

PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology

(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and
Engineering

By

SAI BIGHNESH PRUSTY S

(Reg. No. 38110479)

And

ROHIT VARMA A

(Reg. No. 38110464)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMIL NADU**

April 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabamauniversity.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **SAI BIGHNESH PRUSTY S (38110479)** and **ROHIT VARMA A(38110464)** who carried out the project entitled “**Student’s Performance Classification on Virtual Learning Environment Using Data Mining Techniques**” under my supervision from November 2021 to April 2022.

Internal Guide

Mrs. D. Deepa

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **SAI BIGHNESH PRUSTY (REG.NO. 38110479)** and **ROHIT VARMA A(REG. NO. 38110464)** hereby declare that the Project Report entitled **Student's Performance Classification on Virtual Learning Environment Using Data Mining Techniques** done by me under the guidance of **Mrs. D. Deepa** (Internal) at **Sathyabama Institute of Science and Technology, CHENNAI** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

S.Sai Bighnesh Prusty

DATE: 08.03.2022

S. Sai Bighnesh Prusty

A.Rohit Varma

A. Rohit Varma

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D., Dean**, School of Computing , **Dr.S.Vigneshwari M.E., Ph.D.**, and **Dr.L.Lakshmanan M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mrs. D. Deepa** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

In many organizations, data mining techniques are used for analyzing large amount of available data's, information's for decision making process. In educational sector, Data mining is used for wide variety of application's such as performance of the students like mark, attendance, staff opinion, extracurricular activities, Ragging and stress. The data mining techniques used for identifying the performance of the student using Naïve Bayes, K-means, KNN and Lasso algorithms. These two algorithm's identifies and analyses the performance of the student.

Table Of Content

No.	Title	Page No.
	Abstract	5
	Table Of Content	6
1.	Introduction	7-8
1.1	Objective	8
2	LITERATURE REVIEW	9-15
2.1	Existing Sytem	14
2.2	Proposed System	15
3	AIM AND SCOPE OF THE PRESENT INVESTIGATION	16
4	Experimental or materials and methods algorithms used	17-48
5	RESULTS AND DISCUSSION	49-51
5.1	Screenshots	49
6	CONCLUSION	50
7	References	51-52
8	Source Code	53

1. INTRODUCTION

There are many tremendous improvement research interests in using data mining in educational sector. This modern emerging sector, called educational data mining, concerned with improved methods that extract knowledge from data come from the educational sector. Data mining is a technique of sorting which is actually used to extract hidden patterns from huge databases. This concepts and methods can be applied in various fields like marketing, medicine, real estate, customer relationship management, engineering, web mining, etc. Educational data mining is a new emerging or advanced technique of data mining that can be applied on the data related to the field of education. The data can be collected from past used data and operational data reside in the databases of educational institutes. The data of students can be personal information or academic performance. Furthermore it can be achieve from e-learning database systems which have a huge amount of data and information used by most institutes . It uses many techniques for proper implementation of data mining concepts such as K-means Clustering, K-Nearest neighbour and Lasso algorithm. Using these techniques different kinds of knowledge can be discovered using association rules, classification and clustering. By using this we extract knowledge that describes students performance in examination and all their detail information. From This huge amounts of data, the first task is to sort them out, cluster analysis is to classify the raw data in a reasonable way. Clustering is a bunch of physical or abstract objects, as per the degree of similarity between them, divided into several groups, and makes the same data objects within a groups of high similarity and different groups of data objects which are not similar.

1.1 OBJECTIVE

The main aim of this project is to improvise the student performance in studies based on some important factors. Education is an essential element for the betterment and progress of a country. It enables the people of a country civilized and well mannered. Now-a-days developing new methods to discover knowledge from educational database in order to analyse student's trends and behaviours towards education. To analyse the data from different dimensions categorize it and to summarize the relationships. It motivated us to work on student dataset analysis. The data collection, categorization and classification is being performed manually.

2. LITERATURE REVIEW

Performance prediction of students using distributed data mining

The performance of students in higher education in India is a turning point in the academics for all students for their brightest career. In today's generation the amount of data stored in educational database increasing at a great rate. These databases contain secret information for improvement of students' performance; these data can be located at different nodes in distributed system. Classification and prediction are among the major techniques in Data mining and widely used in various fields. In this paper classification techniques are used for prediction of student performance in distributed environment. Data mining methods are often implemented at many advance universities today for analyzing available data and extracting information and knowledge to support decisionmaking. While it is important to have models at local level, their results makes it difficult to extract knowledge that can be useful at the global level. Therefore, to support decision making at this area, it is important to generalize the information contained in those models, specific classifier method can be used to generalize these rules for global model.

Predicting Student Academic Performance

Engineering schools worldwide have a relatively high attrition rate. Typically, about 35% of the first-year students in various engineering programs do not make it to the second year. Of the remaining students, quite often they drop out or fail in their second or third year of studies. The purpose of this investigation is to identify the factors that serve as good indicators of whether a student will drop out or fail the program. In order to establish early warning indicators, principal component analysis is used to analyze, in the first instance, first-year engineering student academic records. These performance predictors, if identified, can then be used effectively to formulate corrective action plans to improve the attrition rate.

Data Mining Approach For Predicting Student Performance

This work proposes a novel approach - personalized forecasting - to take into account the sequential effect in predicting student performance (PSP). Instead of using all historical data as other methods in PSP, the proposed methods only use the information of the individual students for forecasting his/her own performance. Moreover, these methods also encode the "student effect" (e.g. how good/clever a student is, in performing the tasks) and "task effect" (e.g. how difficult/easy the task is) into the models. Experimental results show that the proposed methods perform nicely and much faster than the other state-of-the-art methods in PSP.

A novel approach for upgrading Indian education by using data mining techniques

Education is the backbone of all developing countries. Upgrading of the education system, upgrades the country to the world top ranking level. One of the major problems that the education system facing is predicting the behaviour of students from large database. This paper focus on upgrading Indian education system by using one of the techniques in Data mining namely clustering. Cluster analysis solves the given data into some meaningful groups. Normally the performances of the students can be classified into different patterns as normal, average and below average. In this paper we attempt to analyze student's data in different angle beyond the above indicated patterns through newly proposed UCAM (Unique clustering with Affinity Measures) clustering algorithm.

A Review on Data Mining techniques and factors used in Educational Data Mining to predict student amelioration

Educational Data Mining (EDM) is an interdisciplinary ingenuous research area that handles the development of methods to explore data arising in a scholastic fields. Computational approaches used by EDM is to examine scholastic data in order to study educational questions. As a result, it provides intrinsic knowledge of teaching and learning process for effective education planning. This paper conducts a comprehensive study on the recent and relevant studies put through in this field to date. The study focuses on methods of analysing educational data to develop models for improving academic performances and improving

institutional effectiveness. This paper accumulates and relegates literature, identifies consequential work and mediates it to computing educators and professional bodies. We identify research that gives well-fortified advise to amend edifying and invigorate the more impuissant segment students in the institution. The results of these studies give insight into techniques for ameliorating pedagogical process, presaging student performance, compare the precision of data mining algorithms, and demonstrate the maturity of open source implements.

Data Mining: A prediction of performer or underperformer using classification

Now a day's students have a large set of data having precious information hidden. Data mining technique can help to find this hidden information. In this paper, data mining techniques name Byes classification method is used on these data to help an institution. Institutions can find those students who are consistently perform well. This study will help to institution reduce the drop put ratio to a significant level and improve the performance level of the institution.

Mining students data to analyze e-Learning behavior: A Case Study

Educational data mining concerns with developing methods for discovering knowledge from data that come from educational environment. In this paper we used educational data mining to analyze learning behavior. In our case study, we collected students' data from DataBase course. After preprocessing the data, we applied data mining techniques to discover association, classification, clustering and outlier detection rules. In each of these four tasks, we extracted knowledge that describes students' behavior.

