B.Sc. in Computer Science and Engineering Thesis

Analysis of Student Performance using Data Mining

Submitted by

Suraiya Yeasmin
ID: 201114032

Rubayat Jinnah
ID: 201114038

Atoshi Islam
ID: 201114055

Supervised by

Dr. Syed Akhter Hossain

Professor and Head of the Department

Department of Computer Science and Engineering

Daffodil International University(DIU)

Dhaka, Bangladesh

Department of Computer Science and Engineering
Military Institute of Science and Technology

December 2014

CERTIFICATION

This thesis paper titled **“Analysis of Student Performance using Data Mining”**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in December 2014.

Group Members:

Suraiya Yeasmin

Rubayat Jinnah

Atoshi Islam

Supervisor:

Dr. Syed Akhter Hossain
Professor and Head of the Department
Department of Computer Science and Engineering
Daffodil International University(DIU)
Dhaka, Bangladesh

CANDIDATES’ DECLARATION

This is to certify that the work presented in this thesis paper, titled, “Analysis of Student Performance using Data Mining”, is the outcome of the investigation and research carried out by the following students under the supervision of Dr. Syed Akhter Hossain, Professor and Head of the Department, Department of Computer Science and Engineering, Daffodil International University(DIU), Dhaka, Bangladesh.

It is also declared that neither this thesis paper nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

____

Suraiya Yeasmin
ID: 201114032

____

Rubayat Jinnah
ID: 201114038

____

Atoshi Islam
ID: 201114055

ACKNOWLEDGEMENT

We are thankful to Almighty Allah for his blessings for the successful completion of our thesis. Our heartiest gratitude, profound indebtedness and deep respect go to our supervisor, Dr. Syed Akhter Hossain, Professor and Head of the Department, Department of Computer Science and Engineering, Daffodil International University(DIU), Dhaka, Bangladesh, for his constant supervision, affectionate guidance and great encouragement and motivation. His keen interest on the topic and valuable advices throughout the study was of great help in completing thesis.

We are especially grateful to the Department of Computer Science and Engineering (CSE) of Military Institute of Science and Technology (MIST) for providing their all out support during the thesis work.

Finally, we would like to thank our families and our course mates for their appreciable assistance, patience and suggestions during the course of our thesis.

Dhaka
December 2014

Suraiya Yeasmin

Rubayat Jinnah

Atoshi Islam

ABSTRACT

The aim of this thesis paper is to analyze student performance using data mining. Data mining is the process of prediction, extracting data. Prediction regarding student performance can help a student to take decision. It can help not only the current students but also the future students, to take decision. In this way they can avoid poor performance which will help to enhance their performance. This is also a guideline to take decision. To understand student performance, a survey was conducted by Military Institute of Science Technology (MIST) with the support from the CSE department and the peer learners of different classes. The data collected from survey was normalized, validated and revalidated. After thorough investigation on the survey data, based on statistical analysis techniques, differnt observations were recorded in the form of graphical illlustration in order to find the relations. The experimental analysis of the data through result form the survey was satisfactory whicn led towards further study. In order to proceed further through data mining based on the understanding of the survey, data was collected form the central databse of MIST where the main aim was to relate CGPA and student performance. We investigated different properties of the data; collected and developed a classification hypothesis in order to apply data mining algorithms. In this reasearch a machine learning tool called WEKA develop by the university of New Zealand was used for testing different algorithms on the data. It is to be noted successful appliaction of data minig algorithms in weka requires careful analysis of the source data and develops very specific classes using a very specialized format called ARFF (Attribute related file format) format, which defines attributes a targeted class based on data observations. The experimental results are validated against test data and intersting co-relations are observed. In the future further regorous study to match between demographic data and academic data will lead to much determining factors in order to predict the student performance.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

List of Algorithms

LIST OF ABBREVIATION

MIST	: Military Institute of Science And Technology
KDD	: Knowledge Discovery
EDM	: Educational Data Mining

CHAPTER 1
INTRODUCTION

1.1 Objective

As we enter the twenty-first century, it may not be a new message that the importance of engineering education is growing across the whole world. Technological advancements are growing. Newer and newer inventions are happening day by day. For this engineering education is important. Because engineers are vital to our economy and society. Their knowledge and skills are highly demandable. In Bangladesh advancements of technology is also remarkable. Engineering education is getting a new dimension. A large number of students are interested to admit in different engineering universities. It is a very good sign, but we need to ensure that the capability of becoming an enginneer will be enhanced. In this research paper we study on competency model of students in enginneering education in Bangladesh. The academic performances of students at the engineering universities have come under the spotlight for a various reason. The purpose of this study is to initiate a discussion on the possible factors and addressing them in a paradigm so that the academic performance can be increased. A competency model is a framework of organising a collection of observable performance of people. Competency modelling can help us to find a strategical factors. A number of studies have been done in various country, but in Bangladesh till now no initiatives have been taken. We study those factors by reviewing different university students data and try to paradigm those factors in a competency model. Our study focus on pivotal causes which can be related to academic performance. By this fact-finding study we can analyse and identify the factors that influence the academic performance.

1.2 Motivation

1.3 Layout of the report

Chapter 2 introduces with data mining. It gives the basic idea of machine learning, supervised learning and unsupervised learning, classification and clustering. Some related algorithms are also discussed here.

Chapter 3 discusses about those works of data mining related with student performance that has been done till now.

Chapter 4 discusses details about the survey that was done.It introduces WEKA, a software written in JAVA that we have used.

Chapter 5 discusses with experiment and results of our survey.

Chapter 6 concludes the thesis and discusses about the future scope.

CHAPTER 2
INTRODUCTION TO DATA MINING

2.1 Data mining

Data mining is the other name of knowledge discovery. It is the computer-assisted process of digging through and analyzing enormous sets of data and then extracting the meaning of the data. Here data are analyzed from different perspectives and relationships are established among them which gradually turns into knowledge that is used in many research areas, including mathematics, artificial intelligence, cybernetics, genetics, marketing etc.

The most basic forms of data for mining comes from: database data, data warehouse and transactional data. The amount of raw data stored in corporate databases is exploding. In today's fiercely competitive business environment, companies need to rapidly turn their gigabytabytes of raw data into significant insights into their customers and markets to guide their marketing, investment, and management strategies. Data mining is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses and teres.

Applying data mining methodologies on the educational data has brought a new research discipline. Many of the institutions and other university systems around the globe have tried to overcome the problems of identifying actual student needs through learning analytics. With the help of data mining by analyzing their learning environment and other behavioural factors teachers will be able to provide necessary guidance to improve their capabilities or learning capacities.

2.2 Machine Learning

Machine Learning is a natural outgrowth of the intersection of Computer Science and Statistics. It deals with the construction and study of systems that can be learnt from data that offers a useful ways to approach problems, otherwise defy manual solution. Machine learning algorithms can figure out how to perform important tasks by generalizing from examples. This is often feasible and cost-effective where manual programming is not. One measure of progress in Machine Learning is its significant real-world applications, such as Speech recognition, Computer vision, Bio-surveillance, robot control etc. Machine learning algorithms can be organized into a taxonomy based on the desired outcome of the algorithm or the type of input available during training of the machine. Some popular algorithms of machine learning are Supervised Learning, Unsupervised Learning, Semi-Supervised Learning and Reinforcement Learning [2].

Machine learning and Data mining, these two terms are sometimes confused as they often employ the same methods and overlap significantly. Normally they can be defined as:

- Machine learning is the task of building knowledge and storing it in some form in the computer, based on known properties learned from the training data.

- Data mining focuses on the discovery of (previously) unknown properties in the data and helps us in building models to detect the patterns that allow us to classify or predict situations given an amount of facts or factors.

2.3 Basic Representation

Data mining is also known as KDD (Knowledge Discovery in Databases). 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 [3]

Data pre-processing is an important step in data mining technique. It involves transforming raw data into an understandable format. Data goes through a series of steps during preprocessing:

1. Data cleaning: Data is cleansed through processes such as filling in missing values, smoothing the noisy data or resolving the inconsistencies in the data.
2. Data integration: Data with different representations are put together and conflicts within the data are resolved, e.g. using multiple databases or files.

3. Data transformation: Data is normalized, aggregated and generalized.
4. Data reduction: Aims to present a reduced representation of the data in a data warehouse.
5. Data discretization: This step involves the number of values of a continuous attribute by dividing the range of attribute intervals.

Most data-mining methods are based on tried and tested techniques from machine learning, pattern recognition, and statistics: classification, clustering, regression and so on. The array of different algorithms under each of these headings can often be bewildering to both the novice and the experienced data analyst.

