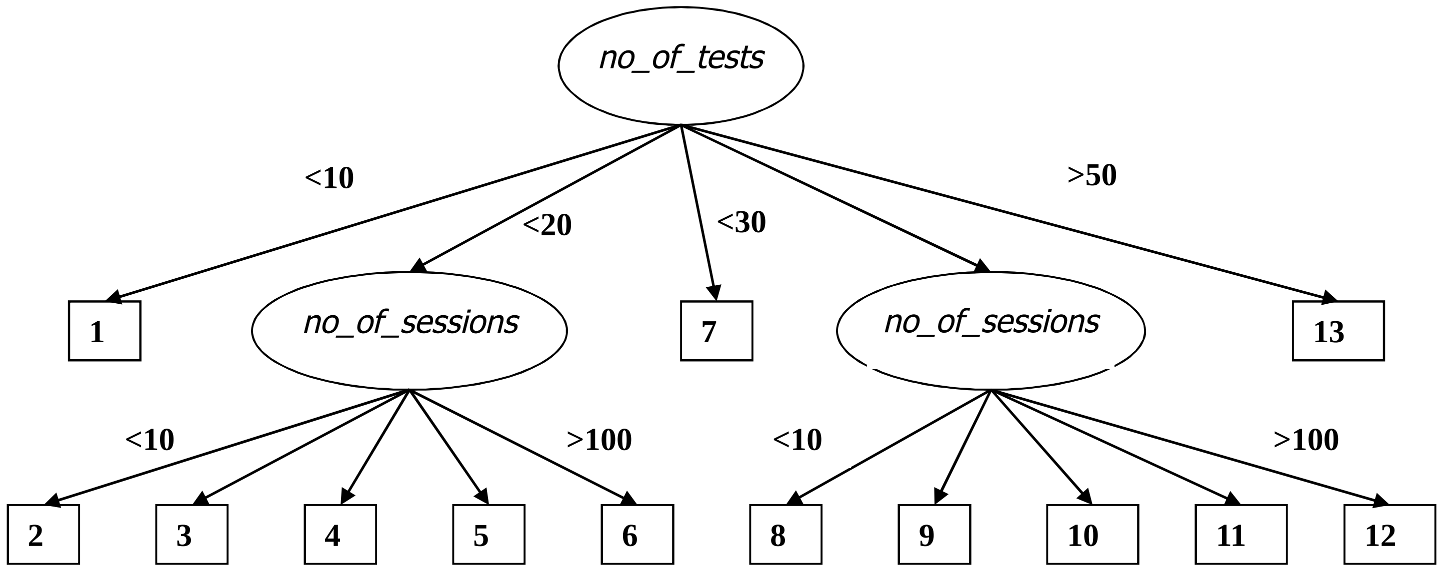
no_of_tests = <10: (60.0/1.0)
no_of_tests = <20
| *no_of_sessions* = <10 (20.0/2.0)
| *no_of_sessions* = <50 (53.0/2.0)
| *no_of_sessions* = <70 (12.0/2.0)
| *no_of_sessions* = <100 (10.0/2.0)
| *no_of_sessions* = >100 (5.0/2.0)
no_of_tests = <30 (40.0/1.0)
no_of_tests = <50
| *no_of_sessions* = <10 (2.0/2.0)
| *no_of_sessions* = <50 (31.0/2.0)
| *no_of_sessions* = <70 (63.0/2.0)
| *no_of_sessions* = <100 (22.0/2.0)
| *no_of_sessions* = >100 (7.0/2.0)
no_of_tests = >50 (50.0/1.0)

Experiment 3 – Decision Trees

Decision Tree:

13 leaves and 16 nodes

Time to build the model: 0.13 seconds



Experiment 3 – Decision Trees

- At beginning, when the data was scarce, the accuracy level was quite low, around 30-40% of instances being correctly classified by cross-validation.
- After three month of running the obtained decision tree had 12 leaves (which represent in fact classes) and 19 nodes

Experiment 3 – Decision Trees

- Stratified cross-validation evaluation technique revealed that :
  - 85% instances were correctly classified
  - 15% were incorrectly classified
- The results prove that obtained model is accurate enough for starting issuing recommendations.

Experiment 3 – Decision Trees

Conclusions

- Quality of classification: 85% correctly classified instances
- This accuracy proves the concept: decision tree induction is a good method for students classification
- Further improvements need to be considered:
  - Improve accuracy: Take other attributes into consideration
  - Use other granularities for considered attributes
  - Define a procedure for giving advice such that a learner may jump from one class to another

Challenge 1: Integration

Integration of:

- Specific e-Learning platform
- Data Collection
- Machine Learning Algorithms
- Knowledge Management
- Feedback
- Metrics

Challenge 2: Automation

Automation: Building a framework which **automatically:**

- Gathers and manages data
- Runs Machine Learning Algorithms on data
- Manages obtained knowledge (i.e. a learner's model)
- Produces output as desired: advice to learners/professors, characterizes learners, characterizes platform, offers statistics about questions, chapters, disciplines or students

Solutions for challenges

Define a set of goals

Set of goals for learners

- Minimization of the time in which a certain level of knowledge is reached. This is accomplished by specifying a desired grade.
- Obtaining for sure a certain grade. The learner has to specify the grade he aims for.