Scholastic achievement of higher secondary students in science stream

The present study was conducted on 400 students (200 boys and 200 Girls) selected from senior secondary school of A.M.U., Aligarh-India, to establish the prognostic value of different measures of cognition, personality and demographic variables for success at higher secondary level in science stream. The scores obtained on different variables were factor-analyzed to get a smaller number of meaningful variables or factors to establish the predictive validity of these predictors. Factors responsible for success in science stream were identified. The prognostic value of the predictors was compared for high achievers and low achievers in order to identify the factors which differentiate them.

Mining student data using decision trees

Student performance in university courses is of great concern to the higher education managements where several factors may affect the performance. This paper is an attempt to use the data mining processes, particularly classification, to help in enhancing the quality of the higher educational system by evaluating student data to study the main attributes that may affect the student performance in courses. For this purpose, the CRISP framework for data mining is used for mining student related academic data. The classification rule generation process is based on the decision tree as a classification method where the generated rules are studied and evaluated. A system that facilitates the use of the generated rules is built which allows students to predict the final grade in a course under study.

An Empirical Study of the Applications of Data Mining Techniques in Higher Education

Few years ago, the information flow in education field was relatively simple and the application of technology was limited. However, as we progress into a more integrated world where technology has become an integral part of the business processes, the process of transfer of information has become more complicated. Today, one of the biggest challenges that educational institutions face is the explosive growth of educational data and to use this data to improve the quality of managerial decisions. Data mining techniques are analytical tools that can be used to extract meaningful knowledge from large data sets. This paper addresses the applications of data mining in educational institution to extract useful information from the huge data sets and providing analytical tool to view and use this information for decision making processes by taking real life examples.

2.1 EXISTING SYSTEM

- As of now, Existing system take only performance into consideration which is not sufficient for having system, which can help us to evaluate performance of a student.
- We are not having a system which would help us to integrate the performance and undesirable into consideration.

2.1.1 DISADVANTAGE OF THE EXISTING SYSTEM

- Existing system miss the undesirable data for the students.
- And It may not check the social data for the student.

2.2. PROPOSED SYSTEM

- The work aims to develop a trust model using data mining techniques, which mines required information, so that the present education system may adopt this as a strategic management tool.
- The proposed system use educational data mining techniques to evaluate performance and identify undesirable behavior.
- In educational sector, Data mining is used for wide variety of application's such as performance of the students like mark, attendance, staff opinion, extracurricular activities, Ragging and stress. The data mining techniques used for identifying the performance of the student using K-means, KNN and Lasso algorithms.

2.2.1. ADVANTAGES OF PROPOSED SYSTEM

- Educational database contain the useful information for Evaluating Students.
- The machine learning techniques are more helpful in classifying educational database and help us in evaluating the performance and undesirable behavior of a student

3. AIM AND SCOPE OF THE PRESENT INVESTIGATION

3.1 Aim

Our Aim from the project is to make a ML model which takes student data trains itself using various data mining techniques and Algorithms(KNN and Lasso) and predict the student performance in virtual environment by considering the overall factors that contribute in his overall performance.

Secondly, to learn the required techstacks and use it to make model with an java application.

And lastly to execute it get output about student performance.

3.2 Scope:

This Project can be used to get the student performance with more accuracy than any other model published earlier and we can also make some mobile or web application based on the model.

4. Experimental or materials and methods algorithms used

4.1 K-NEAREST NEIGHBOR(KNN) CLASSIFICATION METHOD

K-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms. The neighbors are taken from a set of objects for which the class (for k -NN classification) or the object property value (for k -NN regression) is known.

STEP 1: BEGIN

STEP 2: Input: $D = \{(x_1, c_1), \dots, (x_N, c_N)\}$

STEP 3: $x = (x_1, \dots, x_n)$ new instance to be classified

STEP 4: FOR each labelled instance (x_i, c_i) calculate $d(x_i, x)$

STEP 5: Order $d(x_i, x)$ from lowest to highest, $(i = 1 \dots N)$

STEP 6: Select the K nearest instances to x : D_x^k

STEP 7: Assign to x the most frequent class in D_x^k

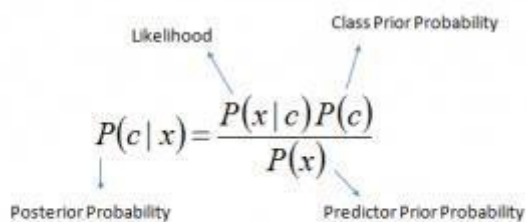
STEP 8: END

4.2 NAIVE BAYES ALGORITHM

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability $P(c|x)$ from $P(c)$, $P(x)$ and $P(x|c)$. Look at the equation below:



The diagram shows the equation $P(c|x) = \frac{P(x|c)P(c)}{P(x)}$. Arrows point from labels to the terms in the equation: 'Likelihood' points to $P(x|c)$, 'Class Prior Probability' points to $P(c)$, 'Posterior Probability' points to $P(c|x)$, and 'Predictor Prior Probability' points to $P(x)$.

$$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \dots \times P(x_n | c) \times P(c)$$

Above,

- $P(c|x)$ is the posterior probability of *class* (c , *target*) given *predictor* (x , *attributes*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

4.3 How Naive Bayes algorithm works?

Let's understand it using an example. Below I have a training data set of weather and corresponding target variable 'Play' (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather condition. Let's follow the below steps to perform it.

Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
All	5	9
	$\approx 5/14$	$\approx 9/14$
	0.36	0.64

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Problem: Players will play if weather is sunny. Is this statement is correct?

We can solve it using above discussed method of posterior probability.

$$P(\text{Yes} \mid \text{Sunny}) = P(\text{Sunny} \mid \text{Yes}) * P(\text{Yes}) / P(\text{Sunny})$$

Here we have $P(\text{Sunny} \mid \text{Yes}) = 3/9 = 0.33$, $P(\text{Sunny}) = 5/14 = 0.36$, $P(\text{Yes}) = 9/14 = 0.64$

Now, $P(\text{Yes} \mid \text{Sunny}) = 0.33 * 0.64 / 0.36 = 0.60$, which has higher probability.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

4.3.1 What are the Pros and Cons of Naive Bayes?

Pros:

- It is easy and fast to predict class of test data set. It also perform well in multi class prediction
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
- It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

Cons:

- If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as “Zero Frequency”. To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
- On the other side naive Bayes is also known as a bad estimator, so the probability outputs from `predict_proba` are not to be taken too seriously.
- Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

4.3.2 Applications of Naive Bayes Algorithms

- **Real time Prediction:** Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.
- **Multi class Prediction:** This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
- **Text classification/ Spam Filtering/ Sentiment Analysis:** Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)
- **Recommendation System:** Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

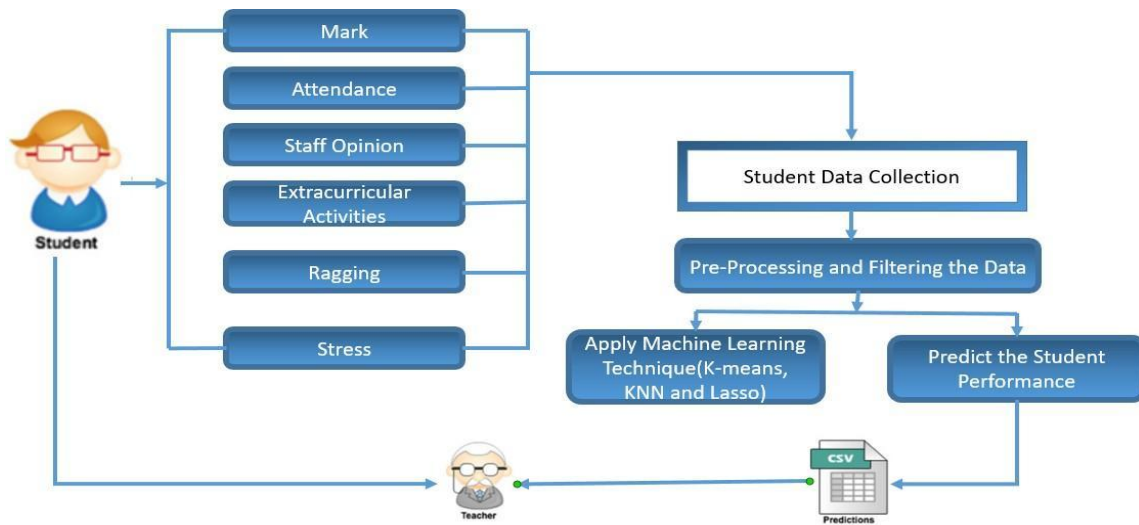
4.4 LASSO ALGORITHM

- The aim of using Lasso regression is somewhat similar to that of Ridge regression, that is to reduce overfitting. However, additionally, Lasso regression also solves the purpose of feature selection. Lasso regression makes use of L1 Regularization and is specially used when there is more number of features. L1 regularisation adds a penalty proportional to the absolute value of the coefficient's magnitude. The formula is given as:

$$\sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda |m|$$

- Here again, λ is the penalty and m denotes the magnitude of the coefficient or the slope. The value of λ should always be greater than 0. That is .

4.5 SYSTEM ARCHITECTURE



4.6 SYSTEM REQUIREMENTS

4.6.1 HARDWARE

REQUIREMENTS:

- System - Pentium-IV
- Speed - 2.4GHZ
- Hard disk - 40GB
- Monitor - 15VGA color
- RAM - 512MB

4.6.2 SOFTWARE REQUIREMENTS:

- Operating System - Windows XP
- Coding language - Java
- IDE - Net beans
- Database -MYSQL

4.7 INTRODUCTION

Java is one of the world's most important and widely used computer languages, and it has held this distinction for many years. Unlike some other computer languages whose influence has wearied with passage of time, while Java's has grown.

4.7.1 APPLICATION OF JAVA

Java is widely used in every corner of world and of human life. Java is not only used in softwares but is also widely used in designing hardware controlling software components. There are more than 930 million JRE downloads each year and 3 billion mobile phones run java.

Following are some other usage of Java:

1. Developing Desktop Applications
2. Web Applications like Linkedin.com, Snapdeal.com etc
3. Mobile Operating System like Android
4. Embedded Systems
5. Robotics and games etc.

4.7.2 FEATURES OF JAVA

The prime reason behind creation of Java was to bring portability and security feature into a computer language. Beside these two major features, there were many other features that played an important role in moulding out the final form of this outstanding language. Those features are;

1) Simple

Java is easy to learn and its syntax is quite simple, clean and easy to understand. The confusing and ambiguous concepts of C++ are either left out in Java or they have been re-implemented in a cleaner way.

Eg: Pointers and Operator Overloading are not there in java but were an important part of C++.

2) Object Oriented

In java everything is Object which has some data and behaviour. Java can be easily extended as it is based on Object Model.

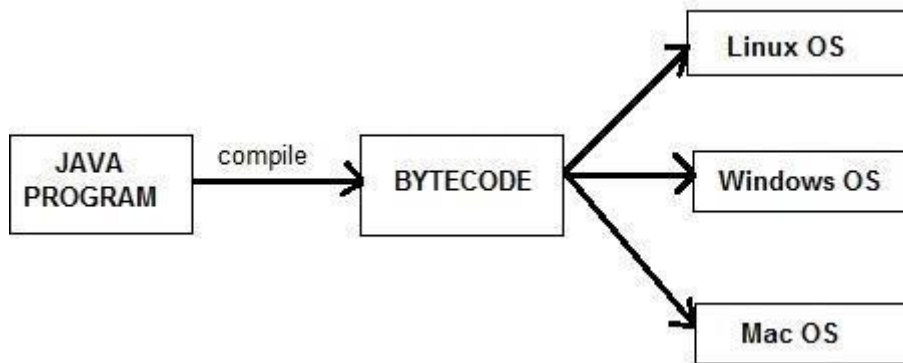
3) Robust

Java makes an effort to eliminate error prone codes by emphasizing mainly on compile time error checking and runtime checking. But the main areas which Java improved were Memory Management and mishandled Exceptions by introducing automatic Garbage Collector and Exception Handling.

4) Platform Independent

Unlike other programming languages such as C, C++ etc. which are compiled into platform specific machines. Java is guaranteed to be write-once, run-anywhere language.

On compilation Java program is compiled into byte code. This byte code is platform independent and can be run on any machine, plus this byte code format also provide security. Any machine with Java Runtime Environment can run Java Programs.



5) Secure

When it comes to security, Java is always the first choice. With java secure features it enable us to develop virus free, temper free system. Java program always runs in Java runtime environment with almost null interaction with system OS, hence it is more secure.

6) Multi-Threading

Java multithreading feature makes it possible to write program that can do many tasks simultaneously. Benefit of multithreading is that it utilizes same memory and other resources to execute multiple threads at the same time, like While typing, grammatical errors are checked along.

7) Architectural Neutral

Compiler generates byte codes, which have nothing to do with a particular computer architecture, hence a Java program is easy to interpret on any machine.

8) Portable

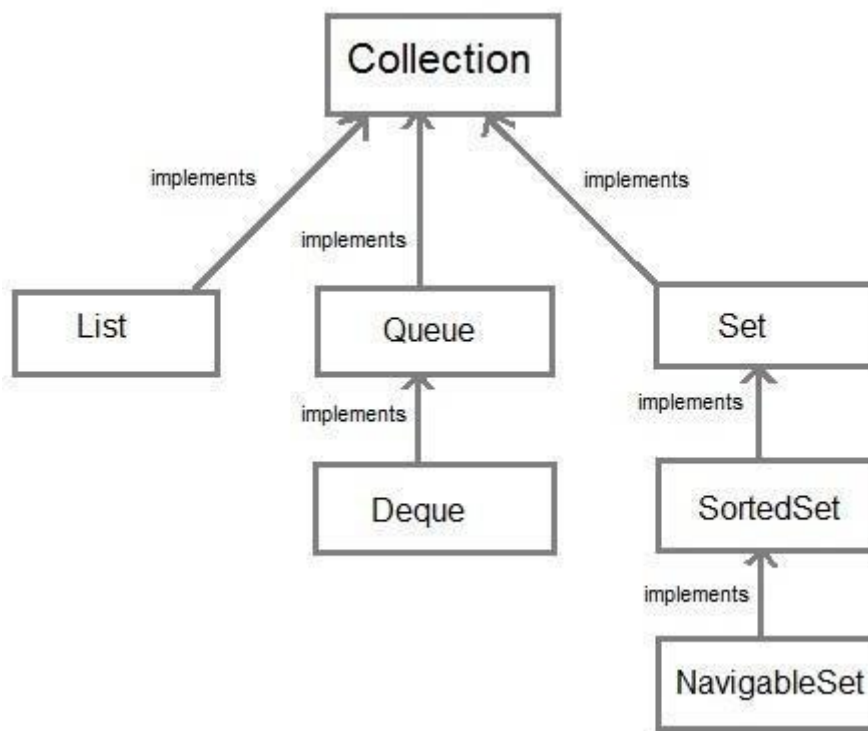
Java Byte code can be carried to any platform. No implementation dependent features. Everything related to storage is predefined, example: size of primitive data types

10) High Performance

Java is an interpreted language, so it will never be as fast as a compiled language like C or C++. But, Java enables high performance with the use of just-in-time compiler.

4.7.3 COLLECTION FRAMEWORK

Collection framework was not part of original Java release. Collections was added to J2SE 1.2. Prior to Java 2, Java provided adhoc classes such as Dictionary, Vector, Stack and Properties to store and manipulate groups of objects. Collection framework provides many important classes and interfaces to collect and organize group of alike objects.