Four basic components in each algorithm are:

1. Model or Pattern Structure: Determining underlying structure or functional form, we seek from data
2. Score Function: Judging the quality of the fitted model
3. Optimization and Search Method: Searching over different model and pattern structures
4. Data Management Strategy: Handling data access efficiently

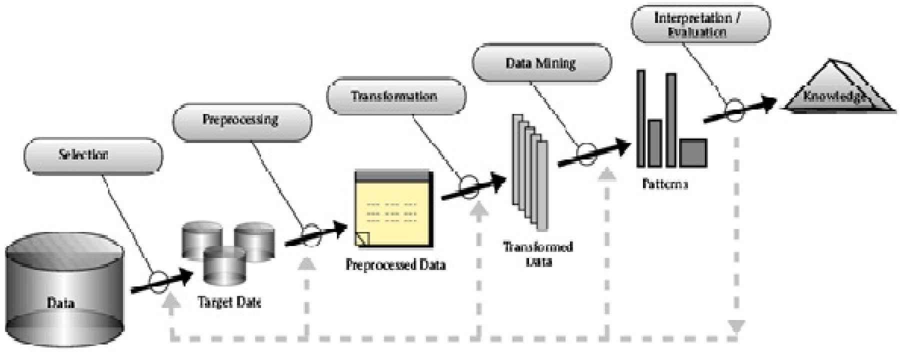


Figure 2.1: an overview of data mining process [1]

The process can be represented like this: [4]

Extract, transform, and load transaction data onto the data warehouse system ⇒ Store and manage the data in a multidimensional database system ⇒ Provide data access to business analysts and information technology professionals ⇒ Analyze the data by application software ⇒ Present the data in a useful format

2.4 Classification

Data mining offers promising ways to uncover hidden patterns within large amounts of data. These hidden patterns can potentially be used to predict future behavior. Classification consists of predicting a certain outcome based on a given input. It is often referred to as supervised learning because the classes are determined before examining the data. Classification creates a function from training data that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. The classification algorithm described in [5], offers an interesting combination of approaches. It consists of main GP (Genetic programming) algorithm, where each individual represents an IF-THEN prediction rule, having the rule modeled as a Boolean expression tree [6].

2.5 Clustering

Clustering is the process of making group of abstract objects into classes of similar objects. It is similar to classification except that the groups are not predefined, but rather defined by the data alone. In the context of machine learning, clusters correspond to hidden pattern, the search for clusters is unsupervised learning, and the resulting system represents a data concept. As a data mining function, cluster analysis serve as a tool to gain insight into the distribution of data to observe characteristics of each cluster.

Classification and clustering can be seemed similar because both data mining algorithms essentially divide the datasets into sub-datasets. But there is a noticeable difference between them.

A machine learning algorithm for face recognition can be defined by the following example: Suppose, you will show several images of faces and not-faces and a good algorithm will eventually learn and be able to predict whether or not an unseen image is a face. This example of face recognition is supervised, which means that your examples must be labeled, or explicitly say which ones are faces and which ones are not. The task that is done here can be an example of classification. In an unsupervised algorithm your examples are not labeled, i.e. you do not say anything about face. Of course in such a case the algorithm itself cannot invent what a face is, but it could be able to cluster the data in different classes; e.g. it could be able to distinguish that faces are very different from human. This is called clustering.

In general, in classification you have a set of predefined classes and want to know which class a new object belongs to. Whereas Clustering tries to group a set of objects and find whether there is some relationship between the objects.

2.6 Related Algorithms

2.6.1 K-means Algorithm

The k-means algorithm is a simple iterative method to partition a given dataset into a user-specified number of clusters, k. This algorithm has been discovered by several researchers across different disciplines, most notably Lloyd [**?**] , Forgey, Friedman and Rubin, and McQueen. A detailed history of k-means alongwith descriptions of several variations are given in [7]. Gray and Neuhoff [8] provide a nice historical background for k-means placed in the larger context of hill-climbing algorithms.The algorithm operates on a set of d-dimensional vectors, $D = x_i | i = 1, \ldots, N$, where $x_i \in R^d$ denotes the ith data point. The algorithm is initialized by picking k points in R^d as the initial k cluster representatives or centroids. Techniques for selecting these initial seeds include sampling at random from the dataset, setting them as the solution of clustering a small subset of the data or perturbing the global mean of the data k times. Then the algorithm iterates between two steps till convergence:

Step 1: Data Assignment. Each data point is assigned to its closest centroid, with ties broken arbitrarily. This results in a partitioning of the data.

Step 2: Relocation of means. Each cluster representative is relocated to the center (mean) of all data points assigned to it. If the data points come with a probability measure(weights), then the relocation is to the expectations (weighted mean) of the data partitions.

The algorithm converges when the assignments (and hence the c_j values) no longer change. Note that each iteration needs $N \times k$ comparisons, which determines the time complexity of one iteration. The number of iterations required for convergence varies and may depend on N, but as a first cut, this algorithm can be considered linear in the dataset size. One issue to resolve is how to quantify closest in the assignment step. The default measure of closeness is the Euclidean distance, in which case one can readily show that the non-negative cost function,

$$\sum_{i=1}^{N} (argmin_j \left\| xi - cj \right\|)$$

will decrease whenever there is a change in the assignment or the relocation steps, and hence convergence is guaranteed in a finite number of iterations. The greedy-descent nature of k-means on a non-convex cost also implies that the convergence is only to a local optimum, and indeed the algorithm is typically quite sensitive to the initial centroid locations.

2.6.2 K-nearest Neighbour Classification

k-nearest neighbor (kNN) classification finds a group of k objects in the training set that are closest to the test object, and bases the assignment of a label on the predominance of a particular class in this neighborhood. There are three key elements of this approach: a set of labeled objects, e.g., a set of stored records, a distance or similarity metric to compute distance between objects, and the value of k, the number of nearest neighbors. To classify an unlabeled object, the distance of this object to the labeled objects is computed, its k-nearest neighbors are identified, and the class labels of these nearest neighbors are then used to determine the class label of the object.

Here is provided a high-level summary of the nearest-neighbor classification method. Given a training set D and a test object $x = (x', y')$, the algorithm computes the distance (or similarity) between z and all the training objects $(x, y) \in D$ to determine its nearest-neighbor list, D_z. (x is the data of a training object, while y is its class. Likewise, x' is the data of the test object and y' is its class.)

Once the nearest-neighbor list is obtained, the test object is classified based on the majority class of its nearest neighbors:

Majority Voting: $y' = argmax_v \sum_{(x_i,y_i)\in D_z} I\,(v = y_i)$

where v is a class label, y_i is the class label for the ith nearest neighbors, and $I()$ is an indicator function that returns the value 1 if its argument is true and 0 otherwise.

```
Input: $D$, the set of $k$ training objects and test object $z = (x', y')$
Process:
Compute $d\,(x', x)$, the distance between $z$ and every object, $(x, y)$.
Select $D_z \subseteq D$, the set of $k$ closest training objects to $z$.
Output: $y' = argmax_v \sum_{(x_i,y_i)\in D_z} I\,(v = y_i)$.
```

The k-nearest neighbor classification algorithm [9]

There are several key issues that affect the performance of kNN. One is the choice of k. If k is too small, then the result can be sensitive to noise points. On the other hand, if k is too large, then the neighborhood may include too many points from other classes. Another issue is the approach to combining the class labels. The simplest method is to take a majority vote, but this can be a problem if the nearest neighbors vary widely in their distance and the closer neighbors more reliably indicate the class of the object. A more sophisticated approach, which is usually much less sensitive to the choice of k, weights each objects vote by its distance, where the weight factor is often taken to be the reciprocal of the squared distance: $w_i = 1/d(x\acute{}, x_i)^2$. This amounts to replacing the last step of the kNN algorithm with the following:

$$Distance - WeightedVoting : y\acute{} = argmax_v \sum_{(x_i,y_i)\in D_z} w_i \times I\,(v = y_i)$$

KNN classification is an easy to understand and easy to implement classification technique. Despite its simplicity, it can perform well in many situations. In particular, a well known result by Cover and Hart [9] shows that the the error of the nearest neighbor rule is bounded above by twice the Bayes error under certain reasonable assumptions. Also, the error of the general kNN method asymptotically approaches that of the Bayes error and can be used to approximate it.

2.6.3 Bayes Classification

A naive Bayes (NB) classifier is a simple probabilistic classifier based on: (a) Bayes theorem, (b) strong (naive) independence assumptions, and (c) independent feature models. It is also an important mining classifier for data mining and app;ied in many real world classification problems because of its high classification performance. A NB classifier can easily handle missing attribute values by simply omitting the corresponding probabilities for those attributes when calculating the likelihood of membership for each class. The NB classifier also requires the class conditional independence, i.e. the effect of an attribute on a given class is independent of these of other attributes. The NB classifier has several advantages such as:

1. Easy to use.
2. Only one scan of training data required.
3. Handling missing attribute values.
4. Continuius data.
5. High classification performance.

Given a training dataset, $D = \{X_1, X_2, \ldots, X_n\}$, each data record is represented as, $X_i = \{x_1, x_2, \ldots, x_n\}$. D contains the following attributes $\{A_1, A_2, \ldots, A_n\}$ and each attribute A_i contains the following attribute values $\{A_i1, A_i2, \ldots, A_in\}$. The attribute values can be discrete or continuous. D also contains a set of classes $C = \{C_1, C_2, C_m\}$. Each training instance, $X \in D$, has a particular class lebel C_i. For a test instance , X, the classifier will predict that X belongs to the class with the highest posterior probability, conditioned on X.

