CHAPTER-4

Analyze Students’ Performance Using Association Techniques

4.1 INTRODUCTION

Over the years, several statistical tools have been used to analyze students’ performance from different points of view. This paper presents data mining in education environment that identifies students’ failure patterns using association rule mining technique. The identified patterns are analyzed to offer a helpful and constructive recommendations to the academic planners in higher institutions of learning to enhance their decision making process. This will also aid in the curriculum structure and modification in order to improve students’ academic performance and trim down failure rate.

Association rule mining has been applied to Education systems for traditionally association analysis (finding correlations between items in a dataset), including, e.g., the following tasks: identifying attributes characterizing patterns of performance disparity between various groups of students, discovering interesting relationships from a student’s usage information in order to provide feedback to the course author, finding out the relationships between each pattern of a learner’s behavior, finding student mistakes often occurring together, guiding the search for the best fitting transfer model of student learning, optimizing the content of syllabus by determining the content of most interest to the user, extracting useful patterns to help educators and masters evaluating and interpreting course activities.

Association rules mining is one of the most well studied data mining tasks. It discovers relationships among attributes in databases, producing if-then statements concerning attribute values [120]. Association rule mining has been applied to EDS (Education System) from two points of view: 1) help professors to obtain detailed feedback of the learning process: e.g.,
finding out how the students learn on the classroom, to evaluate the students based on their navigation patterns, to classify the students into groups, to restructure the contents of the syllabus to personalize the courses; and 2) help students in their interaction with the education system: e.g., adaptation of the course according to the learner's progress, e.g., by recommending to them personalized learning paths based on the previous experiences other similar students.

This Chapter is organized in the following way: First, we describe the background of association rule mining in general and more specifically its application to Education System. Then, we describe the main drawbacks of and some solutions for applying association rule algorithms in EDS. Next, we show a practical implementation of using an association rule mining algorithm over data generated from an Education system. Finally, the conclusions and further research are outlined.

The Apriori algorithm [121] has become the standard approach to mine association rules. We have adapted it to mine class association rules in the way explained by Liu et al. [122]. The second algorithm, Predictive Apriori, has been recently proposed by Scheffer [123]. Both algorithms have their first step in common. They generate frequent item sets in the same way. An item set is called frequent when its support is above a predefined minimum support.

4.2 Generating frequent item sets
Finding association rules can be seen as a simple search problem. But an exhaustive search is intractable because the possible number of association rules is exponential with respect to the number of attributes. For n binary attributes there are \( O(n^{2n-1}) \) rules. It is even worse for discrete valued attributes \( \overline{I} \) assuming there are n attributes and each can take m values, there are \( O(m^n) \) possible rules.
Nonetheless it is possible to perform a search in reasonable time because of the support based downward closure of frequent item sets. An item set \( X \) of length \( k \) is frequent if and only if
all subsets of \( X \) with length \( k \) are frequent. This property allows the search space to be pruned substantially.

Algorithmically we start with all frequent item sets of size 1. This set of frequent item sets consists of all individual items that have a support above a user defined minimum support. This can be done with one pass over the data in linear time. To get the frequent item sets of size 2 there are two steps: First the frequent item sets of size 1 are combined in every possible way to build candidate item sets of size 2, then, in another pass over the data, the candidate item sets are checked to make sure that they are really frequent. All infrequent ones are deleted.

Termination is obvious if either the set of frequent item sets is empty for a distinct \( k \), or \( k \) equals the number of attributes. At most \( n \) linear passes over the data are required if \( n \) is the number of attributes. From the frequent item sets both algorithms generate association rules in different ways using a different measure for interestingness.

### 4.3 The Apriori algorithm

The Apriori algorithm [121] is one of the earliest algorithms for mining association rules and has become the standard approach in this area. The search for association rules is guided by two parameters: support and confidence. Apriori returns an association rule if its support and confidence values are above user defined threshold values. The output is ordered by confidence. If several rules have the same confidence, then they are ordered by support. Thus Apriori favors more confident rules and characterizes these rules as more interesting. The Apriori Mining process is composed of two major steps. The first one (generating frequent item sets) was discussed briefly in the last section. This step can be seen as support based pruning, because only item sets with at least minimum support were considered. The second step is the generation of rules out of the frequent item sets. In this step confidence based pruning is applied. Rule discovery is straightforward. For every frequent item set \( f \) and every
non-empty subset s of f, Apriori outputs a rule of the form \( s \Rightarrow (f \cap \bar{s}) \) if and only if the confidence of that rule is above the user specified threshold.

Any subset of a large item set must be large. Item set having k items can be generated by joining large item sets having \( k-1 \) items, and deleting those that contain any subset that is not large.

Def. \( k \)-item set: large item set with \( k \) items.

Algorithm consists of two phases:

4.3.1 Candidate Generation

4.3.2 Pruning

---

<table>
<thead>
<tr>
<th>Gen candidate itemsets with the given ( L_{k-1} ) as follows:</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_k = \emptyset )</td>
</tr>
<tr>
<td>for all item set ( I \subseteq L_{k-1} ) do</td>
</tr>
<tr>
<td>If ( L_1[1] = L_2[1] ) and ( L_1[2] = L_2[2] ) and ( \ldots ) and ( L_1[k-1] &lt; L_2[k-1] )</td>
</tr>
<tr>
<td>then ( c = L_1[1], L_2[2], \ldots, L_1[k-1], L_2[k-1] )</td>
</tr>
<tr>
<td>( C_k = C_k \cup {c} )</td>
</tr>
</tbody>
</table>

Apriori candidate generation

Example:

\( L_3 = \{ \{1,2,3\}, \{1,2,5\}, \{1,3,5\}, \{2,3,4\} \} \)

Algorithm will generate following itemset

\( \{1,2,3,5\} \) is generated from \( \{1,2,3\} \) and \( \{1,2,5\} \).