4.7.4 MYSQL

MySQL, officially, but also called "My Sequel" is the world's most widely used open-source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases, though SQLite probably has more total embedded deployments. The SQL phrase stands for Structured Query Language.

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL 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 (and other 'AMP' stacks).

LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Freesoftware-open source projects that require a full-featured database management system often use MySQL.

For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, Drupal and other software. MySQL is also used in many high-profile, large-scale websites, including Wikipedia, Google (though not for searches), Facebook, Twitter, Flickr and YouTube.

4.8 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution.

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components.

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must justified. Thus the developed system as well within the budget and this

was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

4.9 REQUIREMENT ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analysing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

4.9.1 FUNCTIONAL REQUIREMENTS

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

Robustness

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

Security

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

Reliability

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time Between Failures). The requirement is needed in order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and survive and even capable of working around any failure.

Compatibility

It is supported by version above all web browsers. Using any web servers like localhost makes the system real-time experience.

Flexibility

The flexibility of the project is provided in such a way that is has the ability to run on different environments being executed by different users.

Safety

Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one's personal information.

4.9.2 NON- FUNCTIONAL REQUIREMENTS

Portability

It is the usability of the same software in different environments. The project can be run in any operating system.

Performance

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

Accuracy

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

Maintainability

Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyse, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

4.10 MODULES

- Data Collection
- Preprocessing
- Classification module
- Prediction

DATA COLLECTION

In this module, Student data's will be collected from the college. Student's data like mark, attendance, staff opinion, Social media, extracurricular activities, Ragging and stress.

PREPROCESSING

Data pre-processing is done to remove the incomplete noisy and inconsistent data. Data must be pre-processed before using in feature selection task.

CLASSIFICATION MODULE

The data mining techniques used for identifying the performance of the student using K-means, KNN and Lasso algorithms. These two algorithm's identifies and analyses the performance of the student.

PREDICTION

In this module, to predict the student performance based upon student mark, attendance, staff opinion, Social Media, extracurricular activities, Ragging and stress.

4.11 UML DIAGRAMS

UML is simply another graphical representation of a common semantic model. UML provides a comprehensive notation for the full lifecycle of object-oriented development.

ADVANTAGES

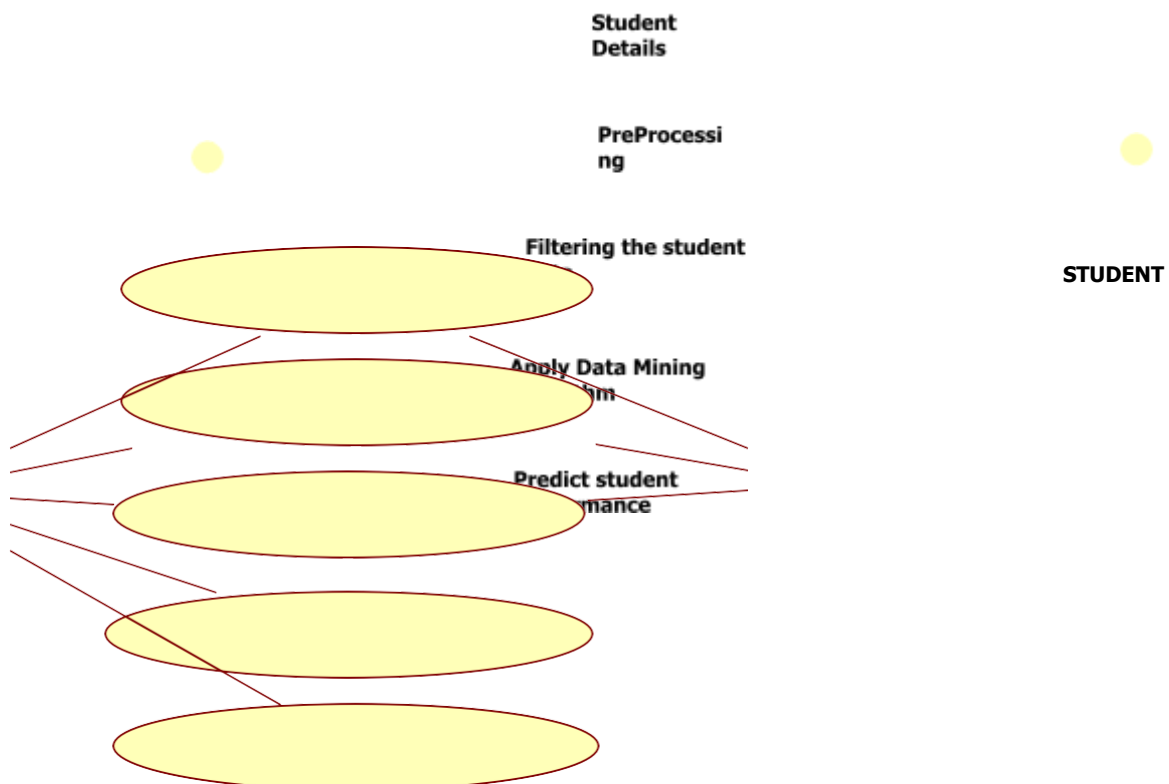
- To represent complete systems (instead of only the software portion) using object oriented concepts
- To establish an explicit coupling between concepts and executable code
- To take into account the scaling factors that are inherent to complex and critical systems
- To creating a modeling language usable by both humans and machines

UML defines several models for representing systems

- The class model captures the static structure
- The state model expresses the dynamic behavior of objects
- The use case model describes the requirements of the user
- The interaction model represents the scenarios and messages flows
- The implementation model shows the work units
- The deployment model provides details that pertain to process allocation

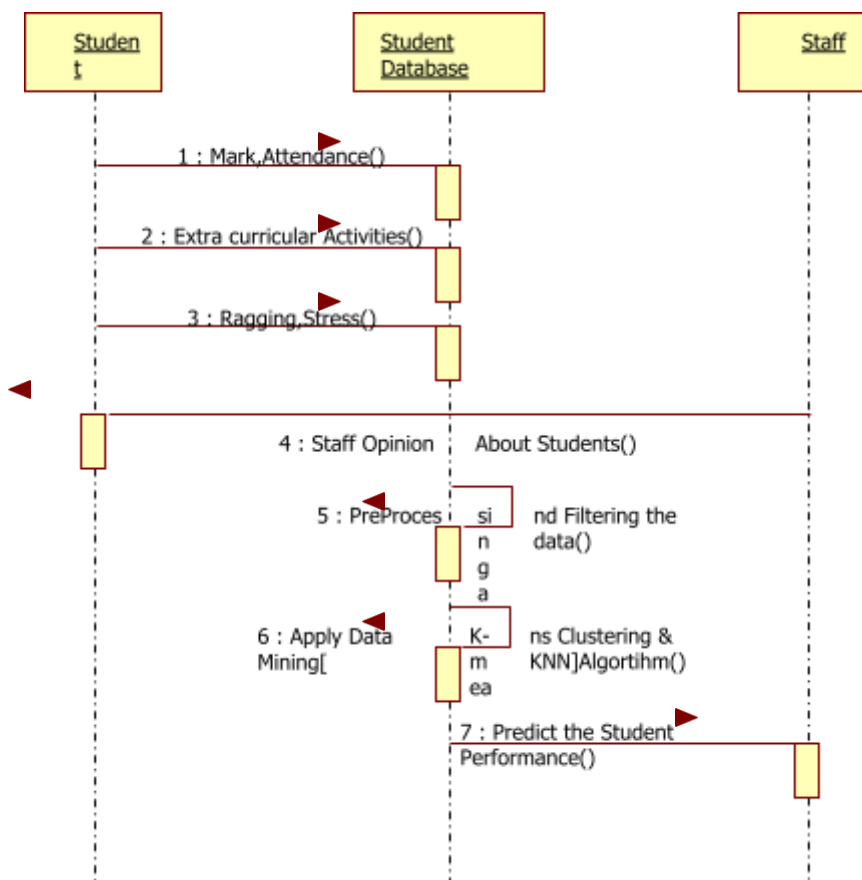
4.12.1 USECASE DIAGRAM

Use case diagrams overview the usage requirement for system. They are useful for presentations to management and/or project stakeholders, but for actual development you will find that use cases provide significantly more value because they describe “the meant” of the actual requirements. A use case describes a sequence of action that provides something of measurable value to an action and is drawn as a horizontal ellipse.