That is, the NB classifier predicts that the instance X belongs to the class C_i, if and only if $P(C_i|X) > p(C_j|X)$ for $1 \leq j \leq m, j \neq i$. The class C_i for which $P(C_i|X)$ is maximized is called the Maximum Posteriori Hypothesis.

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

In this theorem, as $P(X)$ is a constant for all classes, only $P(X|C_i)P(C_i)$ needs to be maximized. If the class prior probabilities are not konwn, then it is commonly assumed that the classes are equally likely, that is $P(C_1) = P(C_2) = \cdots = P(C_m)$, and therefore maximize $P(C_i)$. Otherwise, maximize $P(X|C_i)P(C_i)$. The class prior probabilities are calculated by $P(C_i) = |C_i, D| \,/\, |D|$, where $|C_i, D$ is the number of training instances belonging to the class C_i in D. To compute $P(X|C_i)$ in a dataset with many attributes is extreamly computationally expensive. Thus, the naive assumption of class conditional independence is made in order to reduce computation in evaluating $P(X|C_i)$. The attributes are conditionally independent of another, given the class label of the instance. Thus eq. 2.2 and eq. 2.3 are used to produce $P(X|C_i)$.

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i) \ldots\ldots\ldots\ldots (2.2)$$

$$P(X|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_n|C_i) \cdots (2.3)$$

In eq. 2.2, x_k refers to the value of attribute A_k for instance X. Therefore, these probabilities $P(x_1|C_i), P(x_2|C_i), \cdots, P(x_n|C_i)$ can be easily estimated from the training instances [10].

CHAPTER 3
LITERATURE REVIEW

3.1 Introduction

Educational data mining (EDM) is an emerging interdisciplinary research area that deals with the development of methods to explore data originating in an educational context. EDM uses computational approaches to analyze educational data in order to study educational questions. This paper surveys the most relevant studies carried out in this field to date. It goes on to list the most common tasks in the educational environment that have been resolved through data mining techniques and some of the most promising future lines of research are discussed.

3.2 Student performance and data mining

Data mining represents a computational data process with the goal of extracting implicit and interesting samples [11], trends and information from the data. So it can greatly help every participant in the educational process in order to improve the understanding of the teaching process and it centers on discovering, detecting and explaining educational phenomenons. In recent years there has been an increased interest in using data mining for educational purposes. One of the educational problems that are solved with data mining is the prediction of students' academic performances, whose goal is to predict an unknown variable (outcome, grades or scores) that describes students. The estimation of students' performances includes monitoring and guiding students through the teaching process and assessment. The hidden patterns, associations, and anomalies that are discovered by data mining techniques from educational data can improve decision making processes in higher educational systems. This improvement can bring advantages such as maximizing educational system efficiency, decreasing student's drop-out rate, and increasing student's promotion rate, increasing student's retention rate in, increasing student's transition rate, increasing educational improvement ratio, increasing student's success, increasing student's learning outcome, and reducing the cost of system processes. In this current era we are using the KDD and the data mining tools for extracting the knowledge this knowledge can be used for improving the quality of education .

3.3 Related works

EDM has both applied research objectives, such as improving the learning process and guiding students learning; as well as pure research objectives, such as achieving a deeper understanding of educational phenomena.

“Minaei-Bidgolim, et al. (2003) was among the first authors who classified students by using genetic algorithms to predict their final grade. Using the regression methods, Kotsiantis and Pintelas predicted a students marks (pass and fail classes). Superby, Vandamme and Meskens predicted a students academic success (classified into low,medium, and high risk classes) using different data mining methods (decision trees and neural network). Al-Radaideh, Al-Shawakfa and Al-Najjar applied a decision tree model to predict the final grade of students who studied the C++ course in Yarmouk University, Jordan. Romero et al. (2008) compared different methods of data mining in order to predict final assessment based on the data obtained from the system of e- learning. Zeki-Suac, Frajman-Jaki and Drvenkar created a model for predicting students’ performance using neural networks and classification trees decision-making, and with the analysis of factors which influence students’ success. Kumar and Vijayalakshmi [12] using the decision tree predicted the result of the final exam to help professors identify students who needed help, in order to improve their performance and pass the exam [13] ”.

CHAPTER 4
PROPOSED STUDY OF STUDENT PERFORMANCES

4.1 Research methodology

To understand the competency of students, the triggering factors that motivate or demotivate a student from performing good, initially a survey was conduct among more than 500 students in MIST. Before conducting survey a pre-survey questioning was conducted. Then a question set of more than 50 questions was prepared. After repeated modification a two column questionnaire, containing 36 questions, with multiple choice options was designed to conduct the survey. The sample question was checked and rechecked, to determine if it can lead towards desired outcomes. Data collected from survey was entered into excel sheet. Then data was normalized, validated, revalidated. To analyze data and take decision , different pie-chart, bar-chart and line-curve was created using the survey data. This curves lead us to take decision about student performance and work further. To proceed we decided to take data from student database.Then we collected data from student database of MIST. Using Weka tool the next part of data mining was conduct. Data taken from MIST was converted into ARFF format.

4.2 Data preparation and preprocessing

The data was prepared into two steps. Initially survey data was prepared and processed. Then Data collected from database was prepared and processed.

Data collected from survey was processed using different kind of filtering. Ambiguous data, Empty field were not considered during this process.

Data was normalized using different code. Normalized data was validated, revalidated. The data was prepared.The prepared data was processed using standard deviation, variance, minimum value, maximum value and mod value. Data collected from database were mostly functional.

4.3 Identification of parameters for data mining

To conduct our data mining, we collected data from MIST student database. The database had so many attribute values. The values used as parameters for data mining are :

Level and Term wise GPA : The Grade point average of every student in their every level and term. This is a numeric value while converted into ARFF format.

CGPA : Cumulative grade point average. This is also a numeric value.

Gender: If a student is a male student or a female student. This is a string value with a set of values.

4.4 Using data mining tool Weka

Weka (Waikato Environment for Knowledge Analysis) is a machine learning software written in Java, developed at the University of Waikato, New Zealand.

The Weka contains a collection of visualization tools and algorithms for data analysis and predictive modeling, together with graphical user interfaces for easy access to this functionality [14].

Advantages of Weka include:

- free availability under the GNU General Public License.
- portability, since it is fully implemented in the Java programming language and thus runs on almost any modern computing platform.
- a comprehensive collection of data preprocessing and modeling techniques.
- ease of use due to its graphical user interfaces.

Weka supports several standard data mining tasks, specifically,

data preprocessing, clustering, classification, regression, visualization, and feature selection.

All of Weka's techniques are predicated on the assumption that the data is available as a single flat file or relation, where each data point is described by a fixed number of attributes (normally, numeric or nominal attributes, but some other attribute types are also supported).

Weka provides access to SQL databases using Java Database Connectivity and can process the result returned by a database query. It is not capable of multi-relational data mining.

4.5 Weka file format

An ARFF (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes. ARFF files were developed by the Machine Learning Project at the Department of Computer Science of The University of Waikato for use with the Weka machine learning software. ARFF files have two sections: Header information and Data information [15].

The Header of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types.

The @RELATION, @ATTRIBUTE and @DATA declarations are case insensitive.Lines that begin with a percentage sign are comments.

The ARFF Header Section

The ARFF Header section of the file contains the relation declaration and attribute declarations.

The @relation Declaration

The relation name is defined as the first line in the ARFF file. The format is:

@relation $< relation - name >$

where $< relation - name >$ is a string. The string must be quoted if the name includes spaces [16].

The @attribute Declarations

The format for the @attribute statement is:

@attribute $< attribute - name >$ $< datatype >$

where the $< attribute - name >$ must start with an alphabetic character. If spaces are to be included in the name then the entire name must be quoted. The $< datatype >$ can be any of the four types currently (version 3.2.1) supported by Weka:

- numeric
- $< nominal - specification >$
- string
- $< date - format >$

ARFF Data Section

The ARFF Data section of the file contains the data declaration line and the actual instance lines.

The @data Declaration

The @data declaration is a single line denoting the start of the data segment in the file. The format is: @data

The instance data

Each instance is represented on a single line, with carriage returns denoting the end of the instance. Attribute values for each instance are separated by commas. They must appear in the order that they were declared in the header section (i.e. the data corresponding to the nth @attribute declaration is always the nth field of the attribute).

Missing values are represented by a single question mark, as in:

@data $4.4,?,1.5,?,Iris - setosa$

Values of string and nominal attributes are case sensitive.

