STUDENT PERFORMANCE EVALUATION

Enrol. No.	-	101268
Name of Student	-	Naveen Singla
Name of supervisor(s)	-	Asst. Prof. Suman Saha

Submitted in partial fulfillment of the Degree of

Bachelor of Technology

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,

WAKNAGHAT

TABLE OF CONTENTS

S.No. Topic Page No

CERTIFICATE

This is to certify that the work titled **Student Performance Evaluation** submitted by **Naveen Singla** in partial fulfillment for the award of degree of B. Tech of Jaypee University of Information Technology, Waknaghat has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor	……………………..
Name of Supervisor	Mr. Suman Saha
Designation	Assistant Professor
Date	……………………..

Acknowledgement

We would like to express our sincere gratitude to Mr. Suman Saha for providing us an opportunity to do our project on “Student Performance Evaluation using Machine Learning ” and rendering his help and guidance during the period of our project.

Signature of the student ……………………..

Name of Student Naveen Singla

Date ……………………..

Summary

The main perpose of the model is to predict the student performance in the final examination based on the previous records of the students. The Naïve Bayes and ID3 algorithms are trained using the training data set. The training inputs for the algorithm are directly taken from the stored file. After the algorithm is trained, then the inputs from the testing file are fed into the algorithm for prediction of the final result one by one. All the records are maintained in the stored file. In order to provide the input to the algorithm, an output window has been developed. The output window has been Coded, Developed and Designed in Net Beans. Each record from the training set is fed into the GUI interface one by one and then its output result is predicted and noted as either pass or fail. The process is repeated for all testing record set.

Signature of Student	Signature of Supervisor
Name:	Name:
Date:	Date:

1.0 Introduction

1.1 Abstract

Aim of this project is to make a machine learning model to predict automatically the performance of the students. A machine learning model made for this problem can infer the knowledge requirements of problems from student performance data, without analysis of the human? This model takes into consideration the various factors and the influence of the factors on the performance of student in education and predicts their result in the final exams whether Pass or Fail. Various factors such as previous academic results, attendance, family background , participation etc plays an important role in students' education. Some other factors that are not easy to measure such as student motivation, examination preparation level, student behavior , questions chosen for answering in the examination and way of expressing ideas may also be responsible for the final result of the student. I have taken the most common measurable factors of the students. Naive Bayes classifier and ID3(decision tree) algorithms are applied to calculate the performance.

1.2 **Objective**

Machine Learning can be used in educational field to enhance the understanding of learning process, so that we can focus on identifying, extracting and evaluating variables that are related to the student's process of learning . There are different objectives that are educationalfor using classification, such as: to discover potential student groups which have similar characteristics and reactions to some particular strategies , to group students who are hint driven and find out the common misconceptions , to identify learners which have low motivation and find out actions to lower drop out rates , to predict students when they are using the intelligent tutoring systems etc. Academic institutes are increasingly required to monitor the performance of the students . This gives rise to a need to collate, analyze and interpret the data, in order to have evidence to inform academic policies that aimed at, for example, improving student retention rates, allocating teaching resources, or creating intervention strategies to mitigate the factors that may affect the student performance adversely.

1.3 **About Machine Learning**

Machine learning is a branch of artificial intelligence. It is concern with the construction and study of the systems that can easily learn from the data. For eg, a machine learning system could be trained on the emails to learn to distinguish between spam and the messages that are not spam. After learning, it can then easily be used to classify new email messages into spam folders and the folders that are not spam.

Machine learning deals with the representation and the generalization. Representation of the data instances and functions that evaluated on these instances are part of the machine learning. Generalization is the property that system will perform well on unknown data ; the conditions under which this can be guaranteed are the key object of study in the field of computational theory.

We have a wide variety of the tasks and applications. Optical character recognition, in which printed characters recognized automatically based on previous examples, is an classic example of machine learning.

A core objective of the learner is to generalize from its experience. Generalization in this context is the ability of learning machine to perform accurately on new, examples/tasks that are not known after having experienced a learning dataset. The training examples comes from some generally probability distribution that is not known and the learner has to build a model that enables it to produce sufficiently accurate predictions in some unknown/new cases.

Some machine learning systems attempts to eliminate need for human in the data analysis, while some adopt collaborative approach between human and machine. Human intuition cannot, properly eliminated, since the system's designer must specify how data is to be represented and what mechanisms to be used to search for a characterization of the data.

Algorithm Types

Machine learning algorithms can easily be organized into a taxonomy based on the desired outcome of the algorithm or type of input available during training the machine.

- Supervised learning algorithms are trained based on labelled examples, i.e., input where output is known(needed). The supervised learning algorithm attempts to generalise a function or mapping from i/p to o/p which can then be used to speculatively generate an output for previously unknown inputs.
- Algorithms that are not supervised operate on the unlabelled examples, i.e., input where the desired output is not known. Here the objective is to find out structure in the data .
- Semi-supervised learning combines the labeled and unlabelled eg's to generate an appropriate classifier.
- Transduction interference tries to predict new outputs on particular and fixed test cases from observedand particular training cases.
- Reinforcement learning is concerned with how the agents that are intelligent ought to act in an environment to maximize some notion of the reward. The agent executes actions which cause observable state of environment to change. Through a sequence of actions, the agent attempts to gather knowledge about how the environment responds to its actions, and attempts to synthesize a sequence of actions that maximizes the reward.
- Learning to learn its own inductive bias based on experience(previous one).
- Developmental learning elaborated for the Robot learning, generates its own sequences of learning situations to acquire repertoires of novel skills through the social interaction with human teachers, and using guidance mechanisms such as the imitation, active learning, motor synergies.

2.0 Background material

2.1 Student Performance Evaluation

There is a growing interest among researchers that use data mining in educational technologies. There are several fields from which educational data mining methods are derived such as data mining and machine learning, psychometrics and other areas of statistics, information visualization, and computational modeling. A view point on educational data mining is given by Baker, which classifies work in educational data mining as Prediction, Clustering and Relationship mining. The process of the formation of significant models and assessment within Knowledge Discovery in Databases is referred to as data mining. The use of data mining techniques may improve the efficiency of educational institutions. If data mining techniques such as clustering, decision tree, association be applied to education processes, it can help improve student's performance, their life cycle management, their retention rate and grant/fund management of an institution. Now a day the data in the educational institutes has increased to many folds. These databases contain a lot of hidden information that can be used for improvement of students' performance. The performance in education is a turning point in the academic and professional life of all students. The ability to predict student's performance is very important in educational environments. The academic performance can be influenced by many factors; hence it is essential to apply a predictive data mining algorithm for student's performance so as to identify the different factors that highly influence student's academics. Moreover after having the entire factors one can also predict the final examination results.

2.2 Hardware and Software Requirements:

Software Requirements

1. JDK 1.7
    - NetBeans 7.1

2. Database
    - MySQL Database Server 5.0

3. Jdbc Driver for MySQL Database Server
    - mysql-connector-java-5.1.7-bin.jar

4. Operating System
    - Windows Vista / XP sp3 / Linux Fedora

Hardware Requirements:

1. Intel P4 processor with minimum 2.0Ghz Speed

2. RAM: Minimum 512MB

3. Hard Disk: Minimum 20GB

2.3 SYSTEM DESIGN

2.3.1 Tools and Technology used:

2.3.1.1 JAVA:

Java is basically a programming language that is concurrent, object-oriented, based on class and designed to have as few implementation dependencies as possible. It is used to let application developers "write once and run it anywhere", meaning that code that can easily runs on one platform does not need to compile again to run on another platform. Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of the computer architecture. Java is one of the popular programming languages that are in use, particularly for the client-server web applications. Java was originally developed by James Gosling at SunMicrosystems and released in 1995 as a core component of SunMicrosystems' Java platform. The language derives much of its syntax from C and C++, but it has some facilities than either of them.