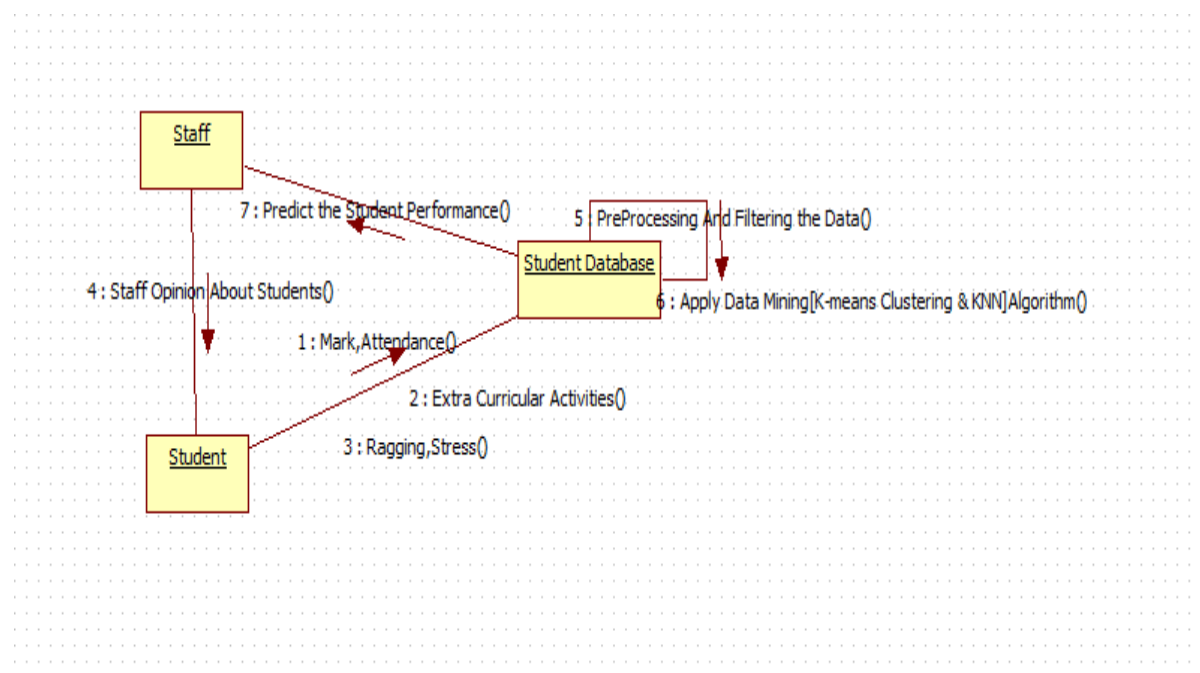
4.12.2 SEQUENCE DIAGRAM

Sequence diagram model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and commonly used for both analysis and design purpose. Sequence diagram are the most popular UML artifact for dynamic modeling, which focuses on identifying the behavior within your system.



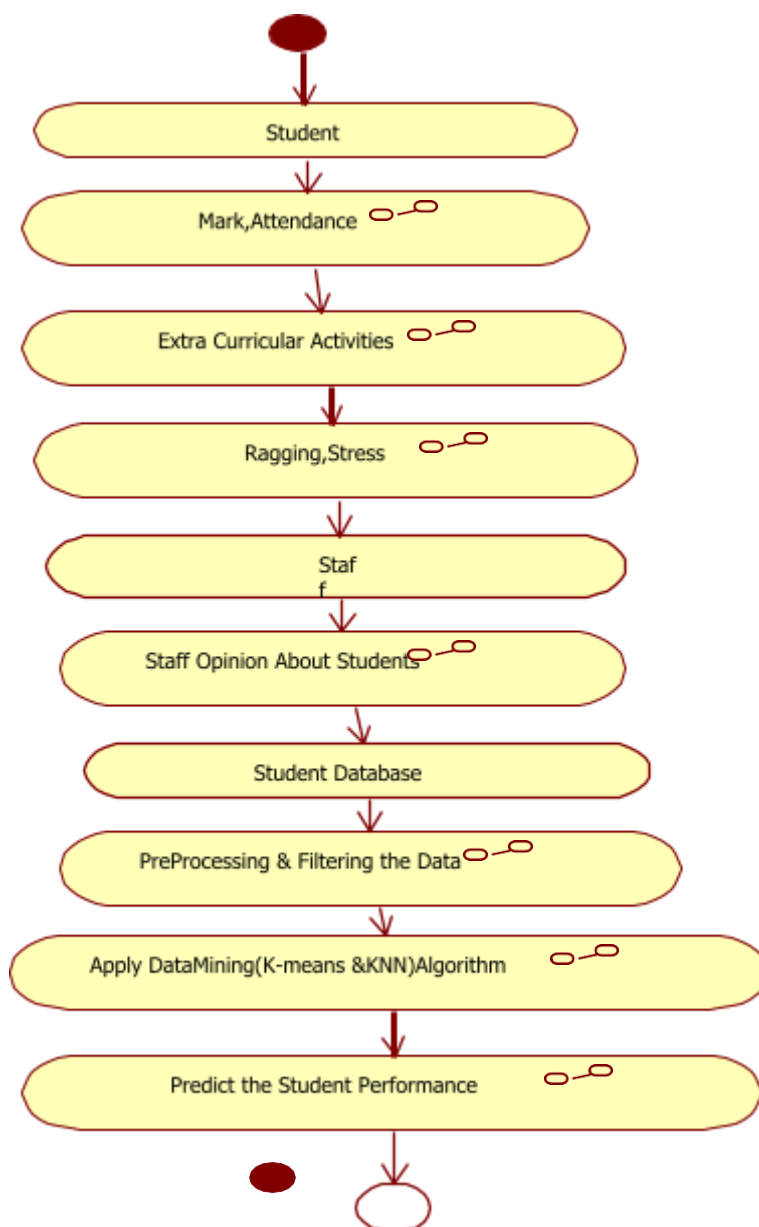
4.12.3 COLLABORATION DIAGRAM

Another type of interaction diagram is the collaboration diagram. A collaboration diagram represents a collaboration, which is a set of objects related in a particular context, and interaction, which is a set of messages exchange among the objects within the collaboration to achieve a desired outcome.



4.12.4 ACTIVITY DIAGRAM

Activity diagram are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. The activity diagrams can be used to describe the business and operational step-by- step workflows of components in a system. Activity diagram consist of Initial node, activity final node and activities in between.



4.13 SYSTEM DESIGN AND TESTING PLAN

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

What data should be given as input?

How the data should be arranged or coded?

The dialog to guide the operating personnel in providing input.

Methods for preparing input validations and steps to follow when error occur.

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the

- Future.

- Signal important events, opportunities, problems, or warnings.

- Trigger an action.

- Confirm an action.

4.14 Test plan

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of a software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

Basics of software testing

There are two basics of software testing: black box testing and white box testing.

Black box Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

White box Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing. Black box testing is often used for validation and white box testing is often used for verification.

Types of testing

There are many types of testing like

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Stress Testing
- Performance Testing
- Usability Testing
- Acceptance Testing
- Regression Testing
- Beta Testing

Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

Stress Testing

Stress testing is the testing to evaluate how system behaves under unfavorable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

Performance Testing

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

Usability Testing

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

Acceptance Testing

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

Regression Testing

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box test

5. RESULTS AND DISCUSSION

We can upload the datasets for student and create the samples student database by own. And using 6 attributes for predicting student's performance. The dataset contains the attributes such as mark, attendance, staff opinion, extracurricular activities, Ragging and stress. These attributes can perform classification and clustering using tool named as Net beans for WINDOWS OS with any configuration. We can evaluate the performance of each algorithm and compare the performance based on performance graph. We have created a large amount of dataset in order to get accurate results. The proposed framework will help students to perform good or bad in exams with the help of machine learning.

