STUDENT PERFORMANCE EVALUATION USING MACHINE LEARNING

Enrollment No. 101301
Name of Student: Anshul Mittal
Name of Supervisor: Mr. Suman Saha

Submitted in partial fulfillment of the Degree of

Bachelor of Technology

Department of Computer Science and Engineering,
Jaypee University Of Information Technology.

TABLE OF CONTENTS

CERTIFICATE

This is to certify that the work titled '**Student Performance Evaluation using Machine Learning'** submitted by **Anshul Mittal** in partial fulfillment for the award of degree BTech 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

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

Signature of the student ……..

Name of Student Anshul Mittal

Date ……..

SUMMARY

An educational institution wants to have an approximate prior knowledge of the result of their students whether pass or fail. This will help the institution to find out good students and also will provide an opportunity to pay attention to those who get lower grades. Thus, a system has been developed which can predict students' performance from previous students' performances using data mining's techniques. A data set has been analyzed, containing information about students, such as Parents' Education, locality, financial status, marks etc.scored in the examinations. By applying the ID3 (Iterative Dichotomiser 3) on this data, I have predicted the individual performance of the newly admitted students can be predicted.

_____	_____
Signature of Student	Signature of Supervisor
Name: Anshul Mittal	Name: Mr. Suman Saha
Date	Date

1.0 Introduction

1.1 Problem Statement

The aim of this project is to predict whether a student will pass or fail by taking into considerations various factors such as Locality, Parents' Education, Financial Status, Parents' visit to school, Attendance, Extra Class, Internet Usage etc.

1.2 Objective:

Data Mining helps us in educational field to enhance the understanding of learning process to focus on identifying, extracting and evaluating variables related to the learning process of students . There are various educational objectives for using classification, such as: to discover potential student groups with similar characteristics and reactions to a particular pedagogical strategy , to detect students' misuse or game-playing , to group students who are hint-driven or failure-driven and find common misconceptions that students possess , to identify learners with low motivation and find remedial actions to lower drop-out rates , to predict/classify students when using intelligent tutoring systems etc. Academic institutions are increasingly required to monitor their performance and the performance of their students. This gives rise to a need to collate, analyze and interpret data, in order to have evidence to inform academic policies that are aimed at, for example, improving student retention rates, allocating teaching and support resources, or creating intervention strategies to mitigate factors that may affect student performance adversely.

1.3 About Machine Learning

Machine learning is a part of artificial intelligence. The main function of machine learning is to represent and generalize. Machine Learning Systems comprises Data instances' representation and functions. The property of Generalization will perform on unseen data instances.

There are many types of machine learning tasks and successful applications. Optical character recognition, is a classic example of machine learning.

Human Intuition is eliminated with the help of machine learning systems, while others adopt a collaborative approach between human and machine. Human intuition cannot be entirely eliminated, because how the data will be represented and what type of mechanisms will be used to search for characters of the data is specified by system's designer.

Algorithm Types

On the basis of outcome of algorithm, Machine learning algorithms can be organized into a taxonomy based on the desired outcome of the algorithm

- Supervised learning algorithms are trained on labelled examples. To generalize a function or mapping, Supervised Learning Algorithm is attempted.
- Unsupervised learning algorithms operate on unlabelled examples, In this case, objective is to find out the structure in the data and not to do mapping's generalization.
- Combination of both labled and unlabeled examples becomes Semi-supervised learning.
- To predict new outputs on specific and fixed cases, Transduction is used.
- How intelligent agents can be used to act in a filed to maximise some type of reward, concerns Reinforcement learning. The agent executes actions changes the observable state of filed.

1.4 About Student Performance Evaluation

There is a growing interest among researchers that use data mining in educational technologies, or educational data mining (EDM). There are several fields from which EDM 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 – KDD is referred to as data mining. The use of data mining techniques may improve the efficiency of educational institutions. 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 is influenced by many factors; hence it is essential to apply a predictive data mining algorithm for students' 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. In the present investigation, we had 250 student records, which were used by Bayes classification prediction to predict the final examination results.