Course managers may choose from two goals:

- Having a normal distribution of grades at chapter level.
- Having a testing environment that ensures a minimum time in which learner reaches a knowledge level for passing the exam.

Solutions for challenges

Define two sets of recommendations

For students:

- More study is necessary for chapter X.
- You may go to the next chapter.
- You need to take more tests at chapter X.

For course managers:

- At chapter X there are needed harder/easier questions.
- At chapter X there are to few/many questions

Solutions for challenges

Recommendation

Example: Overall goal of learner is to reach medium level of knowledge

Learner is in class B:

medium performance in chapter 1

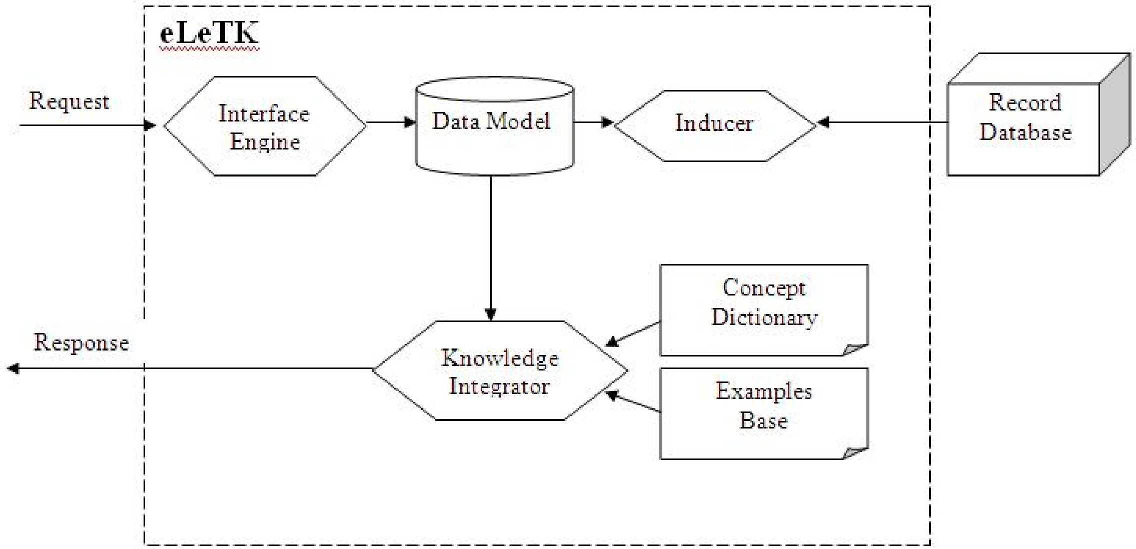
medium performance in chapter 2

minimum performance in chapter 3

minimum performance in chapter 4

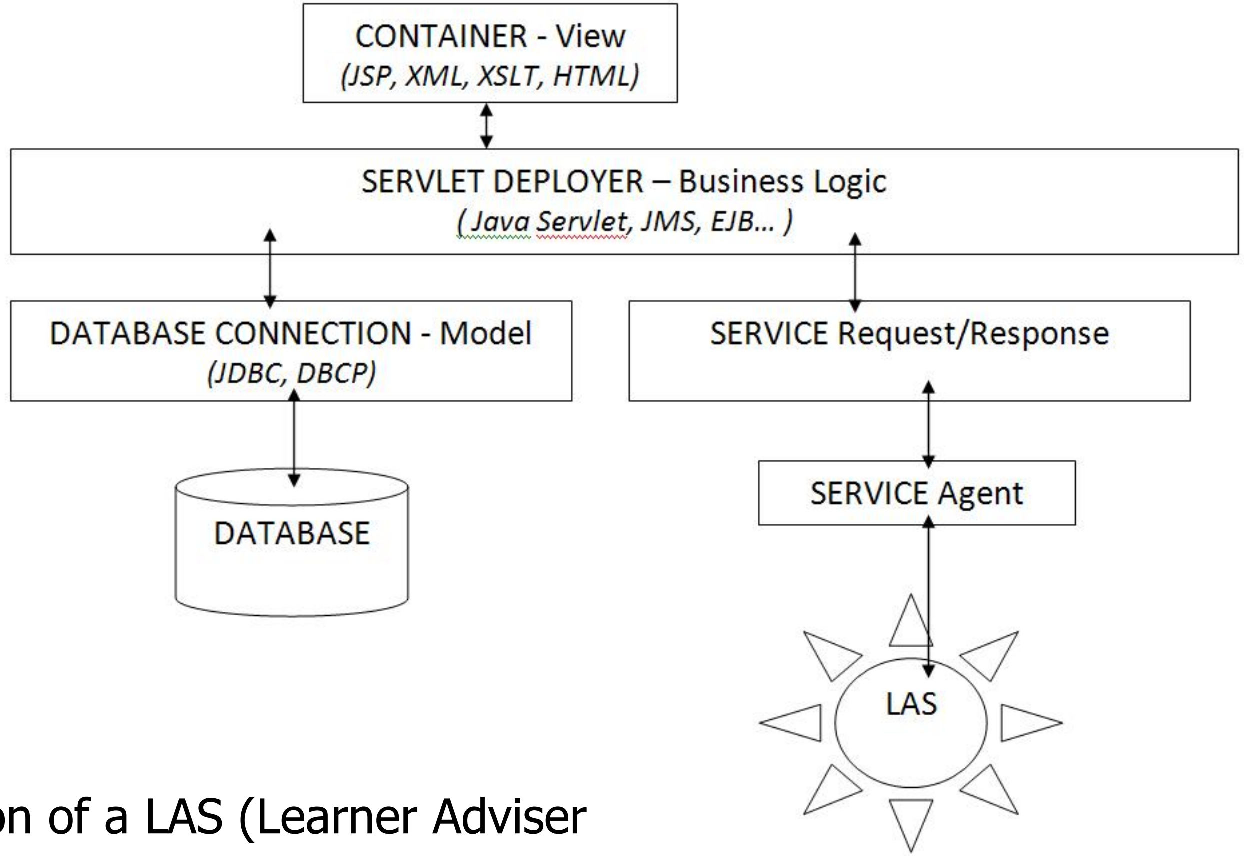
Recommendation will be: “LITTLE MORE study in Chapter 3 and MORE study in Chapter 4”.

Solutions for challenges



Component Based Architecture of eLeTK (e-Learning Enhancer Toolkit)

Solutions for challenges



Integration of a LAS (Learner Adviser Service) in an e-Learning system

Conclusions

- **Data**
  - Need to be representative from quantity and quality point of view
  - Is obtained from a running system (an e-Learning platform)
  - Need to be (semi) structured
  - Represents the INPUT

Conclusions (cont.)

- **Algorithms**
  - Need to be well chosen
  - Must be fine tuned regarding the attributes and their granularity
  - Must provide sound knowledge
  - Improve their results over time

Conclusions (cont.)

- **Analysis logic implementation**
  - Work as a service
  - Have a component based infrastructure
  - Perform as a toolkit along the e-Learning platform that produces data with platform specific setup