5.1 Screenshots

Marks(Grade)	Attendance(Percent)	Staff remarks(Ratings)	Staff Opinion	SocialMedia	Extra Curricular Acti	Stress	Result
4.5	80	2	1	1	0	0	No
5.4	70	3	0	0	1	1	Yes
5.3	65	1	0	0	0	0	No
2	20	0	1	1	0	1	No
2.5	30	2	1	1	0	0	No
3.5	48	3	0	0	1	0	Yes
6.4	78	4	1	0	1	1	Yes
9.5	90	5	1	0	1	1	Yes
3.2	40	3	1	0	1	0	Yes
3.7	45	2	0	1	0	0	No
5.8	69	1	0	1	0	1	No
5	50	3	1	0	1	0	Yes
7.8	80	4	1	0	1	1	Yes
7.5	75	3	1	1	0	0	No
8	80	4	1	1	1	1	Yes
3.5	46	2	0	1	0	1	No
1.5	15	1	0	1	0	0	No
6.5	74	4	1	0	1	1	Yes

Application window with input fields and a process button.

Input Field	Value
Marks(Grade)	4.5
Attendance(Percentage)	80
Staff Remarks(Rating)	2
Ragging	1
Socialmedia	1
Extra Curricular Activities	0
Stress	0

Process

Result: jLabel10

probability: jLabel11

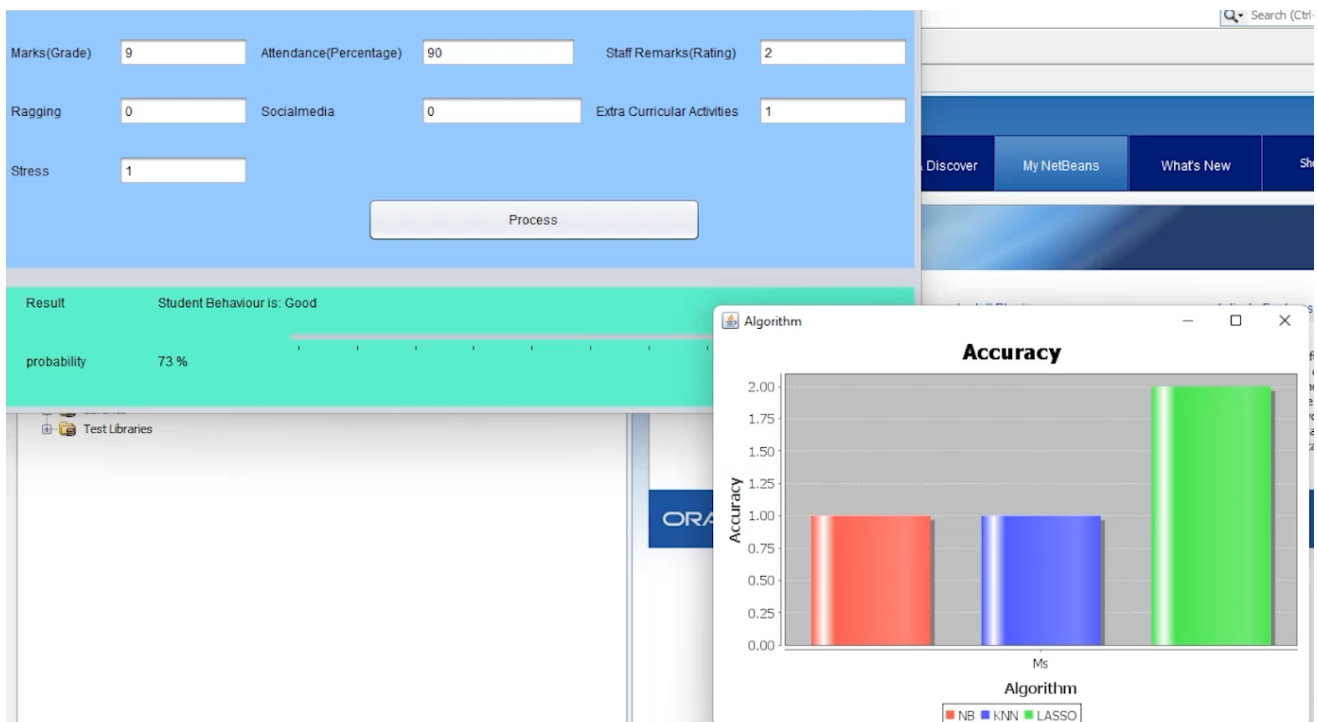
Slider scale: 1 2 3 4 5 6 7 8 9 10

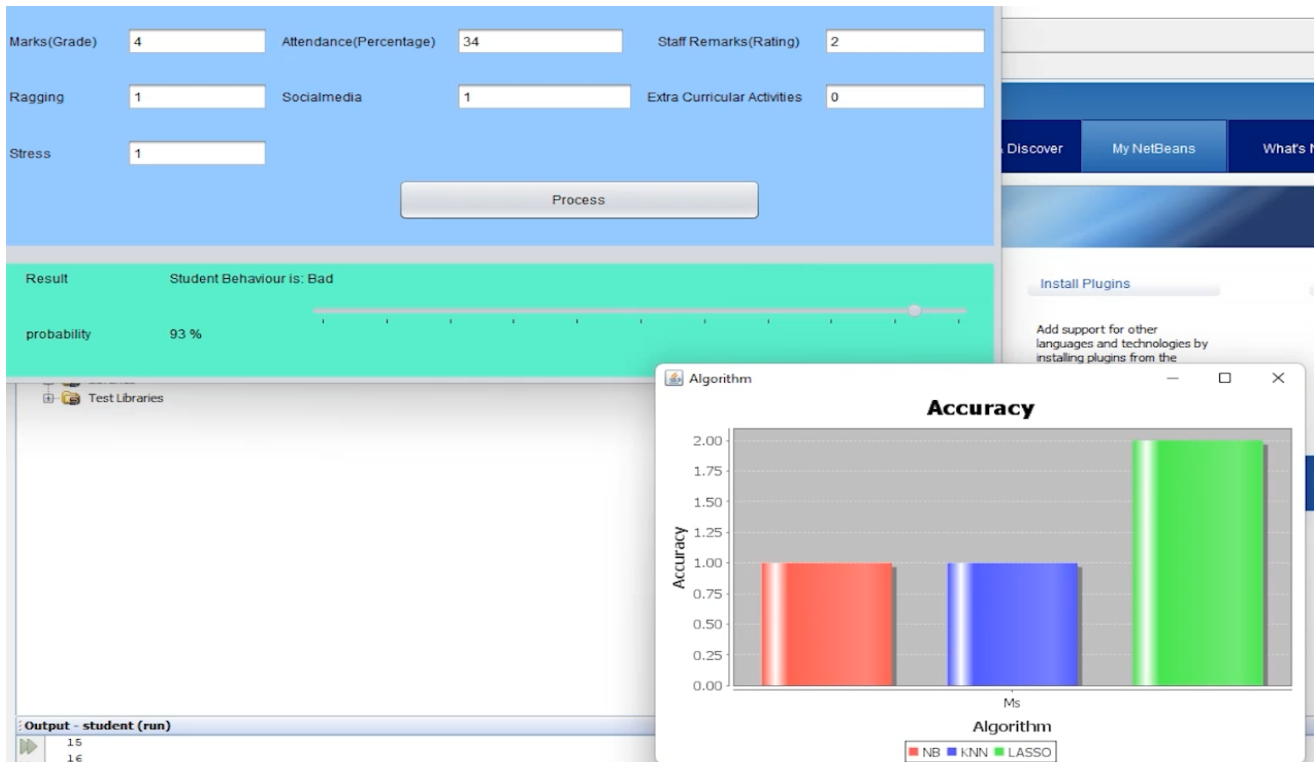
Marks(Grade)	9	Attendance(Percentage)	90	Staff Remarks(Rating)	2
Ragging		Socialmedia		Extra Curricular Activities	
Stress					