Report Overview

In chapter-1, literature Survey, there is a brief introduction about Data Mining and ID3 algorithm. Chapter-2, System Design includes the various technologies such as Java and Eclipse in making this project. Chapter-3 Previous Project Work includes the project work done in last semester. In this section, Naïve Bayes Algorithm is used. In chapter-5 Project Work, various aspects of the project work done in the current semester is given. In the last chapter, Conclusion, a brief overall view of what has been done in the whole project is shown.

2.0 Literature Survey

2.1 Data Mining

Data mining is the process of discovering knowledge which is of great interest, from large size of data which is stored in databases. This process is used in last few years because of availability of large amounts of data in digital form, and there is also a need for transforming such form of data into some useful information and knowledge.

2.1.1 Classification

A data mining technique that maps data into already defined groups or classes is known as Classification. Supervised learning method is used to classify test data into already determined groups or classes. It is made of two phases. The first phase is the learning phase. The second phase is of the classification. In Classification algorithms it is require that classes must be defined based on some data attribute values

2.1.2 Clustering

In Clustering such elements which are similar or common in nature, are grouped together. The common technique used for statistical data analysis is used in various fields such as pattern recognition. Various algorithms can achieve Clustering. These algorithms differ about the similarities which are required between elements of a cluster. Various algorithms are used for clustering which try to create clusters from small distances.

2.2 ID3 Algorithm

ID3 (Iterative Dichotomiser 3) is an algorithm which was invented by Ross Quinlan that is used to generate a decision tree. In the machine learning and natural language processing domains, ID3 is mainly used. The following issues are faced by many of the decision tree algorithms:

- Choosing splitting attributes
- Ordering of splitting attributes
- Number of splits to take
- Balance of tree structure and pruning
- Stopping criteria

The ID3 algorithm is a classification algorithm which is based on Entropy and Information Gain. The basic idea of this algorithm is that according to different values of the condition , examples are mapped to different type of categories. Its main function is to determine the best attribute from the given condition attribute sets.

Information gain is selected as the attribute selection criteria in the algorithm. Generally the attribute which gives the most information gain value is selected as the current node's splitting attribute so that that the divided subsets need smallest.

2.2.1 Entropy

Given probabilities p1, p2, …, ps, where $\sum pi = 1$, Entropy is defined as

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

Entropy finds the amount of order in a given database state. A value of H = 0 identifies a perfectly classified set. In other words, the higher the entropy, the higher the potential to improve the classification process.

3. SYSTEM DESIGN

Tools and Technology used:

3.1 JAVA:

Java is a computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. Java applications are typically compiled to bytecode (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture.. Java was originally developed by James Gosling at Sun Microsystems (which has since merged into Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level 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 May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Classpath (standard libraries), and IcedTea-Web (browser plugin for applets).

3.2 Swings (JAVA):

Swing is the primary Java GUI widget toolkit. It is part of Oracle's Java Foundation Classes (JFC) — an API for providing a graphical user interface (GUI) for Java programs.

To give a sophisticated set of GUI components, Swings was introduced. Swing gives a native look and feel that allows applications to have a look and feel unrelated to the. Swings provide buttons, check. boxes, labels, tables and lists.

Swing components are not implemented by platform-specific code. They are written entirely in Java and thus they are platform-independent..

3.3 Eclipse

In computer programming, **Eclipse** is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications. By means of various plug-ins, Eclipse may also be used to develop applications in other programming languages: Ada, ABAP, C, C++, COBOL,Fortran, Haskell, JavaScript, Lasso, Natural, Perl, PHP, Py thon, R, Ruby (including Ruby on Rails framework), Scala, Clojure, Groovy, Scheme, and Erlang. It can also be used to develop packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM VisualAge. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

Released under the terms of the Eclipse Public License, Eclipse SDK is free and open source software (although it is incompatible with the GNU General Public License[3]). It was one of the first IDEs to run under GNU Classpath and it runs without problems under IcedTea.

4.0 PREVIOUS PROJECT WORK:

4.1 Naïve Bayes approach:

A **naive Bayes classifier** is a probabilistic classifier which is based on applying Bayes' theorem with strong (naive) independence assumptions. A better term for the probability model is "independent feature model".