The original and reference implementation Java compilers, virtual machines, and class libraries were developed by Sun from 1991. As of May2007, in the specifications of the Java Community Process, Sun licensed again most of its Java technologies under the General Public License. Others have also developed some implementations of these Sun technologies, such as the GNU Compiler for Java (byte code compiler), GNU Class path (standard libraries), and Iced Tea-Web (browser plug in for applets).

2.3.1.2 Swings (JAVA):

Swing is the primary Java GUI widget toolkit. It is part of Oracle's Java Foundation Classes — API to provide a graphical user interface for the java programs.

Swing was made to provide a more sophisticate set of GUI components than the older Abstract Window Toolkit . Swing provides a native look and feel that emulates the look of several platforms, and also supports a pluggable look and feel that allows applications to have a look unrelated to the underlying platform. It has more powerful and flexible components than Abstract Window Toolkit. In addition to familiar components such as buttons, check boxes , Swing provides some advanced components such as tabbed panel, scroll panes, tables, and lists.

Unlike the AWT components, Swing components are not implemented by platform-specific code. They are proper written in Java and therefore are platform independent. The term lightweight is used to describe such element.

2.3.1.3 Net Beans

Net Beans is an integrated development environment for developing primarily with Java, but also with other languages, in particular PHP, C/C++, and HTML5. It is also an application platform framework for Java desktop applications and others.

The Net Beans IDE is written in Java and can run on Windows, Linux, Solaris and other platforms supporting a compatible java virtual machine.

The Net Beans Platform allows applications to be developed from a set of modular software components. Applications based on the Net Beans Platform (including the Net Beans IDE itself) can be extended by third party.

The Net Beans Team actively support the product and seek feature suggestions from the community. Every release is preceded for the Community testing and feedback.

Net Beans Platform

Framework for simplifying the development of Java Swing applications. The Net Beans IDE bundle contain what is needed to start developing Net Beans plug ins and Net Beans Platform based applications; no need of additional SDK.

Applications can install modules dynamically. Any application can include the Update module to allow users of the application to download signed upgrades and some new features directly into the application that is running. Re installing an upgrade or a new release does not force users to download the whole application again.

The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their app. Among the features of the platform are:

- Users interface management.
- User settings management.
- Storage management.
- Window management.
- Wizard framework.
- Net Beans Visual Library

Net Beans IDE is a free, open source, cross platform which have built in support for java language.

2.3.1.4 MySQL

My SQL is the world's second most used relational database management system that is open sourced . It is named after co-founder Michael Widenius's daughter(My). The SQL phrase stands for Structured Query Language.

The My SQL development project has made its source code available under the terms of the General Public License, as well as under a variety of proprietary agreements. My SQL was owned and sponsored by the single firm, the Swedish company My SQL AB, now owned by Oracle Corporation.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack. LAMP is an acronym for "Linux, Apache, My SQL, Python, Perl." Free-software-open source projects that require full featured database management system usually use MySQL.

Interfaces

MySQL is the relational database management system , and ships with no graphic user interface tools to administer MySQL databases or manage the data contained in the databases. Users may use the included command line tools , desktop software and web applications that create and manage the MySQL databases, build database structures, back up data and work with data records. The MySQL front end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use for anyone.

3.0 PROJECT WORK:

3.1 Naïve Bayes approach:

A naïve bayes classifier is a simple probabilistic classifier and it is based on applying Bayes theorem with strong independent assumptions. A more descriptive term for this model is independent feature model. An overview of statistical classifiers is given in the article on Pattern recognition.

In simple terms, a naive Bayes classifier assumes that the presence or absence of a particular feature is not related to the absence or presence of any other feature, given the class variable. For example, a fruit may be considered to be an apple if its color is red, shape is round, and about 3" in diameter. A naive Bayes classifier considers these features to contribute independently to the probability that this fruit is an apple, regardless of the presence of all other features.

For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for this models uses the method of maximum likelihood; in other words, one can work with the naïve Bayes model without accepting the probability methods.

Despite their naive design and apparently over simplified assumptions, naive Bayes classifiers worked quite well in many complex real-world situations. An analysis of the Bayesian classification problem showed that there are sound theoretical reasons for the implausible efficacy of naive Bayes classifiers. Still, a comprehensive comparison with other classification algorithms showed that Bayes classification is performed better than other approaches.

An advantage of Naive Bayes is that it only requires small amount of data(training) to estimate the parameters necessary for classification. Because independent variables are assumed, only the variables for each class need to be determined and not the entire covariance matrix.

Bayesian learning algorithm is the most practical learning approach for most of the learning problems and is based on evaluating explicit probabilities. Bayes learning classifier is extremely competitive with the other learning algorithms and in many cases it performs better than other algorithms. Bayesian learning algorithms are extremely important in machine learning since they provide unique perspective for understanding many learning algorithms that cannot explicitly manipulate probabilities.

Bayes theorem states that:

$P(a_i/a_j) = (n_c+n*m)/(n+m)$

Where,

n= no. of training examples for which $v=v_j$

n_c= no. of training examples for which $v=v_j$ and $a=a_i$

p=a prior estimate for $p(a_i/v_j)$

m=equivalent sample size.

3.1.1 Data

The data set used in this study has been obtained has been manually stored in the MySQL database. The data comprises of two categories; Training set and testing set. The classification algorithms so used is trained using the training set and then tested using the testing data. A brief description of the attributes that is taken into consideration in this paper is given in the following table:

Sr. No.	**Variable Type**	**Attributes**
1	Locality Type	Village, Town or city
2	Parent Edu Status	Literate or illiterate
3	Financial Status	Low, medium or high
4	Parent’s visit to school	Nil, frequently or rare
5	Attendance	Good, medium or shortage
6	Extra Classes	Yes or no
7	Internet usage	Maximum, medium or Minimum

3.1.2 Implementation

Step wise description for calculations on a single data row:

Suppose we want to predict the result of a student with attributes as below:

Locality	Parent Edu. Status	Financial Status	Parents Visits school	Attendance	Extra Classes	Internet Usage	Previous Result
Village	Literate	Medium	Frequently	Good	Yes	Min	Fail

Here according to the Bayes theorm, we have to find the following values in order to find probabilities:

P(Village|Pass)

P(Literate|Pass)

P(Middle|Pass)

P(Frequent|Pass)

P(Good|Pass)

P(Yes|Pass)

P(Minimum|Pass)

And

P(Village|Fail)

P(Literate|Fail)

P(Middle|Fail)

P(Frequent|Fail)

P(Good|Fail)

P(Yes|Fail)

P(Minimum|Fail)

And multiply them by P(Pass) and P(Fail) respectively.

We estimated these values as follows:

Village:

Pass	Fail
n=20	n=20
n_c=10	n_c=9
p=0.33	p=0.33
m=3	m=3

Literate

Pass	Fail
n=20	n=20
n_c=5	n_c=8
p=0.5	p=0.5
m=3	m=3

Medium

Pass	Fail
n=20	n=20
n_c=5	n_c=12
p=0.33	p=0.33
m=3	m=3

Frequent

Pass	Fail
n=20	n=20
n_c=8	n_c=13
p=0.33	p=0.33
m=3	m=3

Good

Pass	Fail
n=20	n=20
n_c=14	n_c=11
p=0.33	p=0.33
m=3	m=3

Yes

Pass	Fail
n=20	n=20
n_c=5	n_c=9
p=0.5	p=0.5
m=3	m=3

Minimum

Pass	Fail
n=20	n=20
n_c=3	n_c=6
p=0.33	p=0.33
m=3	m=3

Since all the attributes are not binary, therefore the probability of each attribute is calculated as

p=(1/number of attribute values)

The value of m is arbitrary here. Here we have used m=3 but consistent for all attributes. Now applying equation (1) using the pre-computed value of n, n_c, p and m.