Similarly, \( \{1,2,3,5\}, \{2,3,4,5\} \)

**pruning**

The pruning step eliminates the extensions of \( (K-1) \)-itemsets, which are not frequent
Prune(C_k)
for all c \subseteq C_k
for all (k-1)-subsets d of c do
if d \subseteq L_{k-1} then
C_K = C_k \setminus \{c\}

APRIORI ALGORITHM

Initialize: K := 1; C_1 = all the 1-itemsets;
1. Read the database to count the support of C_1 to determine L_1
L_1 = {frequent 1-itemsets};
k := 2; // k represent the pass number//
while (L_{k-1} \neq \emptyset) do
begin
C_K := gen_candidate_itemsets with the given L_{k-1}.
Prune (C_K)
for all transaction t \subseteq T do
increment the counts of all itemsets in C_K
L_K := All itemsets in C_K with minimum support;
k := k + 1;
End
Answer = \bigcup_i L_K:
4.4 The Predictive Apriori algorithm

The Predictive Apriori algorithm [123] differs from standard Apriori in such a way that it employs a different measure of interestingness of an association rule. Both techniques use a support based search and take advantage of the downward closure of the support, which allows the exponential search space of possible association rules to be pruned. Apriori favors more confident rules and ranks the rules accordingly. Predictive Apriori on the other hand evaluates the confidence of rules depending on their support. Its interestingness measure is to maximize the expected accuracy an association rule will have on unseen data. This suits the requirements of the classification task we want to perform afterwards. One problem of the confidence and support based Apriori measurement scheme is that one can always find very general rules with high support and low confidence and very specific ones with high confidence and low support. After mining a set of rules we use them for classification. So we are interested in whether the rule set built using the training instances is capable of generalization and predicting the class labels of test instances correctly. We want to mine association rules which associate items that are correlated not only in the training data, but in reality, too. Instead of confidence the algorithm employs the so-called predictive accuracy. Scheffer defines predictive accuracy as:

Let D be a database whose individual records r are generated by a static process P, let \( X \Rightarrow Y \) be an association rule.

The **predictive accuracy** \( c(X \Rightarrow Y) = \Pr (r \text{ satisfies } Y \mid r \text{ satisfies } X) \) is the conditional probability of \( Y \subseteq r \) given that \( X \subseteq r \) when the distribution of \( r \) is governed by \( P \) [123]. The confidence \( c(X \Rightarrow Y) \) of the association rule \( X \Rightarrow Y \) is the relative frequency of the predictive accuracy in the data; that is the relative frequency of a correct classification in the training database. Hence the confidence value is optimistically biased if one wants to use it for a predictive task [123]. Uses a Bayesian framework to calculate the predictive accuracy.
out of the support and confidence of a rule. In doing so the support is a rough guideline of how much we should mistrust the confidence. The higher the support, the more the confidence converges to the expected accuracy on future data. This approach is called Bayesian frequency correction [123], because the predictive accuracy equals a corrected confidence value.

4.4.1 Calculation of the Predictive Accuracy – Theoretical Issues

We are interested in the expected accuracy $E(c(r)|\hat{\mu}(r), s(X))$ of a rule $r X Y$ given its confidence $\hat{\mu}$ and the support of the rule body $s(X)$. Bayes formula shows us how to calculate that expectation.

$$E(c(r)|\hat{\mu}(r), s(X)) = \frac{\mathbf{P}(\hat{\mu}(r)|c(r), s(X))}{\mathbf{P}(c(r)|s(X))}$$

The likelihood $\mathbf{P}(\hat{\mu}(r)|c(r), s(X))$ can be modeled by a binomial distribution $\mathbf{B}[p, n](k) = \binom{n}{k}p^k(1-p)^{n-k}$. The correspondence to a coin flipping experiment, the standard example for a binomial distribution, can be easily seen. The rule $r$ either classifies the rule body $X$ correctly or not. The probability value $p$ for a correct prediction is just the predictive accuracy $c(r)$, the total number of coin flipping events $n$ matches $s(X)$, the total number of times the rule body occurs in the data set. The number of heads (or respectively tails) $k$ in the coin flipping experiments equals the number of database records which are correctly classified by the rule $r$. That is $\hat{\mu}(r)s(X)$ which corresponds to the support $s(r)$ of the whole rule $r$. Keeping that in mind, we now give the derivation of the expected accuracy as shown by Scheffer [123]. We provide it in this thesis, because of the central role the expected accuracy has in this approach. Furthermore this measure is the crucial difference to the Apriori algorithm.

Let $r$ be an arbitrary association rule of the form $X\Rightarrow Y$. The expectation value can be decomposed by integrating over all possible values of $c$

$$E(c(r)|\hat{\mu}(r), s(X)) = \int c(r)\mathbf{B}[p, n](k)dk$$

$$= \left\{ \begin{array}{lr} \int c(r)\binom{n}{k}p^k(1-p)^{n-k}dk & \text{if } 0 < p < 1, \\ 0 & \text{otherwise} \end{array} \right.$$
We can re-express this using Bayes formula:

\[
\left( \hat{\mathbf{c}}(r) = \mathbf{c}(r), \hat{\mathbf{s}}(X) \right)
\]

\[
\hat{\mathbf{c}}(r) = \mathbf{c}(r), \hat{\mathbf{s}}(X)
\]

\[
\frac{1}{\hat{\mathbf{c}}(r)} = \mathbf{c}(r), \hat{\mathbf{s}}(X)
\]

Remembering that \( P(\hat{\mathbf{c}}(r)|\mathbf{c}(r), \mathbf{s}(X)) \) is governed by a binomial distribution, we can restate this in the following way under the standard assumption of independent and identically distributed instances.

The last argument which applies, is that the distribution \( p(\mathbf{c}(r) = \mathbf{c} | \hat{\mathbf{c}}(r), \mathbf{s}(X)) \) has to integrate to 1.

\[
\frac{1}{\hat{\mathbf{c}}(r)} = \mathbf{c}(r), \hat{\mathbf{s}}(X)
\]

\[
\frac{1}{\left( \hat{\mathbf{c}}(r) \right)} = \mathbf{c}(r), \hat{\mathbf{s}}(X)
\]

\[
= \left( \hat{\mathbf{c}}(r), (\ ) \right) = 1
\]

\[
\Leftrightarrow \frac{1}{\hat{\mathbf{c}}(r)} = \mathbf{c}(r), \hat{\mathbf{s}}(X)
\]

\[
\Leftrightarrow \frac{1}{\hat{\mathbf{c}}(r)} = \mathbf{c}(r), \hat{\mathbf{s}}(X)
\]

Now we are able to express the expected accuracy as:

\[
\hat{\mathbf{c}}(r), \hat{\mathbf{s}}(X) = \frac{1}{\hat{\mathbf{c}}(r)}
\]
This equation calculates the expected accuracy over unseen instances given the support of the rule body and the confidence of the rule, given that the instances are independent and identically distributed. This expectation value is called predictive accuracy. The Predictive Apriori algorithm ranks rules according to this interestingness measure.

Table 4.1: Algorithm PredictiveApriori: discovery of \( n \) most predictive association rules.

1. **Input:** \( n \) (desired number of association rules), database with items \( a_1, \ldots, a_k \).

2. Let \( \tau = 1 \).

3. **For** \( i = 1 \ldots k \) **Do:** Draw a number of association rules \([x \Rightarrow y]\) with \( i \) items at random. Measure their confidence (provided \( s(x) > 0 \)). Let \( \pi_i(c) \) be the distribution of confidences.