Result jLabel10

probability jLabel11

Marks - out of 10
 Attendance - out of 100
 Staff remarks - out of 5
 Ragging - 1 or 0
 Social media - 1 or 0
 Activities - 1 or 0
 Stress - 1 or 0
 where 1 is True and 0 is False





6. CONCLUSION

In this paper, the classification task is used on student database to predict the students division on the basis of previous database. As there are many approaches that are used for data classification, the Naïve Bayesian Classifier and Weighted Naïve Bayesian Classifier are used here. Information's like Attendance, Class test, Seminar and Assignment marks were collected from the student's previous database, to predict the performance at the end of the semester. This study will help to the students and the teachers to improve the division of the student. This study will also work to identify those students which needed special attention to reduce fail ration and taking appropriate action for the next semester examination. This can help the students improve in their academics, which eventually leads to a good performance in their end examinations. By this the suicide rates of students will also get reduced since

the stress is reduced. This could help in our country development by providing good and efficient engineers to the country.

7. References

- [1] M.Durairaj, C.Vijitha .Educational Data mining for Prediction of Student performance Using Clustering Algorithms. The data mining techniques are more helpful in classifying educational database Which contain the useful information for predicting a student's performance.
- [2] Kin Fun Li, David Rusk and Fred Song.Predicting Student Academic Performance,The performance predictors, if identified, can then be used effectively to formulate corrective action plans to improve the attrition rate.
- [3] E.Osmanbegovic, Mirza Suljic.Data Mining Approach For Predicting Student Performance,Using data mining the aim was to develop a model which can derive the conclusion on student's academic success.
- [4] A. Banumathi,A. Pethalakshmi. A novel approach for upgrading Indian education by using data mining techniques.

- [5] Anoopkumar M ; A. M. J. Md. Zubair Rahman, A Review on Data Mining techniques and factors used in Educational Data Mining to predict student amelioration.
- [6] U . K. Pandey, and S. Pal, “Data Mining: A prediction of performer or underperformer using classification”, (IJCSIT) International Journal of Computer Science and Information Technology, Vol. 2(2), pp.686-690, ISSN:0975-9646, 2011.
- [7] Alaa el-Halees, “Mining students data to analyze e-Learning behavior: A Case Study”, 2009..
- [8] Z. N. Khan, “Scholastic achievement of higher secondary students in science stream”, Journal of Social Sciences, Vol. 1, No. 2, pp. 84-87, 2005..
- [9] Q. A. AI-Radaideh, E. W. AI-Shawakfa, and M. I. AI-Najjar, “Mining student data using decision trees”, International Arab Conference on Information Technology(ACIT'2006), Yarmouk University, Jordan, 2006.

[10] Kumar, V. (2011). An Empirical Study of the Applications of Data Mining Techniques in Higher Education. IJACSA - International Journal of Advanced Computer Science and Applications, 2(3), 80-84. Retrieved from <http://ijacsa.thesai.org>.

8. Source Code

Homepage

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Random;
import java.util.logging.Level;
import java.util.logging.Logger;
import org.jfree.ui.RefineryUtilities;

public class HomePage extends javax.swing.JFrame {
    public static int a=0;
    private int duration2;
    public HomePage() {
        initComponents();
    }
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN: initComponents
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jTextField1 = new javax.swing.JTextField();
        jTextField2 = new javax.swing.JTextField();
        jTextField3 = new javax.swing.JTextField();
        jLabel4 = new javax.swing.JLabel();
        jTextField4 = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();
        jTextField5 = new javax.swing.JTextField();
        jLabel6 = new javax.swing.JLabel();
        jTextField6 = new javax.swing.JTextField();
        jLabel7 = new javax.swing.JLabel();
        jTextField7 = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jPanel2 = new javax.swing.JPanel();
        jLabel8 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        jLabel10 = new javax.swing.JLabel();
        jLabel11 = new javax.swing.JLabel();
        jSlider1 = new javax.swing.JSlider();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jPanel1.setBackground(new java.awt.Color(153, 204, 255));

        jLabel1.setText("Marks (Grade)");

        jLabel2.setText("Attendance (Percentage)");

        jLabel3.setText("Staff Remarks (Rating) ");
```

```

jTextField1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField1ActionPerformed(evt);
    }
});

jLabel4.setText("Ragging");

jTextField4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField4ActionPerformed(evt);
    }
});

jLabel5.setText("Socialmedia");

jLabel6.setText("Extra Curricular Activities");

jLabel7.setText("Stress");

jTextField7.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField7ActionPerformed(evt);
    }
});

jButton1.setText("Process");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addComponent(jLabel1)
        .addComponent(jLabel4)
        .addComponent(jLabel7)
        .addGap(26, 26, 26)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jTextField4)
        .addComponent(jTextField1)
        .addComponent(jTextField7)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jLabel2)
        .addComponent(jLabel5)

```

```

        .addGap(18, 18, 18)

.addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING, false)
        .addGroup(jPanell1Layout.createSequentialGroup()
            .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE, 146,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(28, 28, 28)
            .addComponent(jLabel3)
            .addGap(18, 18, 18)
            .addComponent(jTextField3,
javax.swing.GroupLayout.PREFERRED_SIZE, 141,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(jPanell1Layout.createSequentialGroup()
            .addComponent(jTextField5)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jLabel6)
            .addGap(18, 18, 18)
            .addComponent(jTextField6,
javax.swing.GroupLayout.PREFERRED_SIZE, 141,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap()
        .addGroup(jPanell1Layout.createSequentialGroup()
            .addGap(341, 341, 341)
            .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 313,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
    jPanell1Layout.setVerticalGroup(

jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanell1Layout.createSequentialGroup()
        .addGap(25, 25, 25)

.addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
        .addComponent(jLabel1)
        .addComponent(jLabel2)
        .addComponent(jLabel3)
        .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jTextField3,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(27, 27, 27)

.addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
        .addComponent(jLabel4)
        .addComponent(jTextField4,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel5)

```

```

        .addComponent(jLabel6)
        .addComponent(jTextField6,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jTextField5,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(28, 28, 28)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
        .addComponent(jLabel7)
        .addComponent(jTextField7,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE,
41, Short.MAX_VALUE)
        .addGap(25, 25, 25))
);

jPanel2.setBackground(new java.awt.Color(102, 255, 204));

jLabel8.setText("Result");

jLabel9.setText("probability");

jLabel10.setText("jLabel10");

jLabel11.setText("jLabel11");

jSlider1.setBackground(new java.awt.Color(255, 153, 153));
jSlider1.setMinorTickSpacing(10);
jSlider1.setPaintLabels(true);
jSlider1.setPaintTicks(true);
jSlider1.setEnabled(false);

javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(20, 20, 20)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
        .addComponent(jLabel9)
        .addComponent(jLabel8))
        .addGap(68, 68, 68)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
                .addComponent(jLabel10)

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGroup(jPanel2Layout.createSequentialGroup()
                .addComponent(jLabel11)

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 76,
Short.MAX_VALUE)
        .addComponent(jSlider1,
javax.swing.GroupLayout.PREFERRED_SIZE, 564,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(23, 23, 23)))
    );
    jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup())
        .addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup())

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
        .addComponent(jLabel8)
        .addComponent(jLabel10))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 39,
Short.MAX_VALUE)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
        .addComponent(jLabel9)
        .addComponent(jLabel11)))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup())
        .addGap(0, 0, Short.MAX_VALUE)
        .addComponent(jSlider1,
javax.swing.GroupLayout.PREFERRED_SIZE, 54,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(35, 35, 35))
    );