P(Village|Pass)=0.4739
P(Village|Fail)=0.4304
P(Literate|Pass)=0.2826
P(Literate|Fail)=0.4130
P(Medium|Pass)=0.2565
P(Medium|Fail)=0.5608
P(Frequent|Pass)=0.3869
P(Frequent|Fail)=0.6043
P(Good|Pass)=0.6478
P(Good|Fail)=0.5173
P(Yes|Pass)=0.2826
P(Yes|Fail)=0.4565
P(Minimum|Pass)=0.1695
P(Minimum|Fail)=0.3000

We have P(Pass)=0.5 and P(Fail)=0.5, so we can calculate the final probability as:

For v = Pass, we have

P(Pass)*P(Village|Pass)*P(Literate|Pass)*P(Middle|Pass)*P(Frequent|Pass)
*P(Good|Pass)*P(Yes|Pass)*P(Minimum|Pass)
=0.5*0.4739*0.2826*0.2565*0.3869*0.6478*0.2826*0.1695
=2.0620432659E-4
=0.0002062

For v = Fail, we have

P(Fail)*P(Village|Fail)*P(Literate|Fail)*P(Middle|Fail)*P(Frequent|Fail)

*P(Good|Fail)*P(Yes|Fail)*P(Minimum|Fail)

=0.5*0.4304*0.4130*0.5608*0.6043*0.5173*0.4565*0.3000

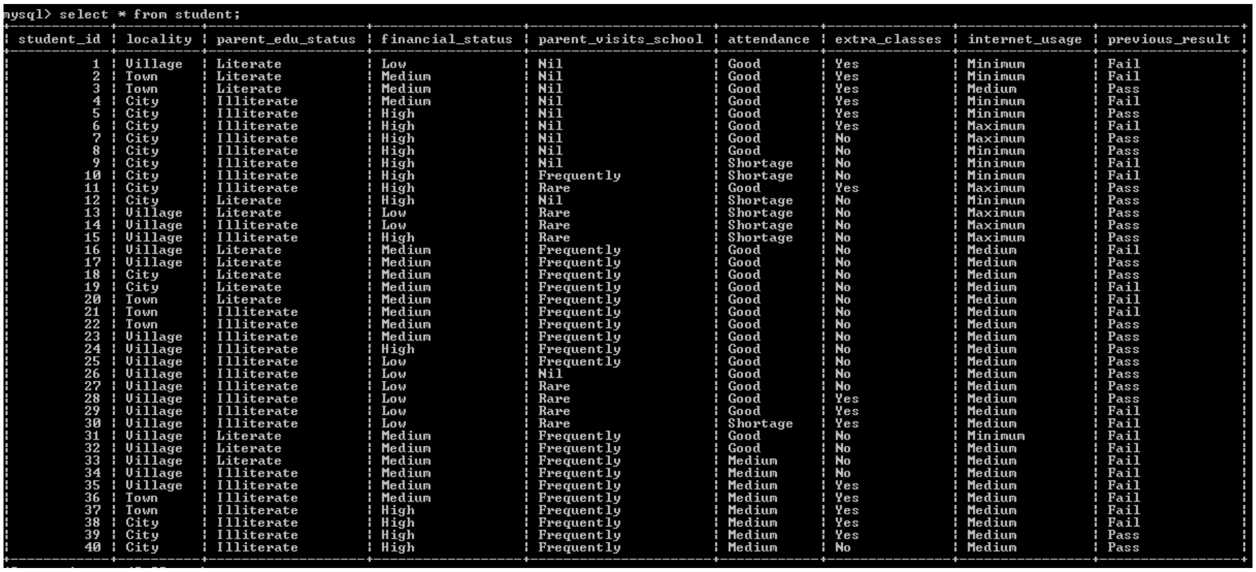
=0.002133818

Since 0.0002062<0.002133

The data element gets classified as "FAIL" (Same is shown in the output window)