4.6 Convert data to ARFF format

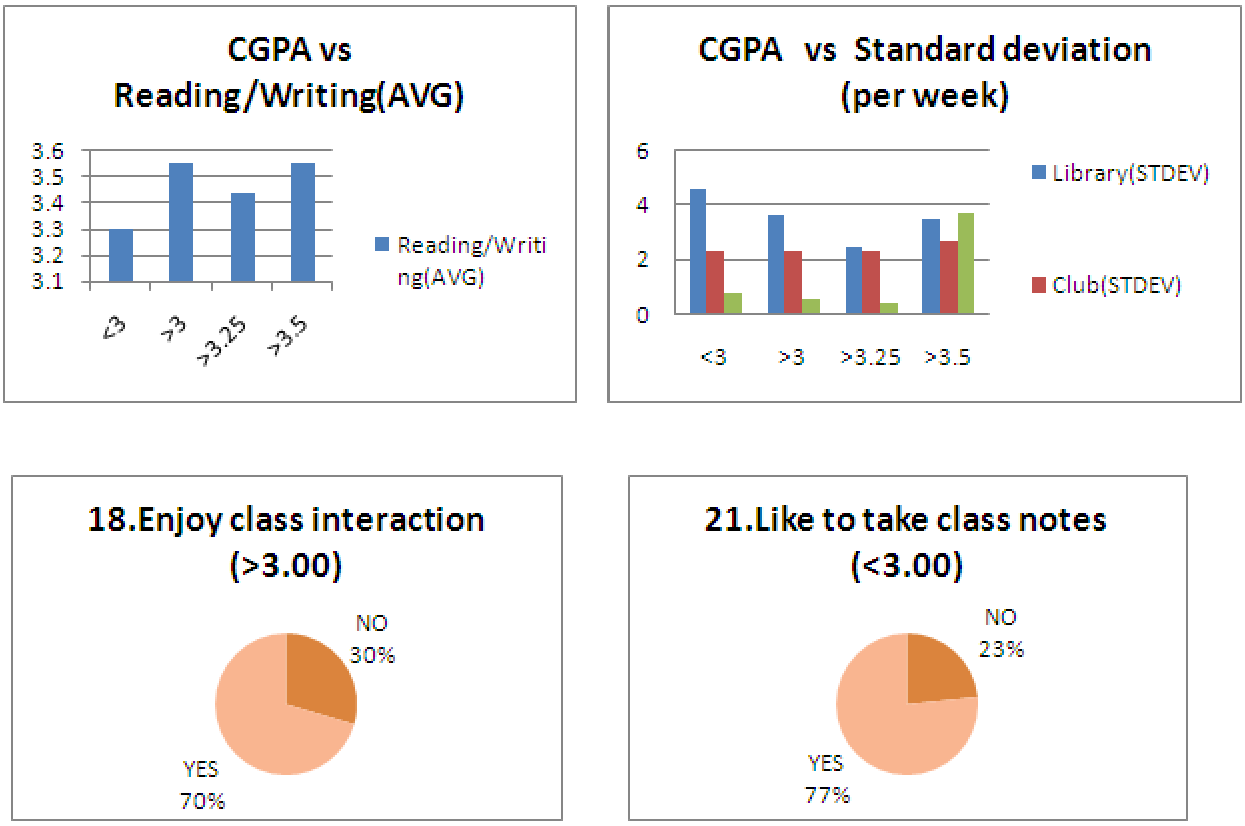
The data collected from database was in excel sheet. To convert data to ARFF format we had to convert data in CSV(comma separated value) files. This CSV files were text files. Later they were converted to ARFF format. Thus the conversion was done in two steps:

Step 1:Preparing CSV file : The excel data were converted to CSV file.

Step 2:Define the attribute types: Select the attribute and define their types [15].

4.7 After conduct survey processing data

After conducting survey we normalize and validate data. After that we make the graph according to our data we get from our question paper given to the students.



CGPA vs Reading/Writing(AVG)

CGPA vs Standard deviation (per week)

18.Enjoy class interaction (>3.00)

21.Like to take class notes (<3.00)

Table 4.1: Hours spent per day on social media

CHAPTER 5
EXPERIMENTAL RESULTS

5.1 Selected WEKA algorithm

In a decision tree algorithm, one of the main complication is the optimal size of final tree.All theoreticians and specialist are still now searching for techniques to make this algorithm more efficient,cost−effective and accurate. A tree if it is too big then there is a risk to overfitt the training data and poor generalization to new sample. If a tree is too small then it might not capture important structural information about the sample space.Sometimes the addition of a single extra node will dramatically decrease error. This problem is known as horizon effect. So a common strategy is to use tree pruning to remove nodes that do not provide additional information. The goal of pruning is reduced complexity of the final classifier and better predictive accuracy by the reduction of overfitting and removal of sections of a classifier that may be based on noisy or erroneous data. There are two common approaches to tree prunning.

1. Pre-pruning
2. Post-pruning

Pre-pruning a tree is pruned by halting its construction early(e.g. by deciding not to furthur split or partition the subset of training instances at given node). Upon halting, the node becomes a leaf. The leaf may hold the most frequent class among the subset instances or the probability distribution of those instances. In choosing an appropriate thresold, high thresolds could result in oversimplified trees, whereas low thresolds could result in very little simplification [17].

Post-pruning is commonly used tree pruning approach, which removes subtrees from a full grown tree. A subtree at a given node is pruned by removing its branches and replacing it with a leaf. The leaf is lebeled with the most frequent class among the subtree being replaced [18].

Post pruning requires more computation than pre-pruning, yet generally leads to a more reliable tree.

In our decision tree algorithm we use J48 algorithm to classily our instances.

About J48 pruned tree:

J48s default is like C4.5, post pruning using subtree raising with a pruning confidence of 0.25. Subtree replacement can be used as an alternative post pruning method by turning off subtree raising with -S, and pruning confidence can be adjusted with -C. Reduced error pruning (-R) is a post-pruning method that uses a hold out set for error estimates. It also turns off subtree raising (and of course subtree replacement), and will throw an exception if used in conjunction with -C, as it doesnt use confidence for pruning. The proportion of data held back for pruning is controlled by setting the number of folds (an unfortunately overloaded term if youre also doing k-fold cross-validation). The number of folds is set with -N, with one set used for pruning and the rest for training. Setting -N will throw an exception unless -R is set. All pruning can be turned off with -U, which will either throw an exception if any of the above pruning switches are set.

J48 algorithm is an extension of ID3 algorithm and possibly creates a small tree. It uses a divide and conquers approach to growing decision trees that was leaded by Hunt and his co-workers [19].

A. Construction

Some basic steps are given below to construct tree:-

1. check whether all cases belongs to same class, then the tree is a leaf and is labeled with that class.
2. For each attribute, calculate the information and information gain.
3. Find the best splitting attribute (depending upon current selection criterion) [20] .

B. Counting information gain

Entropy is used in this process. Entropy is a measure of disorder of data. Entropy is measured in bits, nats or bans. This is also called measurement of uncertainty in any random variable. Just suppose that there is a fair coin, if single toss is performed on that coin than its entropy will be one bit. A series of two fair coins tosses will have entropy of two bits. Now if coin is not fair than there is uncertainty and this provides lower entropy rate.Entropy for any P can be calculated as:-

$$Entropy(p) = -\sum_{j=1}^{n} \frac{|p_j|}{|p|} log \frac{|p_j|}{|p|}$$

The conditional entropy:

$$Entropy(j|p) = \frac{|p_j|}{|p|} log \frac{|p_j|}{|p|}$$

The conditional entropy is:-If base is 2 for logarithm than entropy measurement unit will be in bits, if base is 10 than unit is dits. Information Gain isused for measuring association between inputs and outputs. It is a state to state change in information entropy. Finally information gain can be calculated as:-

Gain(p,j)=Entropy(p-Entropy(j|p))

To get a small and efficient tree, splitting should be based on highest gain. Just suppose that there are 9 male (m) and 5female (f) in a class instance. This instance in divided further into two different groups or instances on the bases of their calculated entropy and information gain. So 3m and 4f as left instance and 6m and 1f as right instance. Entropy and information gain can be measured just by putting values in formula as given below:-

Entropy$_b ef =$-5/14*log(5/14)-9/14*log(9/14)

Entropy$_l eft =$-3/7*log(3/7)-4/7*log(4/7)

Entropy$_r ight =$-6/7*log(6/7)-1/7*log(1/7)

Entropy$_a ft = 7/14 * Entropy_l eft$+7/14*$Entropy_r ight$

Information$_G ain = Entropy_b ef$-$Entropy_a ft$

k**-fold cross validation** k-fold cross validation is a common technique for estimating the performance of a classifier. Given a set of m traning examples, a single run of k-fold cross validation proceeds as follows:

1. Arrange the training examples in a random order.
2. Divide the training examples into k folds. (k chunks of approximately m/k examples each.)
3. For $i = 1, \ldots, k$:
   - Train the classifier using all the examples that do not belong to Fold i.
   - Test the classifier on all the examples in Fold i.
   - Compute n_i, the number of examples in Fold i that were wrongly classified.
4. Return the following estimate to the classifier error:

$$E = \frac{\sum_{i=1}^{k} n_i}{m}$$

To obtain an accurate estimate to the accuracy of a classifier, k-fold cross validation is run several times, each with a different random arrangement in Step 1. Let $E_1, \ldots, E_t$ be the accuracy estimates obtained in t runs [21]. Define:

$$e = \frac{\sum_{j-1}^{t} E_j}{t}, V = \frac{\sum_{j-1}^{t}(E_j - e^2)}{t-1}, \sigma = \sqrt{V}$$

The esimate for the algorithm performance is an error of e with standard-deviation of σ.