4. **For** all \( c \), Let \( \pi(c) = \sum_{i=1}^{k} \pi_i(c)(\binom{k}{i})(2^i - 1) \sum_{i=1}^{k} \binom{k}{i} (2^i - 1) \)

5. Let \( X_0 = \{\emptyset\} \); Let \( X_1 = \{\{a_1\}, \ldots, \{a_k\}\} \) be all item sets with one single element.

6. **For** \( i = 1 \ldots k - 1 \) **While** \((i = 1 \text{ or } X_i - 1 \neq \emptyset)\).

   (a) **If** \( i > 1 \) **Then** determine the set of candidate item sets of length \( I \) as \( X_i = \{x \cup x' | x, x' \in X_i - 1, |x \cup x'| = i\} \). Generation of \( X_i \) can be optimized by considering only item sets \( x \) and \( x' \in X_i - 1 \) that differ only in the element with highest item index. Eliminate double occurrences of item sets in \( X_i \).

   (b) Run a database pass and determine the support of the generated item sets. Eliminate item sets with support less than \( \tau \) from \( X_i \).

   . (c) **For** all \( x \in X_i \) **Call** RuleGen(\( x \)).

   (d) **If** \( \text{best} \) has been changed, **Then** Increase \( \tau \) to be the smallest number such that \( E(c|1, \tau) \)
If $\tau > \text{database size}$, then Exit.

(c) If $\tau$ has been increased in the last step, then eliminate all item sets from $X_i$ which have support below $\tau$.

7. Output $best[1] \ldots best[n]$, the list of the $n$ best association rules.

**Table 4.2: Algorithm RuleGen: Searching efficiently for all rules with body $x$.**

<table>
<thead>
<tr>
<th>Algorithm RuleGen($x$) (find the best rules with body $x$ efficiently)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10. Let $\gamma$ be the smallest number such that $E(c</td>
</tr>
<tr>
<td>11. For $j = 1 \ldots k -</td>
</tr>
<tr>
<td>(a) If $j = 1$ Then Let $Y_1 = {a_1, \ldots, a_k} \setminus x$.</td>
</tr>
<tr>
<td>(b) Else Let $Y_j = {y \cup y'</td>
</tr>
<tr>
<td>(c) For all $y \in Y_j$ Do</td>
</tr>
<tr>
<td>i. Measure the support $s(x \cup y)$. If $s(x \cup y) \not\supseteq \mathcal{W}$, then eliminate $y$ from $Y_j$ and Continue the for loop with the next $y$.</td>
</tr>
<tr>
<td>ii. Calculate predictive accuracy $E(c</td>
</tr>
<tr>
<td>iii. If the predictive accuracy is among the $n$ best found so far (recorded in best), then update best, remove rules in best that are subsumed by other, at least equally accurate rules (utilize Theorem 1 and test for $x \subseteq x \cup y \supseteq y \supseteq y'$), and Increase $\gamma$ to be the smallest number such that $E(c</td>
</tr>
<tr>
<td>12. If any subsumed rule has been erased in 11(c) iii, then recur from step 10.</td>
</tr>
</tbody>
</table>
Association rule mining algorithms need to be configured before they are executed. So, the user has to give appropriate values for the parameters in advance (often leading to too many or too few rules) in order to obtain a good number of rules. Most of these algorithms require the user to set two thresholds, the minimal support and the minimal confidence, and then find all the rules that exceed the thresholds specified by the user. Therefore, the user must possess a certain amount of expertise in order to find the right settings for support and confidence to obtain the best rules.

One possible solution to this problem can be to use a parameter-free algorithm. Apriori algorithm that solves this problem partially. This algorithm reduces iteratively the minimum support, by a factor delta support ($\Delta s$) introduced by the user, until a minimum support is reached or a required number of rules has been generated.

Another improved version of the Apriori algorithm is the Predictive Apriori algorithm [123], which automatically resolves the problem of balance between these two parameters, maximizing the probability of making an accurate prediction for the data set. In order to achieve this, a parameter called the exact expected predictive accuracy is defined and calculated using the Bayesian method, which provides information about the accuracy of the rule found.

Data mining tool selection is normally initiated after the definition of problem to be solved and the related data mining goals. However, more appropriate tools and techniques can also be selected at the model selection and building phase. Selection of appropriate data mining tools and techniques depends on the main task of the data mining process. The selected software should be able to provide the required data mining functions and methodologies. The data mining functions that were to be carried out in this research project are clustering, to prepare the data for next algorithm, Classification and association rule mining, to find meaningful relationship in the dataset.
The data mining software selected for this research are Orange Canvas, Weka and Rapid Miner. Weka is developed at the University of Waikato in New Zealand. Weka stands for the Waikato Environment of Knowledge Analysis. The system is written in Java, an object-oriented programming language that is widely available for all major computer platforms, and Weka has been tested under Linux, Windows, and Macintosh operating systems. Java allows us to provide a uniform interface to many different learning algorithms, along with methods for pre and post processing and for evaluating the result of learning schemes on any given dataset. Weka expects the data to be fed into to be in ARFF format. It is necessary to have information about each attribute which cannot be automatically deduced form the attribute values [130].

Weka includes a variety of tools for preprocessing a dataset, such as attribute selection, attribute filtering and attribute transformation, feeding into a learning scheme, and analyze the resulting classifier and its performance. Weka is organized in packages that correspond to a directory hierarchy. The important packages of Weka are association, attribute selection, classifiers, clusterers, estimators, and filters packages. The association package has only one association rule mining algorithm, Apriori [130].

- Architecture and operating system. The computer architecture and the operating system on which the software runs should be first studied. Some data mining software operate on specific types of architecture and operating systems. In the research project, Orange Canvas, Rapid Miner and Weka operated of standalone and MS Windows operating system.

- Data sources- Specific data format on which the data mining software will operate is also another important factor to consider. The suitable data format for Orange Canvas, Rapid Miner and Weka data mining software are MS Access or MS Excel, CSV (comma separated text format) and ARFF formats respectively.
4.5 Discovery of understandable rules

A factor that is of major importance in determining the quality of the extracted rules is their comprehensibility. Although the main motivation for rule extraction is to obtain a comprehensible description of the underlying model's hypothesis, this aspect of rule quality is often overlooked due to the subjective nature of comprehensibility, which cannot be measured independently of the person using the system. Prior experience and domain knowledge of this person play an important role in assessing the comprehensibility. This contrasts with accuracy that can be considered as a property of the rules and which can be evaluated independently of the users.