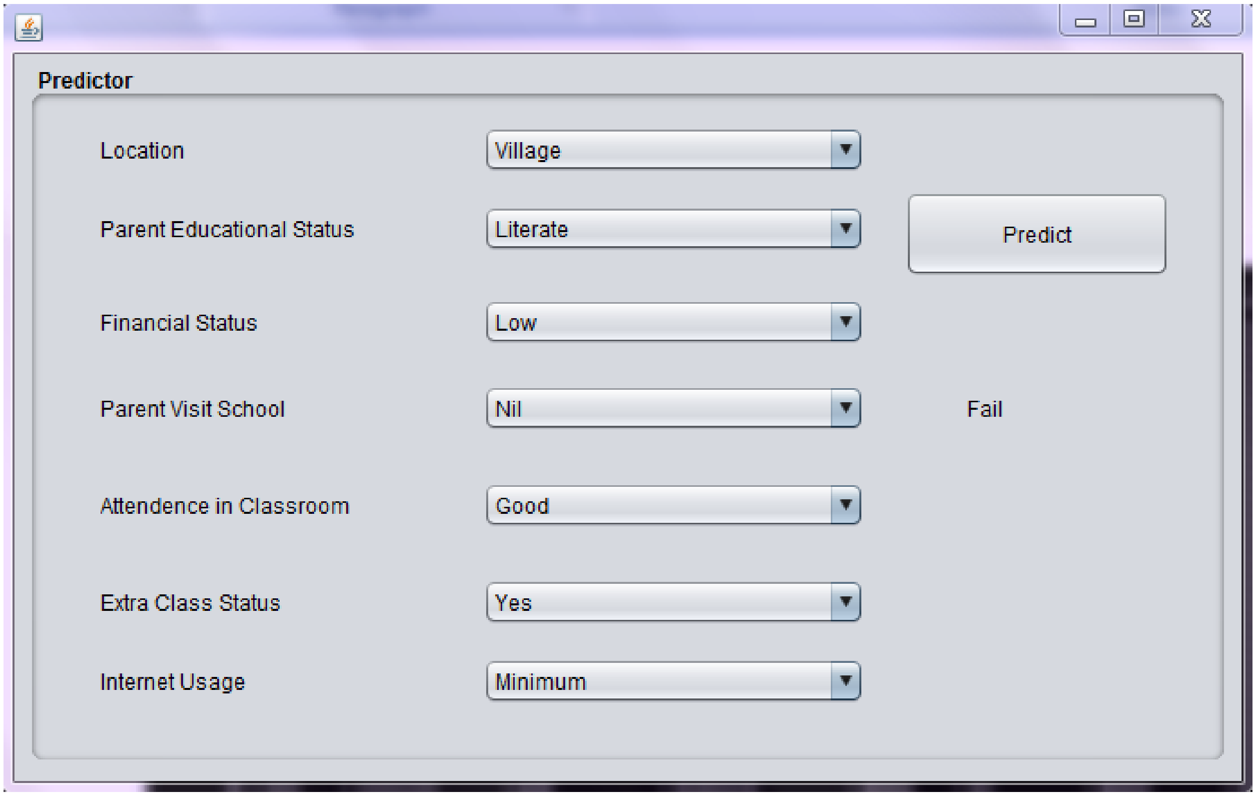
3.1.3 Snapshots

3.1.3.1 Database:



```
mysql> select * from student;
+------------+----------+-------------------+------------------+----------------------+------------+---------------+----------------+-----------------+
| student_id | locality | parent_edu_status | financial_status | parent_visits_school | attendance | extra_classes | internet_usage | previous_result |
+------------+----------+-------------------+------------------+----------------------+------------+---------------+----------------+-----------------+
|          1 | Village  | Literate          | Low              | Nil                  | Good       | Yes           | Minimum        | Fail            |
|          2 | Town     | Literate          | Medium           | Nil                  | Good       | Yes           | Minimum        | Fail            |
|          3 | Town     | Literate          | Medium           | Nil                  | Good       | Yes           | Medium         | Pass            |
|          4 | City     | Illiterate        | Medium           | Nil                  | Good       | Yes           | Minimum        | Fail            |
|          5 | City     | Illiterate        | High             | Nil                  | Good       | Yes           | Minimum        | Pass            |
|          6 | City     | Illiterate        | High             | Nil                  | Good       | Yes           | Maximum        | Fail            |
|          7 | City     | Illiterate        | High             | Nil                  | Good       | No            | Maximum        | Pass            |
|          8 | City     | Illiterate        | High             | Nil                  | Good       | No            | Minimum        | Pass            |
|          9 | City     | Illiterate        | High             | Nil                  | Shortage   | No            | Minimum        | Fail            |
|         10 | City     | Illiterate        | High             | Frequently           | Shortage   | No            | Minimum        | Fail            |
|         11 | City     | Illiterate        | High             | Rare                 | Good       | Yes           | Maximum        | Pass            |
|         12 | City     | Literate          | High             | Nil                  | Shortage   | No            | Minimum        | Pass            |
|         13 | Village  | Literate          | Low              | Rare                 | Shortage   | No            | Maximum        | Pass            |
|         14 | Village  | Illiterate        | Low              | Rare                 | Shortage   | No            | Maximum        | Pass            |
|         15 | Village  | Illiterate        | High             | Rare                 | Shortage   | No            | Maximum        | Pass            |
|         16 | Village  | Literate          | Medium           | Frequently           | Good       | No            | Medium         | Fail            |
|         17 | Village  | Literate          | Medium           | Frequently           | Good       | No            | Medium         | Pass            |
|         18 | City     | Literate          | Medium           | Frequently           | Good       | No            | Medium         | Pass            |
|         19 | City     | Literate          | Medium           | Frequently           | Good       | No            | Medium         | Fail            |
|         20 | Town     | Literate          | Medium           | Frequently           | Good       | No            | Medium         | Fail            |
|         21 | Town     | Illiterate        | Medium           | Frequently           | Good       | No            | Medium         | Fail            |
|         22 | Town     | Illiterate        | Medium           | Frequently           | Good       | No            | Medium         | Pass            |
|         23 | Village  | Illiterate        | Medium           | Frequently           | Good       | No            | Medium         | Pass            |
|         24 | Village  | Illiterate        | High             | Frequently           | Good       | No            | Medium         | Pass            |
|         25 | Village  | Illiterate        | Low              | Frequently           | Good       | No            | Medium         | Pass            |
|         26 | Village  | Illiterate        | Low              | Nil                  | Good       | No            | Medium         | Pass            |
|         27 | Village  | Illiterate        | Low              | Rare                 | Good       | No            | Medium         | Pass            |
|         28 | Village  | Illiterate        | Low              | Rare                 | Good       | Yes           | Medium         | Pass            |
|         29 | Village  | Illiterate        | Low              | Rare                 | Good       | Yes           | Medium         | Fail            |
|         30 | Village  | Illiterate        | Low              | Rare                 | Shortage   | Yes           | Medium         | Fail            |
|         31 | Village  | Literate          | Medium           | Frequently           | Good       | No            | Minimum        | Fail            |
|         32 | Village  | Literate          | Medium           | Frequently           | Good       | No            | Medium         | Fail            |
|         33 | Village  | Literate          | Medium           | Frequently           | Medium     | No            | Medium         | Fail            |
|         34 | Village  | Illiterate        | Medium           | Frequently           | Medium     | No            | Medium         | Fail            |
|         35 | Village  | Illiterate        | Medium           | Frequently           | Medium     | Yes           | Medium         | Fail            |
|         36 | Town     | Illiterate        | Medium           | Frequently           | Medium     | Yes           | Medium         | Fail            |
|         37 | Town     | Illiterate        | High             | Frequently           | Medium     | Yes           | Medium         | Fail            |
|         38 | City     | Illiterate        | High             | Frequently           | Medium     | Yes           | Medium         | Fail            |
|         39 | City     | Illiterate        | High             | Frequently           | Medium     | Yes           | Medium         | Pass            |
|         40 | City     | Illiterate        | High             | Frequently           | Medium     | No            | Medium         | Pass            |
+------------+----------+-------------------+------------------+----------------------+------------+---------------+----------------+-----------------+
```