5.2 Results from WEKA analysis

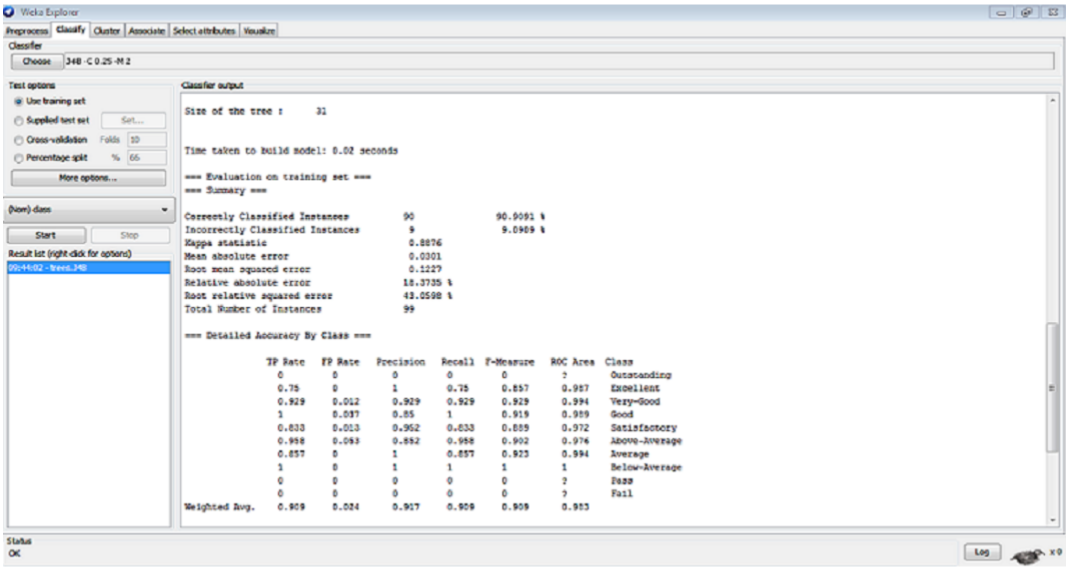
How to predict the final result of BSc life of a student through his running result/GPA is the main concern. By creating a classification tree (decision tree), the data can be mined to determine the academic performance of a student based on his current or running GPA. In the 4 years academic career of MIST each student has to go through 4 levels and each level has 2 terms. The $'Level'$s and $'Term'$s are selected as the possible nodes of the tree in WEKA. There are total 8 class attributes and their CGPA ranges are:

- Outstanding : marks obtained 80% and above (Grade point 4.00)
- Excellent : marks obtained from 75% to less than 80% (Grade point 3.75)
- Very Good : marks obtained from 70% to less than 75% (Grade point 3.50)
- Good : marks obtained from 65% to less than 70% (Grade point 3.25)
- Satisfactory: marks obtained from 60% to less than 65% (Grade point 3.00)
- Above Average: marks obtained from 55% to less than 60% (Grade point 2.75)
- Average : marks obtained from 50% to less than 55% (Grade point 2.50)
- Below Average : marks obtained from 45% to less than 50% (Grade point 2.25)

Load the data file- MIST.arff into WEKA. The file contains the result information of students. We need to divide up our records, so some data instances are used to create the model, and some are used to test the model to ensure that we didn't overfit it.In Weka we use **"Use Training set"** to load our data set and create our model.

Output from WEKA's Classification model

This classification is done by j48.This classification model is done by using the training data set. We choose 60% data as training from main database and the rest database is used as test data set.



Classifier: J48 -C 0.25 -M 2

Test options:
● Use training set
○ Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) class

09:44:02 - trees.J48

Status: OK

```
Size of the tree :      31


Time taken to build model: 0.02 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances          90               90.9091 %
Incorrectly Classified Instances         9                9.0909 %
Kappa statistic                          0.8876
Mean absolute error                      0.0301
Root mean squared error                  0.1227
Relative absolute error                 18.3735 %
Root relative squared error             43.0598 %
Total Number of Instances               99

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0         0         0          0         0          ?        Outstanding
                 0.75      0         1          0.75      0.857      0.987    Excellent
                 0.929     0.012     0.929      0.929     0.929      0.994    Very-Good
                 1         0.037     0.85       1         0.919      0.989    Good
                 0.833     0.013     0.952      0.833     0.889      0.972    Satisfactory
                 0.958     0.053     0.852      0.958     0.902      0.976    Above-Average
                 0.857     0         1          0.857     0.923      0.994    Average
                 1         0         1          1         1          1        Below-Average
                 0         0         0          0         0          ?        Pass
                 0         0         0          0         0          ?        Fail
Weighted Avg.    0.909     0.024     0.917      0.909     0.909      0.983
```

Figure 5.1: Data classification in WEKA

In the first output model there is the classified J48 pruned tree.

There are 84 instances and 9 attributes.In this tree

Number of Leaves : 12

Size of the tree : 23

Here,

Correctly classified instances: 91.6667%

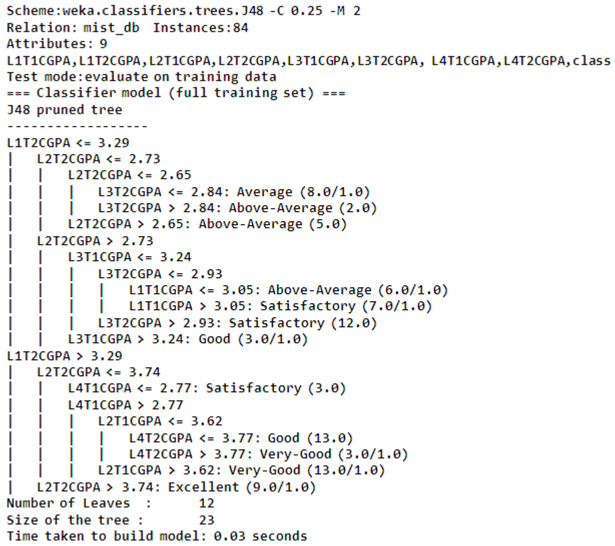
Incorrectly classified instances: 8.3333%

Based on our accuracy rate, we can say that this is a pretty good model to predict.

5.3 Performance analysis of data set

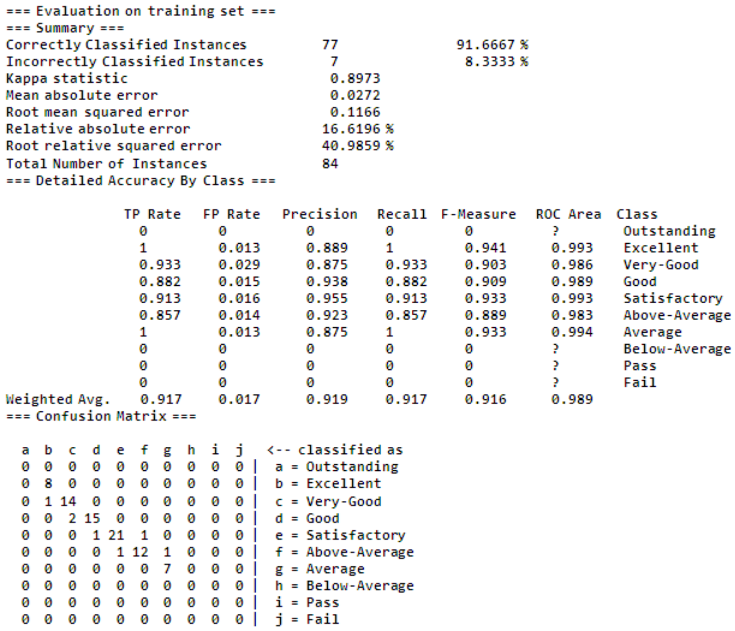
Using the decision tree , in its J48 weka implementation we want to predict the class attribute based on attributes level and term wise GPA. The root is chosen from that attribute where information gain is highest. The root which is divided by CGPA 3.29, has two child, as there are equal number of GPA above and below 3.29. The rest of the nodes are selected by the same procedure. The class attributes are the leaves which is our ultimate result.To visulaize the tree , by right-clicking the model we just created.Reading the graphic we can notice that if level-1 term-2 CGPA is greater than 3.29 and level-2 term-2 CGPA is greater than 3.74 then there are 9 out of 84 objects that fall in Excellent.