In supervised learning setting, naive Bayes classifiers. In many practical applications, maximum likelihood is used in parameter estimation for naive Bayes models. In other words, naive Bayes model can be worked without accepting Bayesian probability.

Despite the naive design of naive Bayes classifiers, Bayes classifiers have worked very well in various difficult real situations.

Naive Bayes only need a small amount of training data to estimate the parameters which are necessary for classification process. Since independent variables are assumed, each class' variances of variables need to be determined.

Bayes theorem states that:
$P(a_i/v_j)= (n_c+m*p)/(n=m)$

Where:
n= the number of training examples for which $v=v_j$
n_c=the number of examples for which $v=v_j$ and $a=a_i$
p=priori estimate for $P(a_i/v_j)$
m=the equivalent sample size

4.2 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 has been 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 Educational Status	Literate or illiterate
3	Financial Status	Low, medium or high
4	Parent Visit to school	Nil, frequently or rare
5	Attendance in class room	Good, medium or shortage
6	Extra Classes status	Yes or no
7	Internet usage	Maximum, medium or Minimur

4.3 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	Middle	Frequent	Good	Yes	Min	Passes

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 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

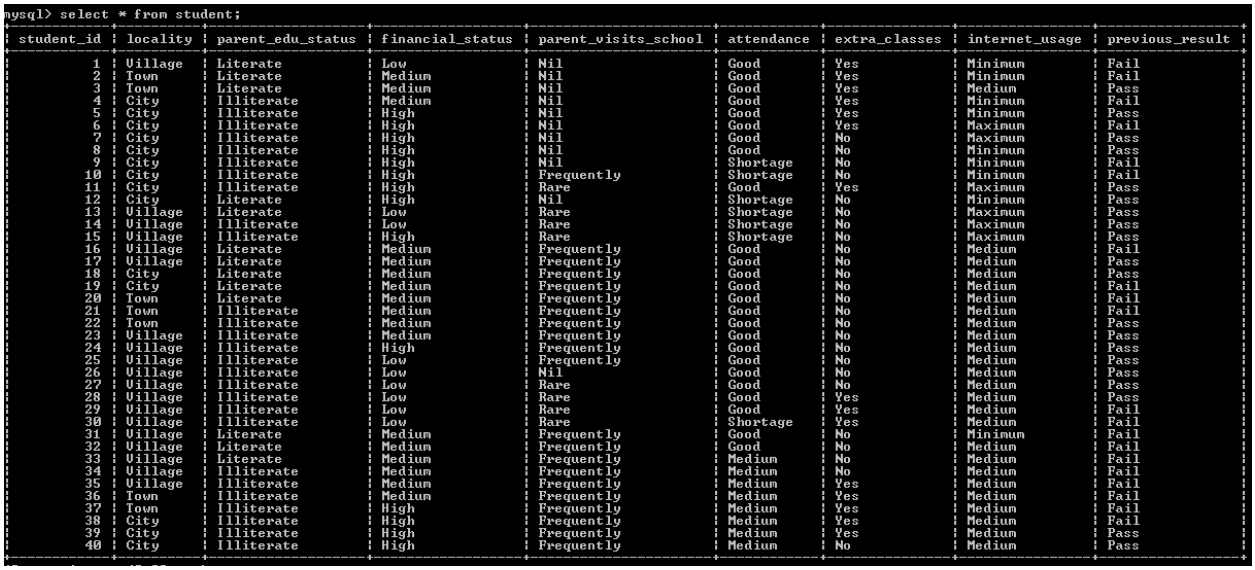
=0.002133818

Since 0.0002062<0.002133

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

4.4 Snapshots

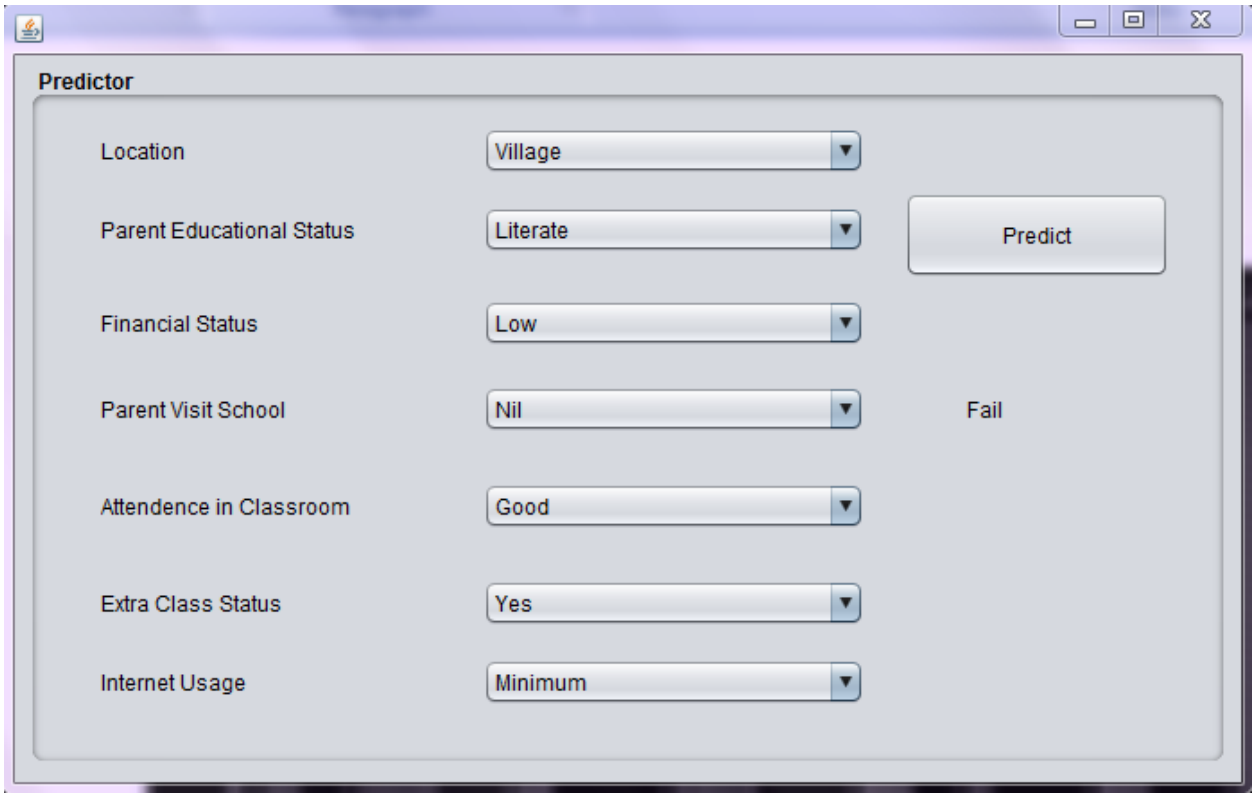
4.4.1 Database:



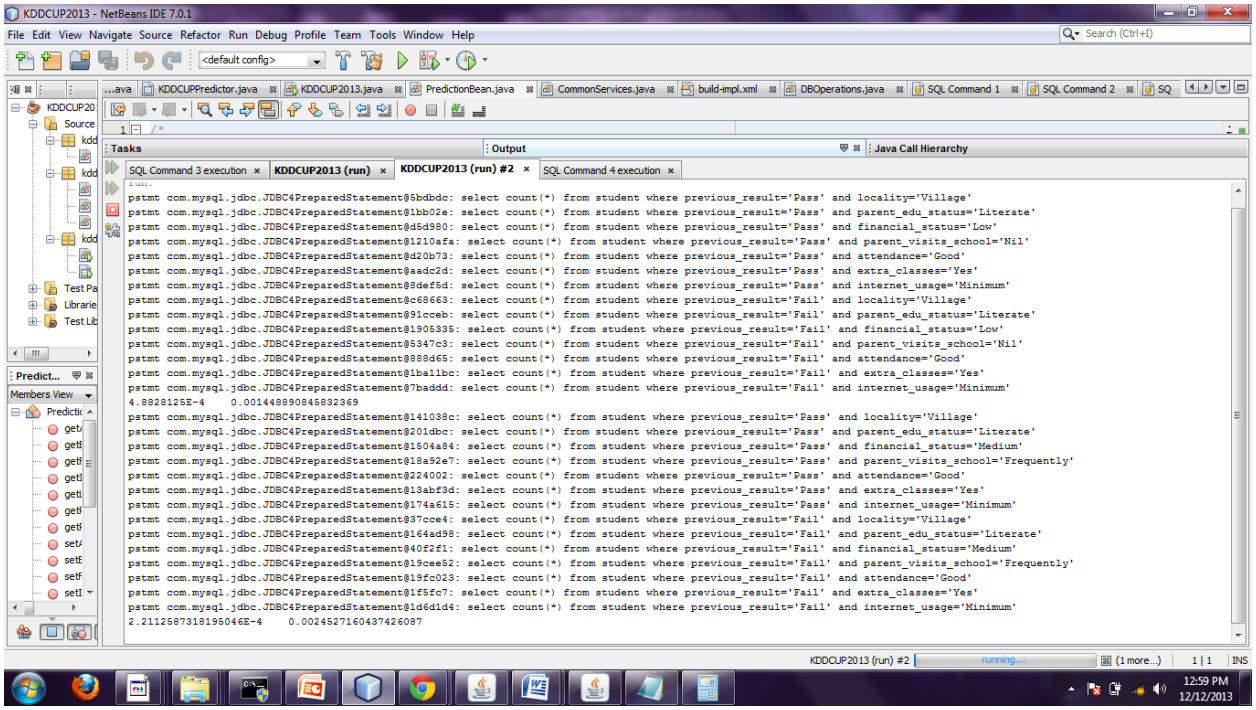
mysql> select * from student;

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

4.4.2 Output Window of the system



4.4.3 Output window of Net Beans:



4.4 Source Code

4.4.1 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;
   }
}
```

4.4.2 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;}}
```

4.4.3 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(),PassFail,p)*

   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;
    }
}
```

4.4.4 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);

}
```

5.0 PROJECT WORK

5.1 Iterative Dichotomiser (ID3) Algorithm

In decision tree learning, **ID3** (**Iterative Dichotomiser 3**) is an algorithm invented by Ross Quinlan used to generate a decision tree from a dataset. ID3 is the precursor to the C4.5 algorithm, and is typically used in the machine learning and natural language processing domains.

The original set S as the root node begins the ID3 algorithm. On every iteration of the algorithm, every unused attribute of the set S is iterated and entropy H(S) of that attribute is calculated. Then the attribute which has the smallest entropy (or largest information gain) value is selected. Each subset of the set is recurvily iterated.

Recursion on a subset can be stopped in one of the following cases:

- each element of the subset belongs to the same class, then the leaf is turned from a node and it is labelled with the class of the examples
- if no more attributes can be selected, but the examples do not belong to the same class , then .the leaf is turned from a node and it will be labelled with the most common class of the examples in the subset
- if in the subset, there are no more examples, this type of case occurs when no example in the parent set is matching a specific value of the selected attribute.

In this algorithm, the decision tree is constructed with the concepts of Entropy and Information Gain.

Summary

1. Calculate the entropy of every attribute using the data set S
2. Split the set S into subsets using the attribute for which entropy is minimum (or, equivalently, information gain is maximum)
3. Make a decision tree node containing that attribute
4. Recurse on subsets using remaining attributes

ID3 Metrics

Entropy

Entropy H(S) is a measure of the amount of uncertainty in the (data) set S (i.e. entropy characterizes the (data) set S).

$$H(S) = -\sum_{x \in X} p(x) \log_2 p(x)$$

Where,

- S - The current (data) set for which entropy is being calculated (changes every iteration of the ID3 algorithm)
- X - Set of classes in S
- $p(x)$ - The proportion of the number of elements in class x to the number of elements in set S

When $H(S) = 0$, the set S is perfectly classified (i.e. all elements in S are of the same class).

In ID3, entropy is calculated for each remaining attribute. The attribute with the smallest entropy is used to split the set S on this iteration. The higher the entropy, the higher the potential to improve the classification here.