3.1.3.2 Output Window of the system



Predictor

Location: Village

Parent Educational Status: Literate

Financial Status: Low

Parent Visit School: Nil

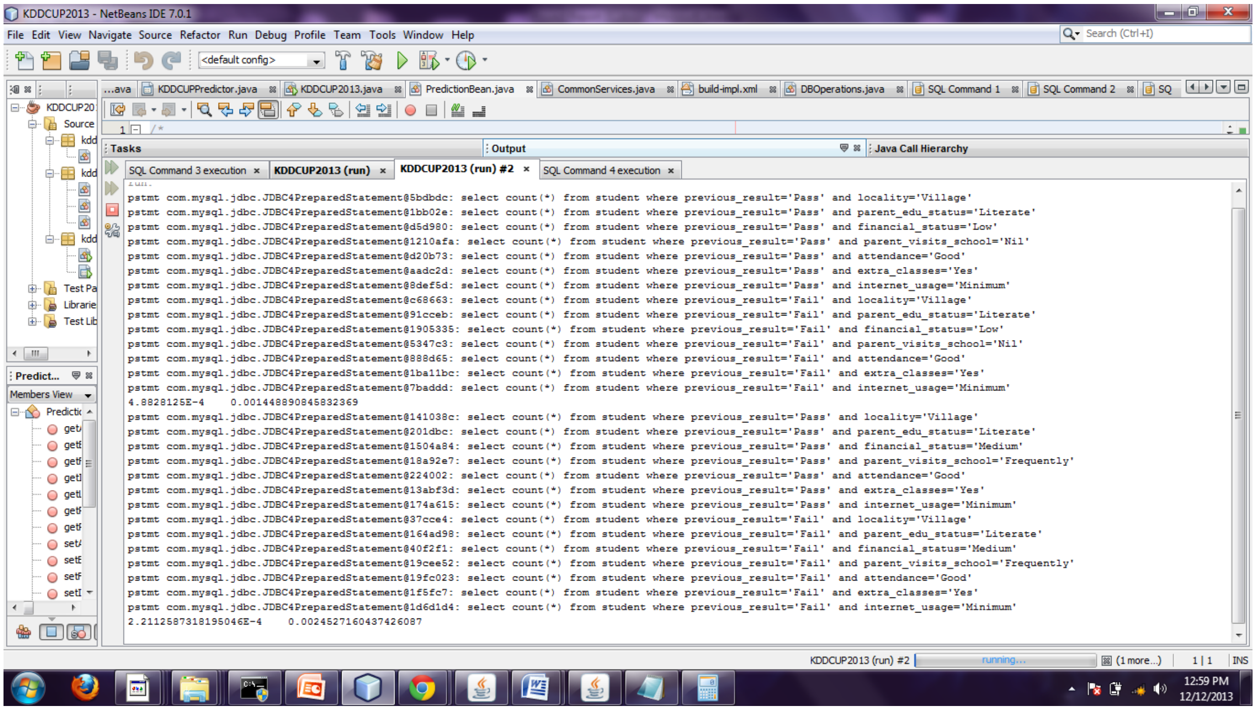
Attendence in Classroom: Good

Extra Class Status: Yes

Internet Usage: Minimum

Fail

3.1.3.3 Output window of Net Beans:



```
pstmt com.mysql.jdbc.JDBC4PreparedStatement@5bdbdc: select count(*) from student where previous_result='Pass' and locality='Village'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@1bb02e: select count(*) from student where previous_result='Pass' and parent_edu_status='Literate'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@d5d980: select count(*) from student where previous_result='Pass' and financial_status='Low'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@1210afa: select count(*) from student where previous_result='Pass' and parent_visits_school='Nil'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@d20b73: select count(*) from student where previous_result='Pass' and attendance='Good'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@aadc2d: select count(*) from student where previous_result='Pass' and extra_classes='Yes'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@8def5d: select count(*) from student where previous_result='Pass' and internet_usage='Minimum'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@c68663: select count(*) from student where previous_result='Fail' and locality='Village'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@91cceb: select count(*) from student where previous_result='Fail' and parent_edu_status='Literate'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@1905335: select count(*) from student where previous_result='Fail' and financial_status='Low'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@5347c3: select count(*) from student where previous_result='Fail' and parent_visits_school='Nil'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@888d65: select count(*) from student where previous_result='Fail' and attendance='Good'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@1ba11bc: select count(*) from student where previous_result='Fail' and extra_classes='Yes'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@7baddd: select count(*) from student where previous_result='Fail' and internet_usage='Minimum'
4.8828125E-4    0.001448890845832369
pstmt com.mysql.jdbc.JDBC4PreparedStatement@141038c: select count(*) from student where previous_result='Pass' and locality='Village'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@201dbc: select count(*) from student where previous_result='Pass' and parent_edu_status='Literate'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@1504a84: select count(*) from student where previous_result='Pass' and financial_status='Medium'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@18a92e7: select count(*) from student where previous_result='Pass' and parent_visits_school='Frequently'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@224002: select count(*) from student where previous_result='Pass' and attendance='Good'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@13abf3d: select count(*) from student where previous_result='Pass' and extra_classes='Yes'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@174a615: select count(*) from student where previous_result='Pass' and internet_usage='Minimum'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@37cce4: select count(*) from student where previous_result='Fail' and locality='Village'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@164ad98: select count(*) from student where previous_result='Fail' and parent_edu_status='Literate'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@40f2f1: select count(*) from student where previous_result='Fail' and financial_status='Medium'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@19cee52: select count(*) from student where previous_result='Fail' and parent_visits_school='Frequently'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@19fc023: select count(*) from student where previous_result='Fail' and attendance='Good'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@1f5fc7: select count(*) from student where previous_result='Fail' and extra_classes='Yes'
pstmt com.mysql.jdbc.JDBC4PreparedStatement@1d6d1d4: select count(*) from student where previous_result='Fail' and internet_usage='Minimum'
2.2112587318195046E-4    0.0024527160437426087
```

3.1.4 Source Code

3.1.4.1 DB connection:

```
package kddcup.services;
import java.sql.Connection;
import java.sql.DriverManager;

public class ConnectDB {

    public static Connection conn = null;

    public static Connection connect() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection("jdbc:mysql:///kddcup", "root", "root");
        } catch (Exception e) {
            System.out.println("Exception in Connection connect():" + e);
        }
        return conn;
    }
}
```

3.1.4.2 DB operations:

```
package kddcup.services;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class DBOperations {
    public int getNoOfStudents(String previousResult)
    {
        Connection conn=null;
        PreparedStatement pstmt=null;
        ResultSet rs=null;
        int count=0;
        try {
            conn=ConnectDB.connect();
            pstmt=conn.prepareStatement("select count(*) from student where previous_result=?");// may be problem9
            pstmt.setString(1, previousResult);
            rs=pstmt.executeQuery();
            if(rs.next())
            {
                count=rs.getInt(1);
            }
        } catch (Exception e) {
            System.out.println("Exeption in getNoOfStudent():"+e);
        }
        finally{
            try {
                pstmt.close();
                rs.close();
                conn.close();
            } catch (Exception e) {
            System.out.println("Exeption in getNoOfStudent():"+e);
            }
        }
        return count;
    }
    public int getNoOfStudentsByColName(String previousResult,String
colName,String colVal)
    {
        Connection conn=null;
        PreparedStatement pstmt=null;
        ResultSet rs=null;
        int count=0;
        try {
            conn=ConnectDB.connect();

            pstmt=conn.prepareStatement("select count(*) from student where
previous_result=? and "+colName+"=?");
            pstmt.setString(1, previousResult);
            pstmt.setString(2, colVal);
            System.out.println("pstmt "+pstmt);
            rs=pstmt.executeQuery();
            if(rs.next())
            {
                count=rs.getInt(1);

            }
        } catch (Exception e) {

            System.out.println("Exeption in getNoOfStudentsByColName():"+e);
        }
        finally{
            try {
                pstmt.close();
                rs.close();
                conn.close();
            } catch (Exception e) {
            System.out.println("Exeption in getNoOfStudentsByColName():"+e);
            }
        }
        return count;}}
```

3.1.4.3 Common Services:

```
package kddcup.services;
import kddcup.beans.PredictionBean;

public class CommonServices {

    double m=3;
    public double getProbabilityPassFail(PredictionBean objBean,String PassFail)
    {
        double p=1/3d;
        double p1=1/2d;

        double
totalProbabilityFail=0.5*optionPassFail("locality",objBean.getLocality().trim(),Pass
Fail,p)* // previously we set values in object bean,PassFail means value pass or
fail(see the get final result method)

optionPassFail("parent_edu_status",objBean.getParentEduStatus().trim(),PassFail,p1
)*
```

```
optionPassFail("financial_status",objBean.getFinancialStatus().trim(),PassFail,p)*

optionPassFail("parent_visits_school",objBean.getParentVisitStatus().trim(),PassFail
,p)*
                optionPassFail("attendance",objBean.getAttendence().trim(),PassFail,p)*

optionPassFail("extra_classes",objBean.getExtraClassStatus().trim(),PassFail,p1)*

optionPassFail("internet_usage",objBean.getInternetUsage().trim(),PassFail,p);
        return totalProbabilityFail;
    }

    public double optionPassFail(String column,String columnValue,String
PassFail,double p)  //call 7 times
    {
        DBOperations objDB=new DBOperations();
        double pOption=0;
        int noOfPassFail=objDB.getNoOfStudents(PassFail);
        int noOfOptionPassFail=objDB.getNoOfStudentsByColName(PassFail,
column, columnValue);

        pOption=(noOfOptionPassFail+(m*p))/(noOfPassFail+m);
        return pOption;
    }

    public String getFinalResult(PredictionBean objBean)
    {
        String result="Fail";
        double pPass=getProbabilityPassFail(objBean,"Pass");
        double pFail=getProbabilityPassFail(objBean,"Fail");
```

```
            System.out.println(pPass+"    "+pFail);
            if(pPass>pFail)
            {
               result="Pass";
            }
            return result;
        }
    }
```

3.1.4.4 Predictor:

```
    Private void btnPredictActionPerformed(java.awt.event.ActionEvent evt) {

            PredictionBean objBean=new PredictionBean();
            objBean.setLocality(ddlLocality.getSelectedItem().toString());
            objBean.setParentEduStatus(ddlParentEduStatus.getSelectedItem().toString());
            objBean.setFinancialStatus(ddlFinancialStatus.getSelectedItem().toString());

    objBean.setParentVisitStatus(ddlParentVisitStatus.getSelectedItem().toString());
            objBean.setAttendence(ddlAttendence.getSelectedItem().toString());
            objBean.setExtraClassStatus(ddlExtraClassStatus.getSelectedItem().toString());
            objBean.setInternetUsage(ddlInternetUsage.getSelectedItem().toString());

            CommonServices objCommonServices=new CommonServices();
            String result=objCommonServices.getFinalResult(objBean);
            lblResult.setText(result);
}
```

3.2Decision tree algorithm:

3.2.1 Introduction

Almost every year educational colleges admit students under various courses from different locations, educational background and with varying merit scores in entrance examinations. Apart from this schools and other junior colleges may be affiliated to the different boards, each board is having different subjects in their curricula and also have different level of depths in their subjects. Analyzing the previous performance of admitted students would provide a better perspective of the probable academics performance of students in the near future. This can easily be achieved by using the concepts of machine learning.