There are some traditional techniques that have been used in order to improve the comprehensibility of discovered rules, such as, constraining the number of items in the antecedent or consequent of the rule, or performing a discretization of numerical values. Discretization divides the numerical data into categorical classes that are easier to understand for the teacher (categorical values are more user-friendly for the lecturer than precise magnitudes and ranges).

Lastly, we consider very important to mention another aspect that can facilitate the comprehensibility of discovered rules, the visualization. The goal of visualization is to help analysts in inspecting the data after to applying the mining task by means of some visual representation of the corpus of rules extracted. A range of visual representations is used, such as tables, two-dimensional matrices, graphs, bar charts, grids, mosaic plots, parallel coordinates, etc. Association rule mining can be integrated with visualization techniques in order to allow users to drive the association rule finding process, giving them control and visual cues to ease understanding of both the process and its results. However, the visualization methods are still difficult to understand for a non-expert in data mining, such as a teacher. Therefore, we consider this question need to be addressed in a near future, and the challenge it will be to apply these techniques in a more intuitive way identifying within these
data structures the rules that are relevant and meaningful in the context of the learning analysis, and representing them in a simple way such as icons, colors, etc, using interactive bi-dimensional and three-dimensional representations.

The process of applying association rule mining over the Education data consists of the same four steps as the general data mining process.

4.6 Data Understanding
This initial phase focus on understand the study objective and requirements from the student data. The data understanding phase starts with and initial data collection and proceeds with actions in order to get familiar with the data.

4.7 Collect Data.
The ED system is used by students and the usage and interaction information is stored in the database. We are going to use the students' usage data of the Educational system.

4.8 Data Preparation
The objective of this phase is to understand the structure and nature of the data in order to select what to be tested and how it should be tested. It includes those major steps, which are data transformation and attribute selection. The data set used in this study was obtained from Vikram University, Ujjain of course B.A., B.Sc., and B.Com., from session 2009 to 2010. Our objective is to use the Examination data of the student. The data is stored in a database: MS Excel, but it can also be used with Oracle, Access, Interface, any database supporting ODBC connections and others. We have used MS Excel because is the world's most popular database. Model has more than 14 tables. In a University results overall performance of a student is determined by internal assessment as well as external exam. Internal assessment is made on the bases of a student's assignment marks, class quiz, lab work, attendance previous year grade and his/her involvement in extra curriculum activities. While at the same time
external assessment of a student based on marks scored in final exam. In proposed model makes prediction about fail and pass ratio of students based on final exam.

4.8.1 Attribute Selection:
It is always better to scope the data set to only relevant attributes so that the set can be reduced to more manageable size and the running time of the software is minimized. However, this step is considered significant to the project. Many iteration of this step were carried out. At the end of the iteration, the possible choices of test are reviewed regarding the remained dataset. A brief summary of each iteration is described below:

a) **First Iteration:** Elimination all trivial attributes --- This iteration includes the following:

b) Eliminate all attributes that contained only null value ų This include only one attribute (Roll Number)

c) Eliminate all attributes that contained almost the same value in all records --- This includes attribute (Max marks in all subject)

d) Eliminate all attributes that is not important to the study --- This include only on attribute (Student Name)

Since the data mining software used to generate association rules accepts data only in arff format, the researcher first converted the data on Ms Excel file into comma separated text format and then to arff format. Data in raff format is then given to Weka software, Apriori algorithm and predictive Apriori for association rule mining. Data in arff format is then given to Weka software is shown below.
@relation BA1

@attribute BA1OBTN1 {27-53,54-80,0-26,81-100}

@attribute BA1OBTN2 {31-60,61-90,0-30,91-100}

@attribute BA1OBTN3 {27-53,54-80,0-26,81-100}

@attribute BA1OBTN4 {27-53,54-79,0-26,80-100}

@attribute GRANDBA1OBTN {103-204,205-306,0-102,307-600}

@attribute RESULT {FAIL,PASS,ATKT,ABST,W.H.}

<table>
<thead>
<tr>
<th>BA1OBTN1</th>
<th>BA1OBTN2</th>
<th>BA1OBTN3</th>
<th>GRANDBA1OBTN</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>27-53</td>
<td>31-60</td>
<td>27-53</td>
<td>103-204</td>
<td>FAIL</td>
</tr>
<tr>
<td>54-80</td>
<td>31-60</td>
<td>54-80</td>
<td>205-306</td>
<td>PASS</td>
</tr>
<tr>
<td>54-80</td>
<td>31-60</td>
<td>27-53</td>
<td>205-306</td>
<td>FAIL</td>
</tr>
<tr>
<td>27-53</td>
<td>31-60</td>
<td>27-53</td>
<td>103-204</td>
<td>FAIL</td>
</tr>
<tr>
<td>27-53</td>
<td>31-60</td>
<td>27-53</td>
<td>103-204</td>
<td>FAIL</td>
</tr>
<tr>
<td>54-80</td>
<td>31-60</td>
<td>54-80</td>
<td>205-306</td>
<td>PASS</td>
</tr>
<tr>
<td>27-53</td>
<td>31-60</td>
<td>27-53</td>
<td>103-204</td>
<td>FAIL</td>
</tr>
<tr>
<td>27-53</td>
<td>31-60</td>
<td>27-53</td>
<td>103-204</td>
<td>FAIL</td>
</tr>
<tr>
<td>27-53</td>
<td>31-60</td>
<td>27-53</td>
<td>103-204</td>
<td>FAIL</td>
</tr>
<tr>
<td>27-53</td>
<td>61-90</td>
<td>54-80</td>
<td>205-306</td>
<td>PASS</td>
</tr>
<tr>
<td>27-53</td>
<td>61-90</td>
<td>54-80</td>
<td>205-306</td>
<td>ATKT</td>
</tr>
<tr>
<td>54-80</td>
<td>61-90</td>
<td>54-80</td>
<td>205-306</td>
<td>PASS</td>
</tr>
<tr>
<td>54-80</td>
<td>61-90</td>
<td>27-53</td>
<td>205-306</td>
<td>ATKT</td>
</tr>
<tr>
<td>27-53</td>
<td>31-60</td>
<td>27-53</td>
<td>103-204</td>
<td>FAIL</td>
</tr>
</tbody>
</table>
4.9 The proposed Model

Data mining techniques have been applied in many application domains such as banking, fraud detection, and telecommunications [125]. Recently the data mining methodologies were used to enhance and evaluate the higher education tasks. Some researchers have proposed some methods and architectures for using data mining for higher education [126-129].