Confusion Matrix is telling the following:



```
Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: mist_db  Instances:84
Attributes: 9
L1T1CGPA,L1T2CGPA,L2T1CGPA,L2T2CGPA,L3T1CGPA,L3T2CGPA, L4T1CGPA,L4T2CGPA,class
Test mode:evaluate on training data
=== Classifier model (full training set) ===
J48 pruned tree
------------------
L1T2CGPA <= 3.29
|   L2T2CGPA <= 2.73
|   |   L2T2CGPA <= 2.65
|   |   |   L3T2CGPA <= 2.84: Average (8.0/1.0)
|   |   |   L3T2CGPA > 2.84: Above-Average (2.0)
|   |   L2T2CGPA > 2.65: Above-Average (5.0)
|   L2T2CGPA > 2.73
|   |   L3T1CGPA <= 3.24
|   |   |   L3T2CGPA <= 2.93
|   |   |   |   L1T1CGPA <= 3.05: Above-Average (6.0/1.0)
|   |   |   |   L1T1CGPA > 3.05: Satisfactory (7.0/1.0)
|   |   |   L3T2CGPA > 2.93: Satisfactory (12.0)
|   |   L3T1CGPA > 3.24: Good (3.0/1.0)
L1T2CGPA > 3.29
|   L2T2CGPA <= 3.74
|   |   L4T1CGPA <= 2.77: Satisfactory (3.0)
|   |   L4T1CGPA > 2.77
|   |   |   L2T1CGPA <= 3.62
|   |   |   |   L4T2CGPA <= 3.77: Good (13.0)
|   |   |   |   L4T2CGPA > 3.77: Very-Good (3.0/1.0)
|   |   |   L2T1CGPA > 3.62: Very-Good (13.0/1.0)
|   L2T2CGPA > 3.74: Excellent (9.0/1.0)
Number of Leaves  :      12
Size of the tree :       23
Time taken to build model: 0.03 seconds
```

Figure 5.2: Output from weka's classification model (A)



```
=== Evaluation on training set ===
=== Summary ===
Correctly Classified Instances          77               91.6667 %
Incorrectly Classified Instances         7                8.3333 %
Kappa statistic                          0.8973
Mean absolute error                      0.0272
Root mean squared error                  0.1166
Relative absolute error                 16.6196 %
Root relative squared error             40.9859 %
Total Number of Instances               84
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0         0         0         0         0         ?        Outstanding
                 1         0.013     0.889     1         0.941     0.993    Excellent
                 0.933     0.029     0.875     0.933     0.903     0.986    Very-Good
                 0.882     0.015     0.938     0.882     0.909     0.989    Good
                 0.913     0.016     0.955     0.913     0.933     0.993    Satisfactory
                 0.857     0.014     0.923     0.857     0.889     0.983    Above-Average
                 1         0.013     0.875     1         0.933     0.994    Average
                 0         0         0         0         0         ?        Below-Average
                 0         0         0         0         0         ?        Pass
                 0         0         0         0         0         ?        Fail
Weighted Avg.    0.917     0.017     0.919     0.917     0.916     0.989
=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j   <-- classified as
  0  0  0  0  0  0  0  0  0  0 |  a = Outstanding
  0  8  0  0  0  0  0  0  0  0 |  b = Excellent
  0  1 14  0  0  0  0  0  0  0 |  c = Very-Good
  0  0  2 15  0  0  0  0  0  0 |  d = Good
  0  0  0  1 21  1  0  0  0  0 |  e = Satisfactory
  0  0  0  0  1 12  1  0  0  0 |  f = Above-Average
  0  0  0  0  0  0  7  0  0  0 |  g = Average
  0  0  0  0  0  0  0  0  0  0 |  h = Below-Average
  0  0  0  0  0  0  0  0  0  0 |  i = Pass
  0  0  0  0  0  0  0  0  0  0 |  j = Fail
```

Figure 5.3: Output from weka's classification model (B)

The decision tree has classified 8 Excellent objects as Excellent and 1 as Very-good leading in 1 misclassification,. The decision tree has classified 14 Very-good objects as Very-good and 2 as Good, leading in 2 Misclassifications. The decision tree has classified 15 Good objects as Good and 1 as Satisfactory, leading in 1 misclassification. Rest are classified as same as written above.

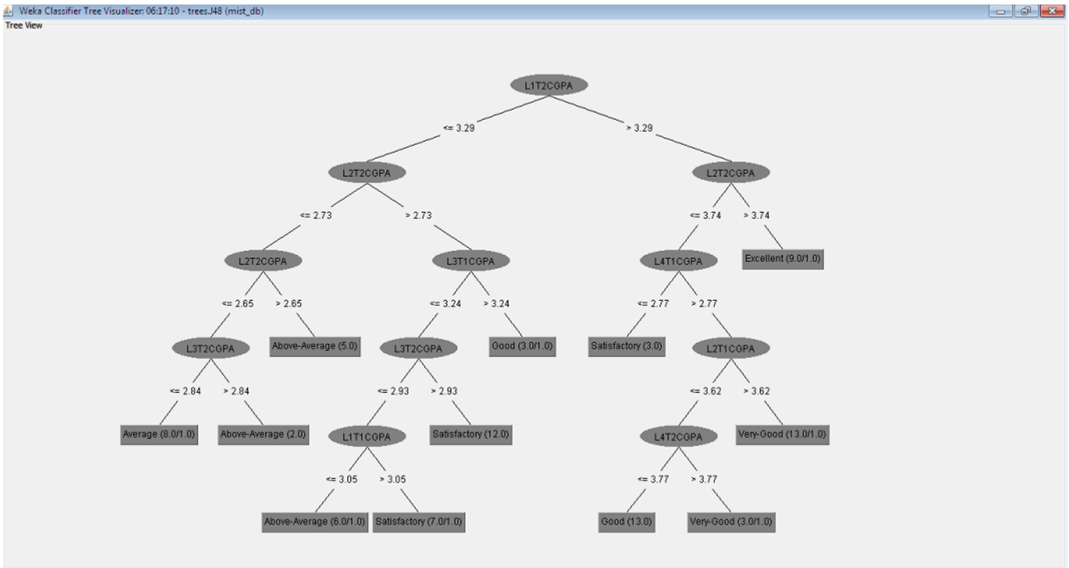


Figure 5.4: Classification tree visualization

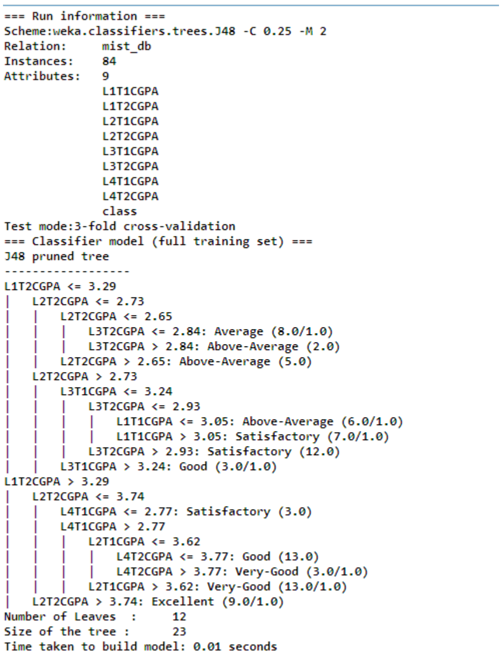
5.4 Validation of experimental results

There is one final step to validating our classification tree, which is to run our test set through the model and ensure the accuracy of the model when evaluting the test set is not too different from the training set. For validating data we use 2 fold cross validation.

Comparing the ”Correctly Classified Instances” from the test set (61.9 percent) with the ” Correctly Classified Instances ” from the training set (91.6 percent). We see that the accuracy of the model is pretty good.