5.2 Data

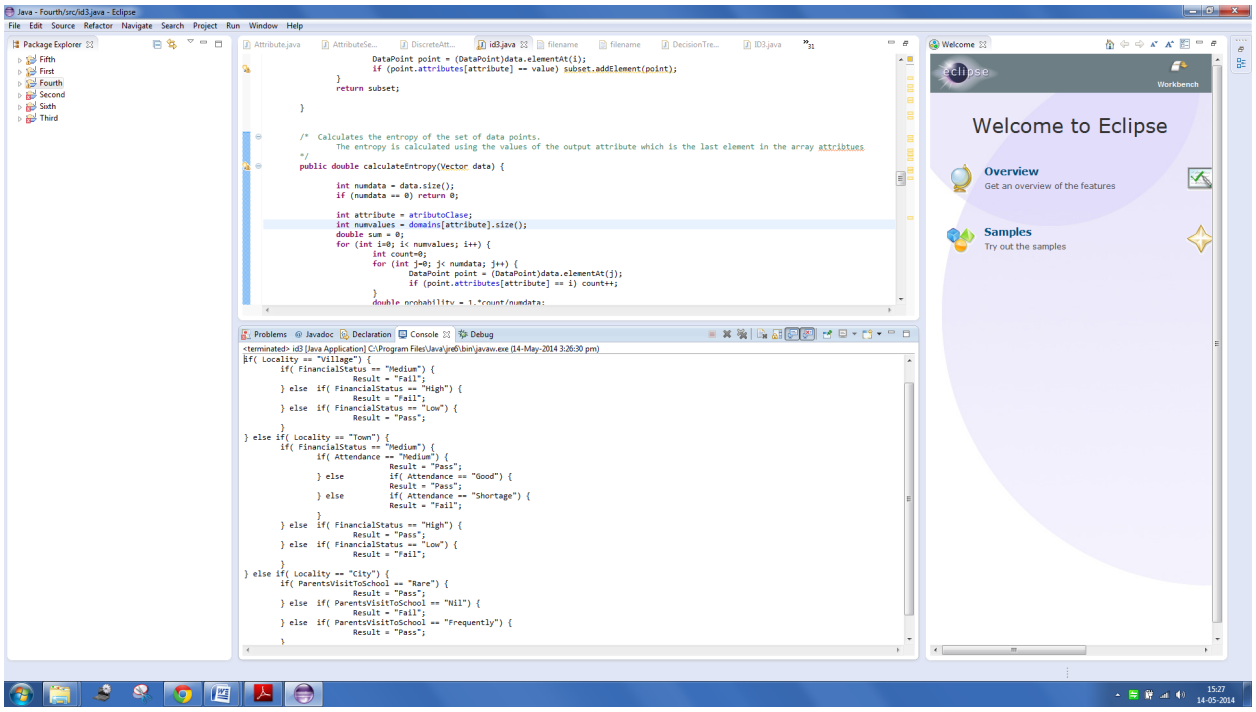
The data set used in this study has been obtained has been manually stored in the text file. The data comprises of two categories; Training set and testing set. The classification algorithms so used is trained using the training set and then tested using the testing data. A brief description of the attributes that has been 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 Educational Status	Literate or illiterate
3	Financial Status	Low, medium or high
4	Parent Visit to school	Nil, frequently or rare
5	Attendance in class room	Good, medium or shortage
6	Extra Classes status	Yes or no
7	Internet usage	Maximum, medium or Minimur

5.3 Implementation

Implementation has been divided into three stages. In the first stage, information about students of the training set is collected. In the second stage, extra information was removed from the database and relevant information will remain in database. The third stage involved applying the ID3 on the training data to obtain decision trees