In this direction, some models have been proposed and implemented. The authors of [6] have proposed a model to represent how data mining can be used in a higher educational system to improve the efficiency and effectiveness of the traditional processes. In the model, several processes are proposed to be enhanced through data mining functions. The model is also presented as a guideline for higher educational system to improve the decision making processes. The research by [127] has used Rough Set theory as a classification approach to analyze student data where the Rosetta toolkit was used to evaluate the student data to describe different dependencies between the attributes and the student status. The discovered patterns are explained in plain English. The data set used in their experiments is the student data of Suranaree University of technology (SUT) during the academic year 2001-2002. The research by [128] describes the results of analyzing data from a large collection of the so called concurrent version system (CVS) created by many students working on a small set of identical projects (course assignments) in the second year undergraduate computer science course. The proposed model is used to extract all information of student behavior in writing the code of assignments and to find some statistical patterns or predictors that can be used to enhance students' performance in writing the code. The results obtained have suggested that aspects such as student work habits, and even code quality, have little bearing on the student's performance. The model of Delavari et al. in [9] is a motivation toward enhancing the proposed analysis model presented in [125] and that is used as a roadmap for the application of data mining in higher educational system. The enhanced model is named Data Mining for Higher Education. To prove the model correctness, one of the sub processes proposed by [125] has been implemented and evaluated. The model allows the decision
makers to better predict which students are less likely to perform well in that specific course, or those who are less likely to be successful in it. The research by Kalles and Pierrakeas in [129] discussed different machine learning techniques (decision trees, neural networks, Naive Bayes, instance-based learning, logistic regression and support vector machines) and compared them with genetic algorithm based induction of decision trees. They have discussed why the approach has a potential for developing into an alert tool. They have embarked in an effort to analyze students’ academic performance through the academic years, as measured by the students’ home work assignments, and attempted to derive short rules that explain and predict success or failure in the final exams. Students’ data are collected from the available data of the academic year 2009-2010. The latest version of WEKA machine learning toolkit [130] was used to evaluate and to experiment the proposed model.
The main objective of this work is to use data mining methodologies to study students' performance in the courses. Data mining provides many tasks that could be used to study the student performance. In this research, the association rule mining analysis between the failed course and suggests relevant causes of the failure to
improve the low capacity students performances, the classification task is used to evaluate student’s performance and as there are many approaches that are used for data classification, the decision tree method is used in this approach and the clustering method.

4.10 Result of Apriori Algorithm

=== Run information ===

Scheme: weka.associations.Apriori -N 1000 -T 0 -C 0.8 -D 0.05 -U 1.0 -M 0.2 -S -1.0 -c - 1

Relation: BA1

Instances: 8231

Attributes: 6

BA1OBTN1
BA1OBTN2
BA1OBTN3
BA1OBTN4
GRANDBA1OBTN
RESULT

=== Associator model (full training set) ===

Apriori

Minimum support: 0.2 (1646 instances)

Minimum metric <confidence>: 0.8

Number of cycles performed: 16

Generated sets of large itemsets:

Size of set of large itemsets L(1): 12
Size of set of large itemsets L(2): 33
Size of set of large itemsets L(3): 15
Size of set of large itemsets L(4):

Best rules found:

1. BA1OBTN1=54-80 BA1OBTN4=54-79 GRANDBA1OBTN=205-306 1763 ==> RESULT=PASS 1708 <conf:(0.97)> lift:(1.73) lev:(0.09) [723] conv:(13.9)

2. BA1OBTN1=54-80 BA1OBTN3=54-80 1842 GRANDBA1OBTN=205-306 ==> RESULT=PASS 1778 <conf:(0.97)> lift:(1.73) lev:(0.09) [749] conv:(12.51)

3. BA1OBTN2=61-90 BA1OBTN3=54-80 1741 ==> GRANDBA1OBTN=205-306 1662 <conf:(0.95)> lift:(1.71) lev:(0.08) [689] conv:(9.61)

4. BA1OBTN3=54-80 GRANDBA1OBTN=205-306 2267 ==> RESULT=PASS 2159 <conf:(0.95)> lift:(1.7) lev:(0.11) [892] conv:(9.18)

5. BA1OBTN1=54-80 2453 GRANDBA1OBTN=205-306 ==> RESULT=PASS 2329 <conf:(0.95)> lift:(1.7) lev:(0.12) [958] conv:(8.66)

6. BA1OBTN4=54-79 2179 GRANDBA1OBTN=205-306 ==> RESULT=PASS 2064 <conf:(0.95)> lift:(1.7) lev:(0.1) [846] conv:(8.29)

7. BA1OBTN1=54-80 BA1OBTN4=54-79 2323 ==> GRANDBA1OBTN=205-306 2189 <conf:(0.94)> lift:(1.69) lev:(0.11) [891] conv:(7.6)

8. BA1OBTN3=54-80 BA1OBTN4=54-79 2013 ==> GRANDBA1OBTN=205-306 1885 <conf:(0.94)> lift:(1.68) lev:(0.09) [760] conv:(6.89)

9. BA1OBTN2=61-90 GRANDBA1OBTN=205-306 1929 ==> RESULT=PASS 1800 <conf:(0.93)> lift:(1.67) lev:(0.09) [722] conv:(6.55)
10. BA1OBTN1=54-80  BA1OBTN2=61-90  2182 ==> GRANDBA1OBTN=205-306 2031
   <conf:(0.93)> lift:(1.67) lev:(0.1) [812] conv:(6.34)

11. BA1OBTN1=27-53 GRANDBA1OBTN=103-204 1814 ==> RESULT=FAIL 1681
   <conf:(0.93)> lift:(2.29) lev:(0.12) [946] conv:(8.06)

12. RESULT=PASS 3058 ==> GRANDBA1OBTN=205-306 2831
   <conf:(0.93)> lift:(1.66) lev:(0.14) [1122] conv:(5.92)

13. BA1OBTN3=27-53 GRANDBA1OBTN=103-204 1872 ==> RESULT=FAIL 1724
   <conf:(0.92)> lift:(2.27) lev:(0.12) [966] conv:(7.48)

14. BA1OBTN4=27-53 GRANDBA1OBTN=103-204 1962 ==> RESULT=FAIL 1803
   <conf:(0.92)> lift:(2.27) lev:(0.12) [1008] conv:(7.3)

15. BA1OBTN1=54-80  BA1OBTN3=54-80  2551 ==> GRANDBA1OBTN=205-306 2338
   <conf:(0.92)> lift:(1.64) lev:(0.11) [912] conv:(5.26)

16. GRANDBA1OBTN=103-204 2690 ==> RESULT=FAIL 2415
   <conf:(0.9)> lift:(2.22) lev:(0.16) [1326] conv:(5.8)

17. BA1OBTN2=31-60 GRANDBA1OBTN=103-204 1881 ==> RESULT=FAIL 1685
   <conf:(0.9)> lift:(2.21) lev:(0.11) [923] conv:(5.68)