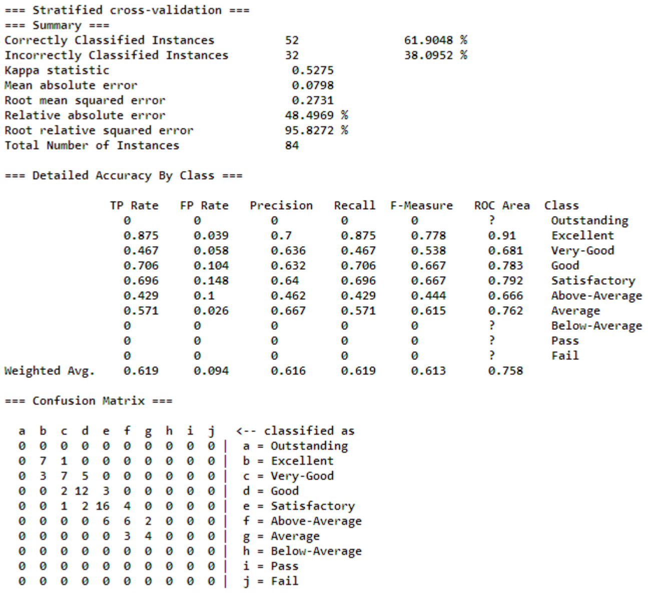
5.5 Constraints and assumptions

Our accuracy model is not so close, Because there is a lots of lackings and missing datas in the database of MIST.Informations are there were not so sufficient. Besides the model we created , this is not generic. Because there is no demographical data of a student. There are a lots of reasons which can effect the academic result of a student for example , his/her physical condition,mental condition,financilal condition,addiction in drug etc.If those informations were available the model could have been better.



```
=== Run information ===
Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     mist_db
Instances:    84
Attributes:   9
              L1T1CGPA
              L1T2CGPA
              L2T1CGPA
              L2T2CGPA
              L3T1CGPA
              L3T2CGPA
              L4T1CGPA
              L4T2CGPA
              class
Test mode:3-fold cross-validation
=== Classifier model (full training set) ===
J48 pruned tree
------------------
L1T2CGPA <= 3.29
|   L2T2CGPA <= 2.73
|   |   L2T2CGPA <= 2.65
|   |   |   L3T2CGPA <= 2.84: Average (8.0/1.0)
|   |   |   L3T2CGPA > 2.84: Above-Average (2.0)
|   |   L2T2CGPA > 2.65: Above-Average (5.0)
|   L2T2CGPA > 2.73
|   |   L3T1CGPA <= 3.24
|   |   |   L3T2CGPA <= 2.93
|   |   |   |   L1T1CGPA <= 3.05: Above-Average (6.0/1.0)
|   |   |   |   L1T1CGPA > 3.05: Satisfactory (7.0/1.0)
|   |   |   L3T2CGPA > 2.93: Satisfactory (12.0)
|   |   L3T1CGPA > 3.24: Good (3.0/1.0)
L1T2CGPA > 3.29
|   L2T2CGPA <= 3.74
|   |   L4T1CGPA <= 2.77: Satisfactory (3.0)
|   |   L4T1CGPA > 2.77
|   |   |   L2T1CGPA <= 3.62
|   |   |   |   L4T2CGPA <= 3.77: Good (13.0)
|   |   |   |   L4T2CGPA > 3.77: Very-Good (3.0/1.0)
|   |   |   L2T1CGPA > 3.62: Very-Good (13.0/1.0)
|   L2T2CGPA > 3.74: Excellent (9.0/1.0)
Number of Leaves  :     12
Size of the tree :      23
Time taken to build model: 0.01 seconds
```

Figure 5.5: Output form WEKA'S classification model of Test Data (A)



```
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances          52               61.9048 %
Incorrectly Classified Instances        32               38.0952 %
Kappa statistic                          0.5275
Mean absolute error                      0.0798
Root mean squared error                  0.2731
Relative absolute error                 48.4969 %
Root relative squared error             95.8272 %
Total Number of Instances               84

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0         0          0         0         0          ?        Outstanding
                 0.875     0.039      0.7       0.875     0.778      0.91     Excellent
                 0.467     0.058      0.636     0.467     0.538      0.681    Very-Good
                 0.706     0.104      0.632     0.706     0.667      0.783    Good
                 0.696     0.148      0.64      0.696     0.667      0.792    Satisfactory
                 0.429     0.1        0.462     0.429     0.444      0.666    Above-Average
                 0.571     0.026      0.667     0.571     0.615      0.762    Average
                 0         0          0         0         0          ?        Below-Average
                 0         0          0         0         0          ?        Pass
                 0         0          0         0         0          ?        Fail
Weighted Avg.    0.619     0.094      0.616     0.619     0.613      0.758

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j   <-- classified as
  0  0  0  0  0  0  0  0  0  0 |  a = Outstanding
  0  7  1  0  0  0  0  0  0  0 |  b = Excellent
  0  3  7  5  0  0  0  0  0  0 |  c = Very-Good
  0  0  2 12  3  0  0  0  0  0 |  d = Good
  0  0  1  2 16  4  0  0  0  0 |  e = Satisfactory
  0  0  0  0  6  6  2  0  0  0 |  f = Above-Average
  0  0  0  0  0  3  4  0  0  0 |  g = Average
  0  0  0  0  0  0  0  0  0  0 |  h = Below-Average
  0  0  0  0  0  0  0  0  0  0 |  i = Pass
  0  0  0  0  0  0  0  0  0  0 |  j = Fail
```

Figure 5.6: Output form WEKA'S classification model of Test Data (B)

CHAPTER 6
CONCLUSION

6.1 Conclusion and Discussion

This paper is a review of performance analysis with respect to EDM and surveys the most relevant work in this area to date. Each study has been classified, not only by the type of data and DM techniques used, but also and more importantly, by the type of educational task that they resolve. In this paper, J48 data mining algorithm was applied on the preoperative assessment data to predict success in a course and the performance of the learning methods were evaluated based on their predictive accuracy, ease of learning and user friendly characteristics. Here some other algorithms, e.g. K-means Clustering, KNN-Classification, Naive Bayes algorithm were discussed and we applied them on our data, tried to find out co-relation. We have learnt that some algorithms improve their classification performance when we apply such preprocessing tasks as discretization and rebalancing data, but others do not. We found highest 61% accuracy when we applied J48 algorithm and so we choose it for our data. Hopefully, this tree can be modified further which will produce more accurate result. We have also known that a good classifier model has to be both accurate and comprehensible for the students and instructors of a University.

6.2 Future Expansion

Our primary aim was to clarify the relation between knowledge discovery and data mining. We provided an overview of the KDD process and basic data-mining methods. Given the broad spectrum of data-mining methods and algorithms, our overview is inevitably limited in scope: There are many data-mining techniques, particularly specialized methods for particular types of data and domain. Although various algorithms and applications might appear quite different on the surface, it is not uncommon to find that they share many common components. Understanding data mining and model induction at this component level clarifies the task of any DM algorithm and makes it easier for the user to understand its overall contribution and applicability to the KDD process. This article represents a step toward a common framework that we hope will ultimately provide a unifying vision of the common overall goals and methods used in KDD. We hope this will eventually lead to a better understanding of the variety of approaches in this multidisciplinary field and we will successful to develop a model which will predict the final result of a student by his current performance analysis and guide him for betterment of his academic performance.

REFERENCES

[1] A. Ayinde, A. Adetunji, M. Bello, and O. Odeniyi, “Performance evaluation of naive bayes and decision stump algorithms in mining students’ educational data.,” *International Journal of Computer Science Issues (IJCSI)*, vol. 10, no. 4, 2013.

[2] T. Joachims, D. Freitag, and T. Mitchell, “Webwatcher: A tour guide for the world wide web,” in *IJCAI (1)*, pp. 770–777, Citeseer, 1997.

[3] C. Romero and S. Ventura, “Educational data mining: a review of the state of the art,” *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 40, no. 6, pp. 601–618, 2010.

[4] “Data mining.” Last accessed on December 21, 2014, at 02:08:00PM. [Online]. Available: http://en.wikipedia.org/wiki/Data$_mining.$

[5] R. R. Mendes, F. B. de Voznika, A. A. Freitas, and J. C. Nievola, “Discovering fuzzy classification rules with genetic programming and co-evolution,” in *Principles of Data Mining and Knowledge Discovery*, pp. 314–325, Springer, 2001.

[6] B. Zhang, *Discovering Legible And Readable Chinese Typefaces For Reading Digital Documents*. PhD thesis, Concordia University, 2011.

[7] R. Krishnapuram and C.-P. Freg, “Fitting an unknown number of lines and planes to image data through compatible cluster merging,” *Pattern recognition*, vol. 25, no. 4, pp. 385–400, 1992.

[8] R. M. Gray, T. Linder, and J. Li, “A lagrangian formulation of zador’s entropy-constrained quantization theorem,” *Information Theory, IEEE Transactions on*, vol. 48, no. 3, pp. 695–707, 2002.

[9] D. Vavilov, K. Kostyushkin, and I. Platonov, “Users behaivour analysis application for open tv platforms,” 2012.

[10] Y. Liu, P. Z. Zhang, and J. P. Gong, “Research on the classification based on naive bayes,” *Applied Mechanics and Materials*, vol. 543, pp. 1643–1646, 2014.

[11] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[12] S. A. Kumar and M. Vijayalakshmi, “Efficiency of decision trees in predicting student’s academic performance,” in *First International Conference on Computer Science, Engineering and Applications, CS and IT*, vol. 2, pp. 335–343, 2011.

[13] D. Shakir Khan, A. Sharma, A. S. Zamani, and A. Akhtar, “Data mining for security purpose & its solitude suggestions,”

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: an update,” *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[15] G. Holmes, A. Donkin, and I. H. Witten, “Weka: A machine learning workbench,” in *Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on*, pp. 357–361, IEEE, 1994.

[16] C. McKay and I. Fujinaga, “jsymbolic: A feature extractor for midi files,” in *Proceedings of the International Computer Music Conference*, pp. 302–5, 2006.