For this purpose, I have analyzed the data of students enrolled in past year of college. This data was obtained from the information that is provided by the admitted students to the college. It includes their parents education, attendance, previous academics record, participation in class etc. I then applied the ID3 algorithm at the dataset to predict the results of these students in their final examination as precisely as possible.

3.2.2 ID3 algorithm

In decision tree learning, ID3 (Iterative Dichotomiser 3) is an algorithm used to generate a decision tree from the given dataset. ID3 is typically used in the machine learning and natural language processing domains. This technique involves constructing a decision tree to model the classification process. Once the decision tree is built, it is then applied to each attribute in the database and results in the classification for that attribute. The following issues are faced by most decision tree algorithms:

- Choose the splitting attributes
- Order of the splitting attributes
- No of splits to take
- Balance of the tree structure
- Stopping criteria

The ID3 algorithm is a algorithm(classification) and it is based on Entropy and information gain, it is based on the idea that all examples are mapped to different categories according to different values of the attribute set; its core is to determine the best classification attribute form condition attribute sets. The algorithm chooses information gain as attribute selection criteria; usually the attribute that has the highest information gain is selected as the splitting attribute of current node, in order to make information entropy that the divided subsets need smallest. According to the different values of the attribute, branches can be established, and the process above is recursively called on International Journal of Data Mining & Knowledge Management Process each branch to create other nodes and branches until all the samples in a branch belong to the same category. To select the splitting attributes, the concepts of Entropy and Information Gain are used.

4.2.2.1 Entropy

Given probabilities p1, p2, …….., ps, where ∑ pi = 1, Formula of entropy is:

$$H(p1, p2, \ldots, ps) = \sum - (pi \log pi)$$

Entropy is used to find the amount of order in a given database . A value of (entropy)H = 0 identifies a perfectly classified set. In some other words, the higher is the "H", the higher is the potential to improve the classification process.

4.2.2.2 Information Gain

ID3 chooses the splitting attribute with the highest IG, where gain is defined as difference between how much information is needed after the split. This can easily be calculated by determining the differences between the entropies of the real dataset and the weighted sum of the entropies from each of the subdivided datasets. The formula to calculate the information gain is:

$$G(D, S) = H(D) - \sum P(Di)H(Di)$$

3.2.3 Data

The data set used in this study has been obtained has been manually stored in the MySQL database. The data comprises of two categories; Training set and testing set. The classification algorithms so used is trained using the training set and then tested using the testing data. A brief description of the attributes that is taken into consideration in this paper is given in the following table:

Sr. No.	Variable Type	Attributes
1.	Attendance	Good, medium or bad
2.	Previous academics record	Excellent , average or below average
3.	Parents_Education	Literate or illiterate
4.	Extra_Class	Yes or No

3.2.4 ID3 Example

- Create rootnode, containing the whole learning set as its subset:

  Entropy(rootnode.subset) =

  -(9/14)log2(9/14) -(5/14)log2(5/14)=0.940

- Compute information gain for each attribute

1) Entropy(no) = - (2/8)log2(2/8) -(6/8)log2(6/8)

   = 0.811

   Entropy(yes) = - (3/6)log2(3/6) -(3/6)log2(3/6)

   = 1

   Gain(S,Extra_Class) = Entropy(S) - (8/14)Entropy(no) - (6/14)Entropy(yes)

   = 0.048

2) Entropy(literate) = - (4/7)log2(4/7) -(3/7)log2(3/7)

   = 0.985

   Entropy(Illiterate) = - (1/7)log2(1/7) -(6/7)log2(6/7)

   = 0.591

   Gain(S,Parent_Edu_Status) = Entropy(S) - (7/14)Entropy(literate)

   - (6/14) Entropy(illiterate)

   = 0.151

3) Entropy(excellent) = - (2/4)log2(2/4) - (2/4)log2(2/4)

   = 1

   Entropy(average) = - (2/6)log2(2/6) -(4/6)log2(4/6)

   = 0.918

   Entropy(below average) = - (1/4)log2(1/4) -(3/4)log2(3/4) = 0.811

Gain(S,Prev_Acad_Record)=0.029

4) Entropy(bad) = - (3/5)log2(3/5) -(2/5)log2(2/5)

   = 0.970

   Entropy(medium) = - (4/4)log2(4/4) -

   = 0

   Entropy(good) = - (2/5)log2(2/5) -(3/5)log2(3/5)

   = 0.970

   Gain(S, Attendance) = 0.246.

Select attribute with the maximum information gain, which is ‘**attendance**' for splitting.

- Now we have to decide which attribute to split next.

1) Entropy(literate) = - (3/3)log2(3/3)

   = 0

   Entropy(Illiterate) = - (2/2)log2(2/2)

   = 0

   Gain(Sbad,parent_edu_status)=E(Sbad)-3/5(E literate)-2/5(E Illiterate)

   = 0.970

2) Entropy(excellent) = - (2/2)log2(2/2)

   = 0

   Entropy(average) = - (1/2)log2(1/2) -(1/2)log2(1/2)

   = 1

   Entropy(below average) = - (1/1)log2(1/1)

   = 0

Gain(Sbad,prev_acad_record)=0.970-(2/5)

= 0.57

3) Entropy(no) = - (2/3)log2(2/3) -(1/3)log2(6/8)
= 0.918

Entropy(yes) = - (1/2)log2(1/2) - (1/2)log2(1/2)

= 1

Gain(Sbad ,Extra_Class) = Entropy(S) - (3/5)Entropy(no) - (2/5)Entropy(yes)

= 0.019

Select attribute with the maximum information gain, which is "Parent_edu_status"' for splitting. Now, As Emedium is zero, it will not split further.

- Splitting of next attribute.