18. BA1OBTN2=61-90 2910 ==> GRANDBA1OBTN=205-306 2568
   <conf:(0.88)> lift:(1.58) lev:(0.11) [942] conv:(3.74)

19. BA1OBTN4=54-79 3301 ==> GRANDBA1OBTN=205-306 2820
   <conf:(0.85)> lift:(1.53) lev:(0.12) [975] conv:(3.02)

20. BA1OBTN1=27-53 GRANDBA1OBTN=103-204 1968 ==> RESULT=FAIL 1681
   <conf:(0.85)> lift:(2.61) lev:(0.13) [1037] conv:(4.6)
21. BA1OBTN4=54-79 GRANDBA1OBTN=205-306 BA1OBTN1=54-80 2064 ==> RESULT=PASS 1708 <conf:(0.83)> lift:(1.53) lev:(0.07) [592] conv:(2.66)

22. BA1OBTN3=54-80 GRANDBA1OBTN=205-306 BA1OBTN1=54-80 2159 ==> RESULT=PASS 1778 <conf:(0.82)> lift:(1.52) lev:(0.07) [611] conv:(2.6)

23. GRANDBA1OBTN=205-306 BA1OBTN1=54-80 2831 ==> RESULT=PASS 2329 <conf:(0.82)> lift:(1.52) lev:(0.1) [798] conv:(2.59)

24. BA1OBTN3=54-80 3737 ==> GRANDBA1OBTN=205-306 3074 <conf:(0.82)> lift:(1.47) lev:(0.12) [986] conv:(2.48)

25. BA1OBTN2=31-60 GRANDBA1OBTN=103-204 2051 ==> RESULT=FAIL 1685 <conf:(0.82)> lift:(2.51) lev:(0.12) [1014] conv:(3.76)

26. BA1OBTN3=54-80 RESULT=PASS 2267 ==> BA1OBTN1=54-80 1842 <conf:(0.81)> lift:(1.5) lev:(0.07) [616] conv:(2.45)

27. BA1OBTN4=54-79 RESULT=PASS 2179 ==> BA1OBTN1=54-80 1763 conf:(0.81)> lift:(1.5) lev:(0.07) [585] conv:(2.4)

**Meaning of the parameters mentioned above**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>-N (required number of rules output)</td>
<td>1000</td>
</tr>
<tr>
<td>-T (metric type by which to rank rules)</td>
<td>0</td>
</tr>
<tr>
<td>-C (the minimum confidence of a rule)</td>
<td>0.8</td>
</tr>
<tr>
<td>D (delta at which the minimum support is decreased at each iteration)</td>
<td>0.05</td>
</tr>
<tr>
<td>-U (upper bound for minimum support)</td>
<td>1.0</td>
</tr>
<tr>
<td>-M (the lower bound for the minimum support)</td>
<td>0.2</td>
</tr>
</tbody>
</table>
Support- The support for a rule $A \Rightarrow B$ is obtained by dividing the number of transactions which satisfy the rule, $N\{A=>B\}$, by the total number of transactions, $N$.

$$N\{A=>B\} / N$$

The support is therefore the frequency of events for which both the LHS and RHS of the rule hold true. The higher the support the stronger the information that both type of events occur together.

Confidence- The confidence of the rule $A \Rightarrow B$ is obtained by dividing the number of transactions which satisfy the rule $N\{A=>B\}$ by the number of transactions which contain the body of the rule, $A$.

$$N\{A=>B\} / N\{A\}$$

The confidence is the conditional probability of the RHS holding true given that the LHS holds true. A high confidence that the LHS event leads to the RHS event implies causation or statistical dependence.

Lift- The lift of the rule $A \Rightarrow B$ is the deviation of the support of the whole rule from the support expected under independence given the supports of the LHS ($A$) and the RHS ($B$).

$$N\{A=>B\} / N\{A\} / N\{B\}$$

Lift is an indication of the effect that knowledge that LHS holds true has on the probability of the RHS holding true. Hence Lift is a value that gives us information about the increase in Probability of the "then" (consequent RHS) given the "if" (antecedent LHS) part.

- when lift is exactly 1: No effect (LHS and RHS independent). No relationship between Events.
For lift greater than 1: Positive effect (given that the LHS holds true, it is more likely that the Operational risk management RHS holds true). Positive dependence between events.

If lift is smaller than 1: Negative effect (when the LHS holds true, it is less likely that the RHS holds true). Negative dependence between events.

**Leverage** — proportion of additional examples covered by both the antecedent and the consequent above those expected if the antecedent and consequent were independent of each other, and finally

**Conviction** — a measure similar to Leverage that measures the departure from independence.

In selecting rules for discussion, the researcher focused on the rules generated from the superset of frequent or large item set consisting of the highest size of large item set. The following are rules selected for discussion from experiment 1:

1. **BA1OBTN1=54-80 BA1OBTN4=54-79 GRANDBA1OBTN=205-306 1763 => RESULT=PASS 1708**<conf:(0.97)> lift:(1.73) lev:(0.09) [723] conv:(13.9)

The meaning of this rules is if a student of B.A. IST year getting the marks in OBTN1 from 54-80, OBTN4 from 54-79 and GRANDOBTN from 205-306 then this student will also have PASS. The support for this rule can be computed by dividing the figure on the right-hand-side of the rule 1708 by the total number of instances considered in generating association rules, 8231. This rule has a support of 20%. The number 1708 on the right-hand-side of the rule indicates the number of items covered by its antecedent. The confidence is also computed by dividing the figure on the left-hand-side of the rule by the figure on the right-hand-side of the rule. Following the rule is the number of those items for which the rule's consequent holds as well. Lift greater than 1: Positive effect (given that the LHS holds true, it
is more likely that the Operational risk management RHS holds true). This rule has a Lift of 1.73 which means that positive dependence between events.

11. BA1OBTN1=27-53 GRANDBA1OBTN=103-204 1814 ==> RESULT=FAIL 1681
<conf:(0.93)> lift:(2.29) lev:(0.12) [946] conv:(8.06)

13. BA1OBTN3=27-53 GRANDBA1OBTN=103-204 1872 ==> RESULT=FAIL 1724
<conf:(0.92)> lift:(2.27) lev:(0.12) [966] conv:(7.48)

14. BA1OBTN4=27-53 GRANDBA1OBTN=103-204 1962 ==> RESULT=FAIL 1803
<conf:(0.92)> lift:(2.27) lev:(0.12) [1008] conv:(7.3)

The meaning of this rules is if a student of BA 1\textsuperscript{st} year getting the marks in OBTN1 from 27-53, OBTN 3 from 27-53, OBTN4 from 27-53 and GRANDOBTN from 103-204 then this student will have fail.