5.3.1 Output Window of Eclipse



5.4 Source Code

```
public class id3 {


    int numAttributes;
    String []attributeNames;
    private int atributoClase;
    Vector []domains;
    public int readData(String filename)  throws Exception {

      FileInputStream in = null;


      try {
            File inputFile = new File(filename);
            in = new FileInputStream(inputFile);
      } catch ( Exception e) {
            System.err.println( "Unable to open data file: " + filename + "\n" + e);
            return 0;
      }

      BufferedReader bin = new BufferedReader(new InputStreamReader(in) );

      String input;
      while(true) {
            input = bin.readLine();
            if (input == null) {
                  System.err.println( "No data found in the data file: " + filename + "\n");
                  return 0;
            }
```

```
        if (input.startsWith("//")) continue;
        if (input.equals("")) continue;
        break;
}


StringTokenizer tokenizer = new StringTokenizer(input);
numAttributes = tokenizer.countTokens();
if (numAttributes <= 1) {
        System.err.println( "Read line: " + input);
        System.err.println( "Could not obtain the names of attributes in the line");
        System.err.println( "Expecting at least one input attribute and one output attribute");
        return 0;
}

domains = new Vector[numAttributes];
for (int i=0; i < numAttributes; i++) domains[i] = new Vector();
attributeNames = new String[numAttributes];

for (int i=0; i < numAttributes; i++) {
        attributeNames[i]  = tokenizer.nextToken();
}


while(true) {
        input = bin.readLine();
        if (input == null) break;
        if (input.startsWith("//")) continue;
        if (input.equals("")) continue;
```

```
            tokenizer = new StringTokenizer(input);
            int numtokens = tokenizer.countTokens();
            if (numtokens != numAttributes) {
                  System.err.println( "Read " + root.data.size() + " data");
                  System.err.println( "Last line read: " + input);
                  System.err.println( "Expecting " + numAttributes  + " attributes");
                  return 0;
            }

            DataPoint point = new DataPoint(numAttributes);
            for (int i=0; i < numAttributes; i++) {
                  point.attributes[i]  = getSymbolValue(i, tokenizer.nextToken() );
            }
            root.data.addElement(point);

      }

      bin.close();

      return 1;}


         public int []attributes;

         public DataPoint(int numattributes) {
               attributes = new int[numattributes];
         }
};
```

```
class TreeNode {
	public double entropy;
	public Vector data;
	public int decompositionAttribute;
	public int decompositionValue;
	public TreeNode []children;
	public TreeNode parent;

	public TreeNode() {
		data = new Vector();
	}

};

TreeNode root = new TreeNode();


public int getSymbolValue(int attribute, String symbol) {
	int index = domains[attribute].indexOf(symbol);
	if (index < 0) {
		domains[attribute].addElement(symbol);
		return domains[attribute].size() -1;
	}
	return index;
}

public int []getAllValues(Vector data, int attribute) {
```

```
        Vector values = new Vector();
        int num = data.size();
        for (int i=0; i< num; i++) {
            DataPoint point = (DataPoint)data.elementAt(i);
            String symbol = (String)domains[attribute].elementAt(point.attributes[attribute] );
            int index = values.indexOf(symbol);
            if (index < 0) {
                values.addElement(symbol);
            }
        }

        int []array = new int[values.size()];
        for (int i=0; i< array.length; i++) {
            String symbol = (String)values.elementAt(i);
            array[i] = domains[attribute].indexOf(symbol);
        }
        values = null;
        return array;
    }

    public Vector getSubset(Vector data, int attribute, int value) {
        Vector subset = new Vector();

        int num = data.size();
        for (int i=0; i< num; i++) {
            DataPoint point = (DataPoint)data.elementAt(i);
            if (point.attributes[attribute] == value) subset.addElement(point);
        }
        return subset;
```

```
}

public double calculateEntropy(Vector data) {

    int numdata = data.size();
    if (numdata == 0) return 0;

    int attribute = atributoClase;
    int numvalues = domains[attribute].size();
    double sum = 0;
    for (int i=0; i< numvalues; i++) {
        int count=0;
        for (int j=0; j< numdata; j++) {
            DataPoint point = (DataPoint)data.elementAt(j);
            if (point.attributes[attribute] == i) count++;
        }
        double probability = 1.*count/numdata;
        if (count > 0) sum += -probability*Math.log(probability);
    }
    return sum;

}

public boolean alreadyUsedToDecompose(TreeNode node, int attribute) {
    if (node.children != null) {