[17] B. Abdullah, I. Abd-Alghafar, G. I. Salama, and A. Abd-Alhafez, “Performance evaluation of a genetic algorithm based approach to network intrusion detection system,” in *13th international conference on aerospace sciences and aviation technology, Military Technical College, Kobry Elkobbah, Cairo, Egypt*, 2009.

[18] K.-M. Osei-Bryson, “Post-pruning in decision tree induction using multiple performance measures,” *Computers & operations research*, vol. 34, no. 11, pp. 3331–3345, 2007.

[19] I. TREE, “Rough set based classification rule mining,”

[20] W.-Y. Loh, “Classification and regression tree methods,” *Encyclopedia of statistics in quality and reliability*, 2008.

[21] R. Kohavi *et al.*, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in *IJCAI*, vol. 14, pp. 1137–1145, 1995.

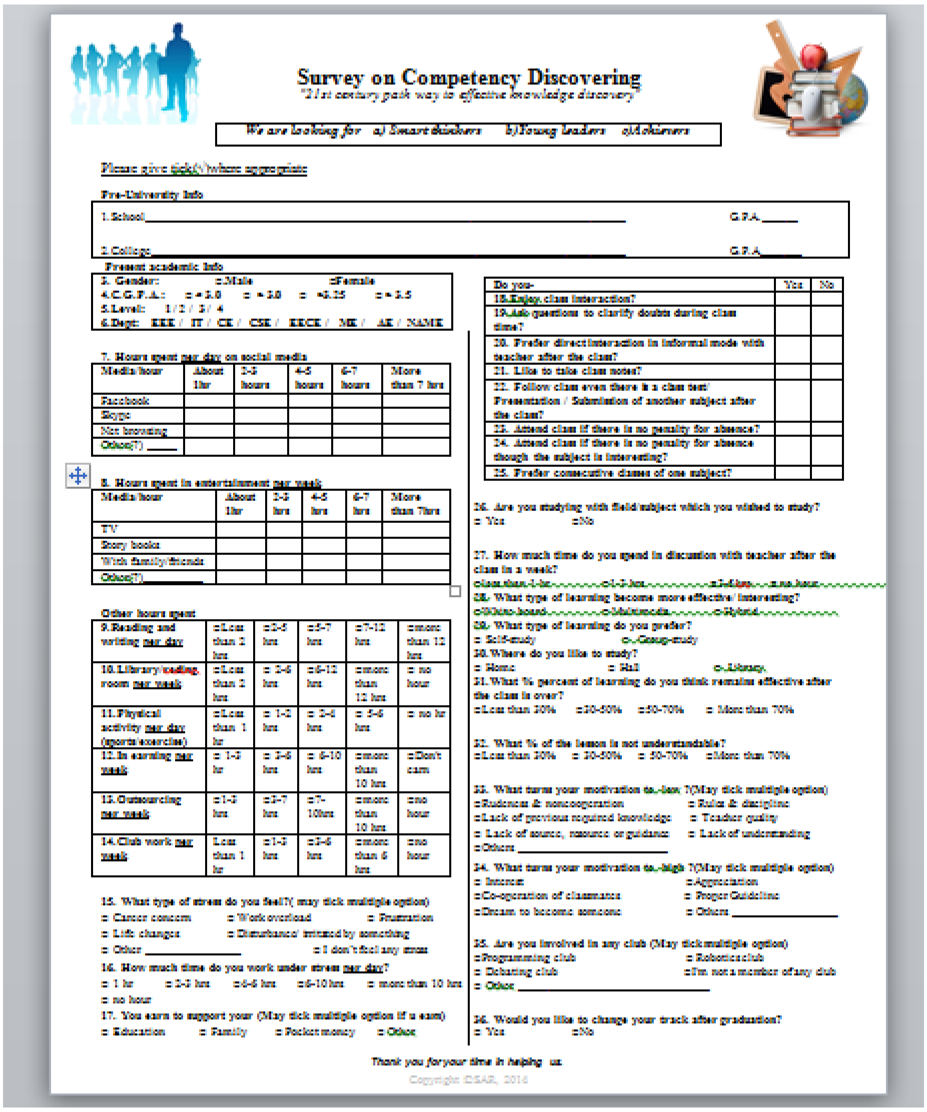
APPENDIX A
ALGORITHMS

A.1 Sample Algorithm

Algorithm

Algorithm 1 Sample Algorithm

Input: D = x1 ,x2 ,.... xn // Training data set , D, which contains a set of training instances and their associated class table. Output: T, Decision tree. Method: 1: T=0; 2: Determine best splitting attribute; 3: T=Create the root node and label it with the splitting attribute; 4: T=Add are to the root node and label it with the splitting attribute; 5: for each arc do; 6 D=Dataset created by applying splitting predicate to D; 7: if stopping point reached for this path, then 8: T'=Create a leaf node and label it with an appropiate class; 9: else 10: T'=DTBuild(D); 11: else if 12: T= Add T' to arc; 13: End for



Survey on Competency Discovering

"21st century path way to effective knowledge discovery"

We are looking for a) Smart thinkers b)Young leaders c)Achievers

Please give tick(√)where appropriate

Pre-University Info

1.School________________________________ G.P.A._____

2.College______________________________ G.P.A._____

Present academic Info

3. Gender: □Male □Female
4.C.G.P.A.: □ < 3.0 □ > 3.0 □ <3.25 □ > 3.5
5.Level: 1 / 2 / 3 / 4
6.Dept: EEE / IT / CE / CSE / EECE / ME / AE / NAME

7. Hours spent per day on social media

Media/hour	About 1hr	2-3 hours	4-5 hours	6-7 hours	More than 7 hrs
Facebook					
Skype					
Net browsing					
Other(?) _____					

8. Hours spent in entertainment per week

Media/hour	About 1hr	2-3 hrs	4-5 hrs	6-7 hrs	More than 7hrs
TV					
Story books					
With family/friends					
Other(?)_______					

Other hours spent

9.Reading and writing per day	□Less than 2 hrs	□2-5 hrs	□5-7 hrs	□7-12 hrs	□more than 12 hrs
10.Library/reading room per week	□Less than 2 hrs	□ 2-6 hrs	□6-12 hrs	□more than 12 hrs	□ no hour
11.Physical activity per day (sports/exercise)	□Less than 1 hr	□ 1-2 hrs	□ 2-4 hrs	□ 5-6 hrs	□ no hr
12.In earning per week	□ 1-3 hr	□ 3-6 hrs	□ 6-10 hrs	□more than 10 hrs	□Don't earn
13.Outsourcing per week	□1-3 hrs	□3-7 hrs	□7-10hrs	□more than 10 hrs	□no hour
14.Club work per week	Less than 1 hr	□1-3 hrs	□3-6 hrs	□more than 6 hrs	□no hour

15. What type of stress do you feel?(may tick multiple option)
□ Career concern □ Work overload □ Frustration
□ Life changes □ Disturbance/ irritated by something
□ Other ____________ □ I don't feel any stress

16. How much time do you work under stress per day?
□ 1 hr □ 2-3 hrs □4-6 hrs □6-10 hrs □ more than 10 hrs
□ no hour

17. You earn to support your (May tick multiple option if u earn)
□ Education □ Family □ Pocket money □ Other

Do you-	Yes	No
18.Enjoy class interaction?		
19.Ask questions to clarify doubts during class time?		
20. Prefer direct interaction in informal mode with teacher after the class?		
21. Like to take class notes?		
22. Follow class even there is a class test/ Presentation / Submission of another subject after the class?		
23. Attend class if there is no penalty for absence?		
24. Attend class if there is no penalty for absence though the subject is interesting?		
25. Prefer consecutive classes of one subject?		

26. Are you studying with field/subject which you wished to study?
□ Yes □No

27. How much time do you spend in discussion with teacher after the class in a week?
□Less than 1 hr □1-2 hrs □2-4 hrs □ no hour

28. What type of learning become more effective/ interesting?
□White board □Multimedia □Hybrid

29. What type of learning do you prefer?
□ Self-study □ Group-study

30.Where do you like to study?
□ Home □ Hall □ Library

31.What % percent of learning do you think remains effective after the class is over?
□Less than 30% □30-50% □50-70% □ More than 70%

32. What % of the lesson is not understandable?
□Less than 30% □ 30-50% □ 50-70% □More than 70%

33. What turns your motivation to low ?(May tick multiple option)
□Rudeness & noncooperation □ Rules & discipline
□Lack of previous required knowledge □ Teacher quality
□ Lack of source, resource or guidance □ Lack of understanding
□Others ______________

34. What turns your motivation to high ?(May tick multiple option)
□ Interest □Appreciation
□Co-operation of classmates □ Proper Guideline
□Dream to become someone □ Others __________

35. Are you involved in any club (May tick multiple option)
□Programming club □ Robotics club
□ Debating club □I'm not a member of any club
□ Other ________________

36. Would you like to change your track after graduation?
□ Yes □No

Thank you for your time in helping us

Figure A.1: Question