There are also a lot of uninteresting rules, like a great number of redundant rules (rules with a generalization of relationships of several rules, like rule 1 with rules 2 and 3). There are some similar rules (rules with the same element in antecedent and consequent but interchanged). And there are some random relationships (rules with random relations between variables).

But there are also rules that show relevant information for educational purposes, which can be very useful for the teacher in decision making about the activities and detecting students with learning problems. Starting from this information, the teacher can pay more attention to these students because they are prone to failure. As a result, the teacher can motivate them in time to pass the course.
4.11 Result of PredictiveApriori

=== Run information ===
Relation:     BA1
Instances:    8231
Attributes:   6

BA1OBTN1
BA1OBTN2
BA1OBTN3
BA1OBTN4
GRANDBA1OBTN
RESULT

=== Associator model (full training set) ===
PredictiveApriori
==================
Best rules found:

1. BA1OBTN2=61-90 GRANDBA1OBTN=307-600 129 ==> RESULT=PASS 129
   acc:(0.99491)

2. BA1OBTN1=54-80 GRANDBA1OBTN=307-600 122 ==> RESULT=PASS 122
   acc:(0.99489)

3. BA1OBTN1=54-80 BA1OBTN2=61-90 BA1OBTN3=81-100 BA1OBTN4=54-79 114
   ==> RESULT=PASS 114   acc:(0.99486)

4. BA1OBTN3=81-100 GRANDBA1OBTN=307-600 113 ==> RESULT=PASS 113
   acc:(0.99486)

5. BA1OBTN1=54-80 BA1OBTN2=61-90 BA1OBTN3=54-80 BA1OBTN4=80-100 107
   ==> RESULT=PASS 107   acc:(0.99483)

6. BA1OBTN4=80-100 GRANDBA1OBTN=307-600 101 ==> RESULT=PASS 101
   acc:(0.9948)

7. BA1OBTN3=27-53 BA1OBTN4=0-26 GRANDBA1OBTN=0-102 90 ==> RESULT=FAIL 90   acc:(0.99472)

8. GRANDBA1OBTN=307-600 222 ==> RESULT=PASS 220   acc:(0.99468)

9. BA1OBTN2=61-90 BA1OBTN3=81-100 BA1OBTN4=54-79 143 ==> RESULT=PASS 142
   acc:(0.99462)

10. BA1OBTN1=0-26 BA1OBTN3=27-53 BA1OBTN4=0-26 77 ==> RESULT=FAIL 77
    acc:(0.99458)

11. BA1OBTN2=61-90 BA1OBTN3=54-80 BA1OBTN4=80-100 121 ==> RESULT=PASS
120  acc:(0.99429)
12. BA1OBTN1=81-100  BA1OBTN2=61-90  BA1OBTN3=54-80  BA1OBTN4=54-79  62
   => RESULT=PASS 62  acc:(0.99426)
13. BA1OBTN3=27-53  GRANDBA1OBTN=0-102  115  ==> RESULT=FAIL 114
   acc:(0.99415)
14. BA1OBTN1=54-80  BA1OBTN2=91-100  BA1OBTN3=54-80  BA1OBTN4=54-79  49
   ==> RESULT=PASS 49  acc:(0.9937)
15. BA1OBTN2=91-100  BA1OBTN4=80-100  46  ==> RESULT=PASS 46  acc:(0.99351)
16. BA1OBTN2=91-100  BA1OBTN3=54-80  BA1OBTN4=54-79  GRANDBA1OBTN=205-306
   42  ==> RESULT=PASS 42  acc:(0.99318)
17. BA1OBTN1=0-26  BA1OBTN2=0-30  BA1OBTN3=54-80  38  ==> RESULT=FAIL 38
   acc:(0.99277)
18. BA1OBTN1=81-100  BA1OBTN3=81-100  BA1OBTN4=54-79  35  ==> RESULT=PASS
   35  acc:(0.99237)
19. BA1OBTN2=0-30  BA1OBTN3=0-26  BA1OBTN4=27-53  GRANDBA1OBTN=0-102
   30  ==> RESULT=FAIL 30  acc:(0.99146)
20. BA1OBTN1=81-100  BA1OBTN2=91-100  BA1OBTN3=81-100  27  ==> RESULT=PASS
   27  acc:(0.99072)
21. BA1OBTN1=0-26  BA1OBTN3=54-80  BA1OBTN4=27-53  26  ==> RESULT=FAIL 26
   acc:(0.99042)
22. BA1OBTN2=61-90  BA1OBTN3=81-100  BA1OBTN4=80-100  26  ==> RESULT=PASS
   26  acc:(0.99042)
23. BA1OBTN1=0-26  BA1OBTN2=0-30  BA1OBTN3=0-26  BA1OBTN4=27-53
   26  ==> RESULT=FAIL 26  acc:(0.99042)
24. BA1OBTN2=91-100  BA1OBTN3=54-80  BA1OBTN4=54-79  68  ==> RESULT=PASS
   67  acc:(0.99036)
25. BA1OBTN1=81-100  BA1OBTN3=81-100  BA1OBTN4=80-100  22  ==> RESULT=PASS
   22  acc:(0.98892)
26. BA1OBTN2=0-30  BA1OBTN3=27-53  BA1OBTN4=0-26  164  ==> RESULT=FAIL 161
   acc:(0.98828)
27. BA1OBTN3=54-80  GRANDBA1OBTN=0-102  20  ==> RESULT=FAIL 20
   acc:(0.98797)
28. BA1OBTN2=0-30  BA1OBTN4=27-53  GRANDBA1OBTN=0-102  56  ==> RESULT=FAIL
   55  acc:(0.98713)
29. BA1OBTN1=81-100  BA1OBTN3=54-80  BA1OBTN4=80-100  18  ==> RESULT=PASS
   18  acc:(0.98663)
30. BA1OBTN1=0-26  BA1OBTN3=54-80  BA1OBTN4=0-26  17  ==> RESULT=FAIL 17
   acc:(0.98587)