        if (node.decompositionAttribute == attribute )
            return true;
```

```
        }
        if (node.parent == null) return false;
        return alreadyUsedToDecompose(node.parent, attribute);
}
public void decomposeNode(TreeNode node) {

        double bestEntropy=0;
        boolean selected=false;
        int selectedAttribute=0;

        int numdata = node.data.size();
        int numinputattributes = numAttributes-1;
        node.entropy = calculateEntropy(node.data);
        if (node.entropy == 0) return;



        for (int i=0; i< numinputattributes; i++) {

           if ( atributoClase == i ) {
              continue;
              }



              int numvalues = domains[i].size();
              if ( alreadyUsedToDecompose(node, i) ) continue;
              double averageentropy = 0;
              for (int j=0; j< numvalues; j++) {
                    Vector subset = getSubset(node.data, i, j);
                    if (subset.size() == 0) continue;
```

```
            double subentropy = calculateEntropy(subset);
            averageentropy += subentropy * subset.size();  // Weighted sum
        }

        averageentropy = averageentropy / numdata;   // Taking the weighted average
        if (selected == false) {
          selected = true;
          bestEntropy = averageentropy;
          selectedAttribute = i;
        } else {
          if (averageentropy < bestEntropy) {
            selected = true;
            bestEntropy = averageentropy;
            selectedAttribute = i;
          }
        }

}

if (selected == false) return;


int numvalues = domains[selectedAttribute].size();
node.decompositionAttribute = selectedAttribute;
node.children = new TreeNode [numvalues];
for (int j=0; j< numvalues; j++) {
  node.children[j] = new TreeNode();
  node.children[j].parent = node;
  node.children[j].data = getSubset(node.data, selectedAttribute, j);
```

```
            node.children[j].decompositionValue = j;
        }

        for (int j=0; j< numvalues; j++) {
            decomposeNode(node.children[j]);
        }

        node.data = null;

    }


    public void printTree(TreeNode node, String tab) {

        int outputattr = atributoClase;

        if (node.children == null) {
            int []values = getAllValues(node.data, outputattr );
            if (values.length == 1) {
                System.out.println(tab + "\t" + attributeNames[outputattr] + " = \"" +
domains[outputattr].elementAt(values[0]) + "\";");
                return;
            }
            System.out.print(tab + "\t" + attributeNames[outputattr] + " = {");
            for (int i=0; i < values.length; i++) {
                System.out.print("\"" + domains[outputattr].elementAt(values[i]) + "\" ");
                if ( i != values.length-1 ) System.out.print( " , " );
            }
            System.out.println( " };");
```

```
            return;
        }

        int numvalues = node.children.length;
        for (int i=0; i < numvalues; i++) {
          System.out.println(tab + "if( " + attributeNames[node.decompositionAttribute] + " == \"" +
                domains[node.decompositionAttribute].elementAt(i) + "\") {" );
          printTree(node.children[i], tab + "\t");
          if (i != numvalues-1) System.out.print(tab +  "} else ");
          else System.out.println(tab +  "}");
        }


}

/
public void createDecisionTree() {
        decomposeNode(root);
        printTree(root, "");
}


public static void main(String[] args) throws Exception {

        id3 me = new id3();

Scanner in = new Scanner(System.in);

System.out.print("Ruta del archivo: ");
```

```
String str = in.nextLine();

System.out.print("atrib Clase: ");
me.atributoClase = in.nextInt();

int status = me.readData( str );
if (status <= 0) {
   return;
}

      me.createDecisionTree();
}
```

6.0 Conclusion

In this project report, I have explained the system which is used to predict the result of students whether pass or fail in school by taking into consideration various factors that affects student's performance.

References:

1. Machine Learning:
   Machine Learning: Neural and Statistical Classification by Donald Michie, David Spiegelhalter, Charles Taylor
   http://en.wikipedia.org/wiki/Machine_learning

2. JAVA:
   Java : The Complete Reference 7th Edition by Herbert Schildt
   http://en.wikipedia.org/wiki/Java_(programming_language)

3. MySQL:
   MySQL 5.1 Plugin Development by Sergei Golubchik
   SQL : The Complete Reference 2nd Edition by Paul Weinberg, James Groff

4. C.J.C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, G. Hullender, Learning to Rank using Gradient.

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

6. E.N. Ogor, "Student Academic Performance Monitoring And Evaluation Using Data Mining Techniques".

7. C.Romero, S.Ventura, P.G. Espejo, and C. Hervas, "Data Mining Algorithms to Classify Students".