1) Entropy(average) = - (2/3)log2(2/3) -(1/3)log2(1/3)

= 0.918

Entropy(below average) = 1

Gain(Sgood,pev_acad_record) = E(Sgood)-3/5(E avg)-2/5(E below average)

= 0.019

2) Entropy(no) = - (3/3)log2(3/3)

= 0

Entropy(yes) = - (2/2)log2(2/2)

= 0

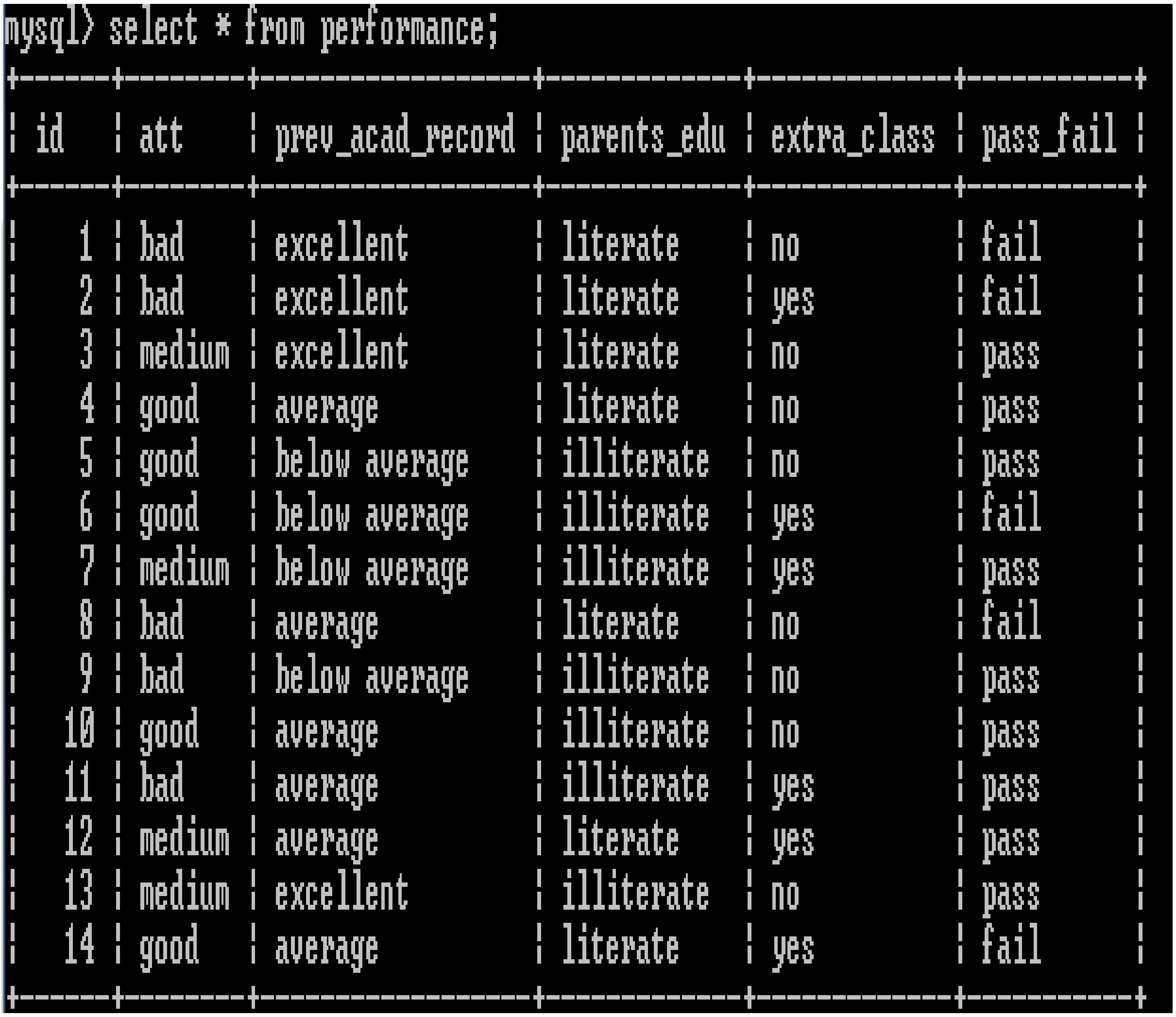
Gain(Sgood,Extra_Class) = Entropy(Sgood) - (3/5)Entropy(no) - (2/5)Entropy(yes)

= 0.970

Hence Extra_class will split below the good.

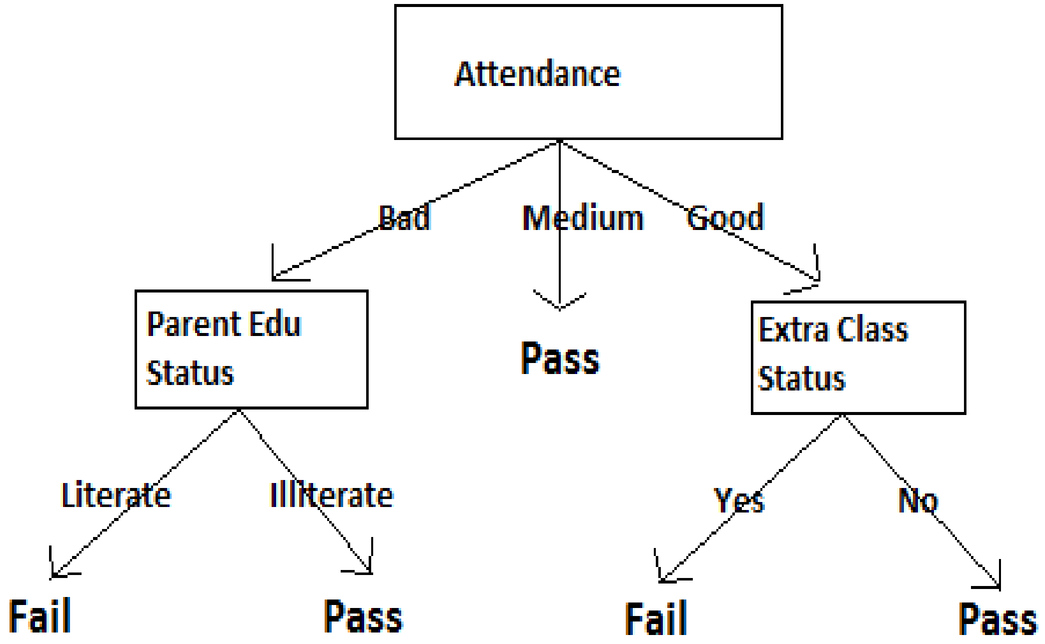
3.2.5 Snapshots:

3.2.5.1 Database:

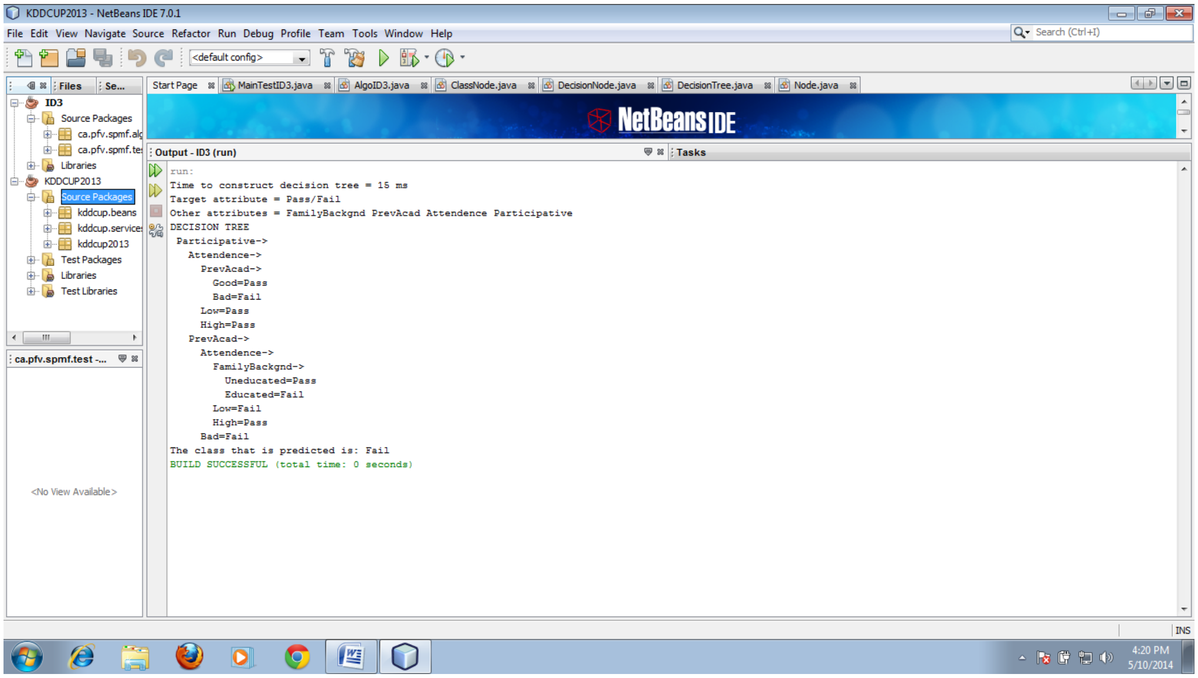


```
mysql> select * from performance;
+------+--------+------------------+-------------+-------------+-----------+
| id   | att    | prev_acad_record | parents_edu | extra_class | pass_fail |
+------+--------+------------------+-------------+-------------+-----------+
|    1 | bad    | excellent        | literate    | no          | fail      |
|    2 | bad    | excellent        | literate    | yes         | fail      |
|    3 | medium | excellent        | literate    | no          | pass      |
|    4 | good   | average          | literate    | no          | pass      |
|    5 | good   | below average    | illiterate  | no          | pass      |
|    6 | good   | below average    | illiterate  | yes         | fail      |
|    7 | medium | below average    | illiterate  | yes         | pass      |
|    8 | bad    | average          | literate    | no          | fail      |
|    9 | bad    | below average    | illiterate  | no          | pass      |
|   10 | good   | average          | illiterate  | no          | pass      |
|   11 | bad    | average          | illiterate  | yes         | pass      |
|   12 | medium | average          | literate    | yes         | pass      |
|   13 | medium | excellent        | illiterate  | no          | pass      |
|   14 | good   | average          | literate    | yes         | fail      |
+------+--------+------------------+-------------+-------------+-----------+
```

3.2.5.2 Decision tree output:



3.2.5.3 NetBeans output:



```
run:
Time to construct decision tree = 15 ms
Target attribute = Pass/Fail
Other attributes = FamilyBackgnd PrevAcad Attendence Participative
DECISION TREE
 Participative->
   Attendence->
     PrevAcad->
       Good=Pass
       Bad=Fail
     Low=Pass
     High=Pass
   PrevAcad->
     Attendence->
       FamilyBackgnd->
         Uneducated=Pass
         Educated=Fail
       Low=Fail
       High=Pass
     Bad=Fail
The class that is predicted is: Fail
BUILD SUCCESSFUL (total time: 0 seconds)
```

3.2.6 Implementation

3.2.6.2 DecisionTree:

```
package ca.pfv.spmf.algorithms.classifiers.decisiontree.id3;
/**
 This class represents a decision tree created by the ID3 algorithm.
* */
public class DecisionTree {

        String []allAttributes;
        Node root = null;
        public void print() {
                System.out.println("DECISION TREE");
                String indent = " ";
                print(root, indent, "");
        }

        private void print(Node nodeToPrint, String indent, String value) {
                String newIndent = indent + "  ";
                if(nodeToPrint instanceof ClassNode){
                        ClassNode node = (ClassNode) nodeToPrint;
                        System.out.println(indent + value + "="+ node.className);
                }else{

                        DecisionNode node = (DecisionNode) nodeToPrint;
                        System.out.println(indent + allAttributes[node.attribute] + "->");

                        for(int i=0; i< node.nodes.length; i++){
                                print(node.nodes[i], newIndent, node.attributeValues[i]);
                        }
                }


        }
        public String predictTargetAttributeValue(String[] newInstance) {
                return predict(root, newInstance);
        }
        private String predict(Node currentNode, String[] newInstance) {
                if(currentNode instanceof ClassNode){
                        ClassNode node = (ClassNode) currentNode;
                        return node.className;
                }else{
                        DecisionNode node = (DecisionNode) currentNode;
                        String value = newInstance[node.attribute];
                        for(int i=0; i< node.attributeValues.length; i++){
                                if(node.attributeValues[i].equals(value)){
                                        return predict(node.nodes[i], newInstance);
                                }
                        }
                }
                return null;
        }
}
```

3.2.6.3 MainTest ID3:

```
package ca.pfv.spmf.test;
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.URL;
import ca.pfv.spmf.algorithms.classifiers.decisiontree.id3.AlgoID3;
import ca.pfv.spmf.algorithms.classifiers.decisiontree.id3.DecisionTree;

/**
  Example of how to use ID3 from the source code.
 **/
public class MainTestID3 {
        public static void main(String [] arg) throws IOException{
                AlgoID3 algo = new AlgoID3();
                DecisionTree   tree   =   algo.runAlgorithm(fileToPath("Dataset_student.txt"),
"Pass/Fail", " ");
                algo.printStatistics();
                tree.print();
                String [] instance = {"Uneducated","Bad","Low","No",null};
                String prediction = tree.predictTargetAttributeValue(instance);
                System.out.println("The class that is predicted is: " + prediction);
        }
public static String fileToPath(String filename) throws UnsupportedEncodingException{
                URL url = MainTestID3.class.getResource(filename);
                 return java.net.URLDecoder.decode(url.getPath(),"UTF-8");
        }
}
```

4.0 Simulation Results and Performance Analysis:

The Naïve Bayes and ID3 algorithms are trained using the training data set. The training inputs for the algorithm are directly taken from the stored file. After the algorithm is trained, then the inputs from the testing file are fed into the algorithm for prediction of the final result one by one. All the records are maintained in the stored file. In order to provide the input to the algorithm, an output window has been developed. The output window has been Coded, Developed and Designed in Net Beans. Each record from the training set is fed into the GUI interface one by one and then its output result is predicted and noted as either pass or fail. The process is repeated for all testing record set.

5.0 Conclusion:

Aim of the project was to predict the performance of the student in the final examination., which is achieved by using the probability approach(naïve bayes algorithm) and decision tree(ID3 Algorithm). These two algorithms are trained using the training data set.After the training of these algorithms, inputs from the test files are fed into the algorithms for the prediction of student's result in the final examination.

References:

1) https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&sqi=2&ved=0CD0QFjAC&url=http%3A%2F%2Fwww.cs.sjsu.edu%2Ffaculty%2Flee%2Fcs157b%2FID3-AllanNeymark.ppt&ei=XVhyU8eNIY-hugT5n4HQCQ&usg=AFQjCNH6QJdF9adVDWhyGkoikFfCKup3dg&sig2=OQuiMPRes7FnRHtzu0AqVg&bvm=bv.66330100,d.c2E2. JAVA:
   Java : The Complete Reference 7th Edition by Herbert Schildt
   http://en.wikipedia.org/wiki/Java_(programming_language)
2. MySQL:
   MySQL 5.1 Plugin Development by Sergei Golubchik
   SQL : The Complete Reference 2nd Edition by Paul Weinberg, James Groff
3. Thechallengedescriptionisfrom
   http://www.sigkdd.org/kdd-cup-2010-student-performance-evaluation
4. http://www.ijser.org/researchpaper%5CStudents-Examination-Result-Mining-A-Predictive-Approach.pdf

5 http://www.cse.msu.edu/rgroups/cse101/ITS/its.htm

6. http://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/2.htmC.Romero, S.Ventura, P.G. Espejo, and C. Hervas, “Data Mining Algorithms to Classify Students”.
7. https://www.cs.umd.edu/grad/scholarlypapers/papers/Bahety.pdf