87
<table>
<thead>
<tr>
<th>Test Case</th>
<th>BA1OBTN1</th>
<th>BA1OBTN2</th>
<th>BA1OBTN3</th>
<th>BA1OBTN4</th>
<th>GrandBA1OBTN</th>
<th>Result</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>27-53</td>
<td>0-30</td>
<td>0-102</td>
<td>52</td>
<td>FAIL</td>
<td></td>
<td>0.98551</td>
</tr>
<tr>
<td>32</td>
<td>54-80</td>
<td>0-30</td>
<td>54-79</td>
<td>205-306</td>
<td>FAIL</td>
<td>16</td>
<td>0.985</td>
</tr>
<tr>
<td>33</td>
<td>81-100</td>
<td>61-90</td>
<td>54-79</td>
<td>95</td>
<td>PASS</td>
<td>93</td>
<td>0.98458</td>
</tr>
<tr>
<td>34</td>
<td>27-53</td>
<td>0-26</td>
<td>0-102</td>
<td>49</td>
<td>FAIL</td>
<td></td>
<td>0.98405</td>
</tr>
<tr>
<td>35</td>
<td>27-53</td>
<td>0-30</td>
<td>27-53</td>
<td>190</td>
<td>FAIL</td>
<td>186</td>
<td>0.98139</td>
</tr>
<tr>
<td>36</td>
<td>54-80</td>
<td>61-90</td>
<td>27-53</td>
<td>103-204</td>
<td>FAIL</td>
<td>12</td>
<td>0.97992</td>
</tr>
<tr>
<td>37</td>
<td>81-100</td>
<td>80-100</td>
<td>80-100</td>
<td>41</td>
<td>PASS</td>
<td>40</td>
<td>0.97868</td>
</tr>
<tr>
<td>38</td>
<td>0-26</td>
<td>0-26</td>
<td>103-204</td>
<td>11</td>
<td>FAIL</td>
<td>11</td>
<td>0.97804</td>
</tr>
<tr>
<td>39</td>
<td>0-26</td>
<td>0-30</td>
<td>27-53</td>
<td>78</td>
<td>FAIL</td>
<td>76</td>
<td>0.97675</td>
</tr>
<tr>
<td>40</td>
<td>0-26</td>
<td>27-53</td>
<td>0-102</td>
<td>38</td>
<td>FAIL</td>
<td>37</td>
<td>0.97592</td>
</tr>
<tr>
<td>41</td>
<td>54-80</td>
<td>0-26</td>
<td>0-26</td>
<td>9</td>
<td>FAIL</td>
<td>9</td>
<td>0.97299</td>
</tr>
<tr>
<td>42</td>
<td>27-53</td>
<td>0-26</td>
<td>0-102</td>
<td>35</td>
<td>FAIL</td>
<td>34</td>
<td>0.97262</td>
</tr>
<tr>
<td>43</td>
<td>0-26</td>
<td>0-30</td>
<td>27-53</td>
<td>113</td>
<td>FAIL</td>
<td>110</td>
<td>0.97189</td>
</tr>
<tr>
<td>44</td>
<td>27-53</td>
<td>0-30</td>
<td>27-53</td>
<td>103-204</td>
<td>FAIL</td>
<td>278</td>
<td>0.97076</td>
</tr>
<tr>
<td>45</td>
<td>27-53</td>
<td>0-30</td>
<td>0-26</td>
<td>33</td>
<td>FAIL</td>
<td>32</td>
<td>0.97004</td>
</tr>
<tr>
<td>46</td>
<td>54-80</td>
<td>91-100</td>
<td>81-100</td>
<td>54-79</td>
<td>PASS</td>
<td>8</td>
<td>0.96952</td>
</tr>
<tr>
<td>47</td>
<td>27-53</td>
<td>0-102</td>
<td>0-102</td>
<td>66</td>
<td>FAIL</td>
<td>64</td>
<td>0.96835</td>
</tr>
<tr>
<td>48</td>
<td>0-30</td>
<td>27-53</td>
<td>27-53</td>
<td>103-204</td>
<td>FAIL</td>
<td>258</td>
<td>0.96797</td>
</tr>
<tr>
<td>49</td>
<td>0-30</td>
<td>27-53</td>
<td>27-53</td>
<td>285</td>
<td>FAIL</td>
<td>277</td>
<td>0.96635</td>
</tr>
</tbody>
</table>
In order to compare the performance of Apriori and Predictive Apriori, we use a uniform measure that is independent of implementation details. For Apriori, we count how many association rules have to be compared against the minconf threshold. This comparison in the innermost loop is the bottleneck of Apriori; the number is independent of the actual minconf threshold. We can determine the number of comparisons from the item sets without actually enumerating all rules. Whereas Predictive Apriori we measure for how many rules we need to determine the accuracy. The Predictive Apriori algorithm returns the $n$ rules which maximize the expected predictive accuracy; the user only has to specify how many rules he or she wants to be presented. This is perhaps a more natural parameter than minsup and minconf, required by the Apriori algorithm.

With all these observations, if academic planners can make use of the extracted hidden patterns from students' failed causes using Apriori and Predictive Apriori association rule mining approach, it will surely help in curriculum re-structuring and also, help in monitoring the students' ability. This will enable the academic advisers to guide students properly on courses they should enroll for. This, eventually, tends to increase the student pass rate.

After applying the Apriori and Predictive Apriori algorithm was executed on the graduation data set of University examination we obtained results for the training data. The rules were generated in order to interpret the importance of a rule. Apriori algorithm and Predictive Apriori algorithm is implemented to generate the hidden pattern from the students failed course dataset which when analyzed will serve as a strong convincing recommendation to academic planning department in institutions of learning for curriculum structure and modification in order to improve the students' performances and minimize failure rate percentage. Some rules of which are presented in Appendix A.
4.12 Conclusion

With all these observations, if academic planners can make use of the extracted hidden patterns from students’ failed causes using association rule mining approach, it will surely help in curriculum re-structuring and also, help in monitoring the students’ ability. This will enable the academic advisers to guide students properly on courses they should enroll for. This, eventually, tends to increase the student pass rate.

This study has bridge the gap in educational data analysis and shows the potential of the association rule mining algorithm for enhancing the effectiveness of academic planners and level advisers in higher institutions of leaning. The analysis was done using undergraduate students’ result in the university. The university offers three programmes; B.A., B.Com. and B.Sc. The analysis reveals that there is more to students’ failure than the students’ ability. It also reveals some hidden patterns of students’ failed courses which could serve as bedrock for academic planners in making academic decisions and an aid in the curriculum re-structuring and modification with a view to improving students’ performance and reducing failure rate. To adopt this approach a larger number of students should be considered from the first year to the final year in the institution. This will surely reveal more interesting patterns. Also, the min. confidence should be of a higher percentage to be able to have more relevant and constructive rules. In future applications, in order to improve the comprehensibility and applicability of the association rules, it will be very useful to also provide an ontology that would describe the content of the courses which will allow the academic planners to understand better the rules that contain concepts related to the analyzed domain.