Tesys e-Learning platform

Tesys e-Learning platform may be tested –

- as administrator:

http://apps.software.ucv.ro/tesys/servlet/tesys?admin=1

- as learner at:

http://apps.software.ucv.ro/tesys/servlet/tesys

Published papers

- **RCIS 2008,** Dumitru Dan Burdescu, Cristian Marian Mihaescu, Bogdan Logofatu, "Knowledge Evaluation Procedure Based on Concept Maps" PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON RESEARCH CHALLENGES IN INFORMATION SCIENCE (RCIS), MARRAKECH, MOROCCO;
- **ICIW-VEWAeL 2008**, Dumitru Dan Burdescu, Cristian Marian Mihaescu, Bogdan Logofatu, "Employing Bayes Classifier for Improving Learner's Proficiency", Third International Conference on Internet and Web Applications and Services First International Workshop on Virtual Environments and Web Applications for e-Learning, Athens, Greece;
- **ICALT 2008**, Dumitru Dan Burdescu, Cristian Marian Mihaescu, Bogdan Logofatu, "Personalized Content Delivery by Usage of Concept Maps and Naïve Bayes Classifier", The 8th IEEE International Conference on Advanced Learning Technologies, Santander, Cantabria, Spain;
- **IDC 2009**, Marian Cristian Mihaescu, Dumitru Dan Burdescu, Mihai Mocanu, Costel Ionascu, "Obtaining Knowledge using Educational Data Mining", 3rd International Symposium on Intelligent Distributed Computing, Ayia Napa, Cyprus;
- **IGI Book Chapter 2009,** Dumitru Dan Burdescu, Cristian Marian Mihaescu,
  - **Book title:** Monitoring and Assessment in Online Collaborative Environments: Emergent Computational Technologies for E-Learning Support
  - **Chapter title:** *Improvement of Self-Assessment Effectiveness by Activity Monitoring and Analysis*

Assessment of quality in education - the way for reputation building

Thank you for your time!