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup())
        .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
        .addComponent(jPanel2,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        .addContainerGap())
    );
    layout.setVerticalGroup(

```

```

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addContainerGap())
    );

pack();
} // </editor-fold> // GEN-END: initComponents

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_jTextField1ActionPerformed
    // TODO add your handling code here:
} // GEN-LAST:event_jTextField1ActionPerformed

private void jTextField4ActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_jTextField4ActionPerformed
    // TODO add your handling code here:
} // GEN-LAST:event_jTextField4ActionPerformed

private void jTextField7ActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_jTextField7ActionPerformed
    // TODO add your handling code here:
} // GEN-LAST:event_jTextField7ActionPerformed

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST:event_jButton1ActionPerformed
    BufferedReader reader = null;
    String trainingFile = "pima.csv";

    String testingFile = "pima1.csv";
    try {
        FileWriter fw = new FileWriter(testingFile);
        fw.append(jTextField1.getText().toString());
        fw.append(',');
        fw.append(jTextField2.getText().toString());
        fw.append(',');
        fw.append(jTextField3.getText().toString());
        fw.append(',');
        fw.append(jTextField6.getText().toString());
        fw.append(',');
        fw.append(jTextField4.getText().toString());
        fw.append(',');
        fw.append(jTextField5.getText().toString());
        fw.append(',');
        fw.append(jTextField7.getText().toString());
        fw.append('\n');
        fw.flush();
        fw.close();

    } catch (IOException ex) {
        Logger.getLogger(HomePage.class.getName()).log(Level.SEVERE, null,
ex);
    }

    try {

```

```

// TODO add your handling code here:

ArrayList<Person> training = new ArrayList<Person>();
ArrayList<Person> testing = new ArrayList<Person>();

String algorithm = "KNN";
String line;
reader = new BufferedReader(new FileReader(trainingFile));
if(testingFile.equals("FOLD")) { //used for testing 10-fold
stratification;
    int errors[] = new int[10];

    for (int i = 1; i <= 10 ; i++) { //run the test 10 times
        reader = new BufferedReader(new
FileReader(trainingFile));
        while((line = reader.readLine()) != null) {
            if(line.equals("fold" + i)) { //use fold i as testing
data
                while((line = reader.readLine()) != null) {
                    if(line.length() > 6) {
                        testing.add(new Person(line,0));
                    }
                    else {
                        break;
                    }
                }
            }
            else if(line.length() > 6) { //use other folds as
training data
                training.add(new Person(line,0));
            }
        }

        //if(algorithm.equals("NB")) {
        NaiveBayes search = new NaiveBayes(training); //initialise
Naive Bayes classifier, i.e evaluate means, and standard deviations
        for(Person p : testing) { //Run the test cases
            if(search.Test(p) != p.getOutcome()) {
                errors[i-1] += 1;
            }
        }
        //}
        //else { //i.e algorithm is K Nearest Neighbours
        //Scanner scan = new
Scanner(algorithm).useDelimiter("\\D+");
        int k = 1;
        KNearestNeighbours search1 = new
KNearestNeighbours(training, k);
        for(Person p : testing) { //Run the test cases
            if(search1.Test(p) != p.getOutcome()) {
                errors[i-1] += 1;
            }
        }
        //}

        Lasso search2 = new Lasso(training); //initialise Lasso
classifier, i.e evaluate means, and standard deviations
        for(Person p : testing) { //Run the test cases
            if(search.Test(p) != p.getOutcome()) {
                errors[i-1] += 1;
            }
        }
    }
}

```

```

        }
        //}

        training.clear();
        testing.clear();
        reader.close();
    }
    double avg = 0;
    for(int i = 0; i < 10; i++) {
        System.out.println("errors for for fold[" + i+ " ] = " +
errors[i]);
        avg += errors[i];
    }
    avg = avg/10;
    System.out.println("Average error rate is: " + avg + "/" +
76.8 ); //76.8 is average size of a fold
    System.out.println("Accuracy is: " + (1 - avg/76.8));
//accuracy of the classifier
    return;
}

    while((line = reader.readLine()) != null) { //parse the csv file
for the training examples, assuming each example has 8 attributes and an
assigned class
        System.out.println(String.valueOf(a));
        training.add(new Person(line, 0));
        a=a+1;
    }
    reader.close();
    reader = new BufferedReader(new FileReader(testingFile)); //parse
the csv file for the testing examples, assuming each example has 8 attributes
but no assigned class
    while((line = reader.readLine()) != null) {
        testing.add(new Person(line, 1));
    }
    reader.close();
    //if(algorithm.equals("NB")) {
    final long startTime = System.nanoTime();
    System.out.println("Naive Bayes classifier Result");
    NaiveBayes search = new NaiveBayes(training); //initilise Naive
Bayes classifier, i.e evaluate means, and standard deviations
    for(Person p : testing) { //Run the test cases
        if(search.Test(p) == Outcome.yes) {
            System.out.println("yes");
        }
        else {
            System.out.println("no");
        }
    }
    final long duration = System.nanoTime() - startTime;
    //}

System.out.println(".....
");

    //else { //i.e algorithm is K Nearest Neighbours
    //Scanner scan = new Scanner(algorithm).useDelimiter("\\D+");
    // System.out.println("K Nearest Neighbours classifier Result");
    // final long startTime1 = System.nanoTime();
    //int k = 2;
    // KNearestNeighbours search1 = new KNearestNeighbours(training,
k);

```

```

        // for(Person p : testing) { //Run the test cases
            //if(search1.Test(p) == Outcome.yes) {
                // System.out.println("yes");
            // }
            // else {
                // System.out.println("no");
            // }
        // }
        // final long duration1 = System.nanoTime() - startTime1;

//System.out.println(".....
..");
        // System.out.println("Accurcy is: Naive Bayes "+
(duration/(60*60))/60+"ms"); //accuracy of the classifier
        //System.out.println("Accurcy is: K Nearest Neighbours "+
(duration1/(60*60))/60+"ms");
        //}

        } catch (FileNotFoundException ex) {
            Logger.getLogger(MainJFrame.class.getName()).log(Level.SEVERE,
null, ex);
        } catch (IOException ex) {
            Logger.getLogger(MainJFrame.class.getName()).log(Level.SEVERE,
null, ex);
        } finally {
            try {
                reader.close();
            } catch (IOException ex) {

Logger.getLogger(MainJFrame.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
        try {
            // TODO add your handling code here:

            ArrayList<Person> training = new ArrayList<Person>();
            ArrayList<Person> testing = new ArrayList<Person>();

            String algorithm = "KNN";
            String line;
            reader = new BufferedReader(new FileReader(trainingFile));
            if(testingFile.equals("FOLD")) { //used for testing 10-fold
stratification;
                int errors[] = new int[10];

                for (int i = 1; i <= 10 ; i++) { //run the test 10 times
                    reader = new BufferedReader(new
FileReader(trainingFile));
                    while((line = reader.readLine()) != null) {
                        if(line.equals("fold" + i)) { //use fold i as testing
data
                            while((line = reader.readLine()) != null) {
                                if(line.length() > 6) {
                                    testing.add(new Person(line,0));
                                }
                                else {
                                    break;
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

        else if(line.length() > 6) { //use other folds as
training data
            training.add(new Person(line,0));
        }
    }

    //if(algorithm.equals("NB")) {
        NaiveBayes search = new NaiveBayes(training); //initialise
Naive Bayes classifier, i.e evaluate means, and standard deviations
        for(Person p : testing) { //Run the test cases
            if(search.Test(p) != p.getOutcome()) {
                errors[i-1] += 1;
            }
        }
        //}
        //else { //i.e algorithm is K Nearest Neighbours
        //Scanner scan = new
Scanner(algorithm).useDelimiter("\\D+");
        int k = 1;
        KNearestNeighbours search1 = new
KNearestNeighbours(training, k);
        for(Person p : testing) { //Run the test cases
            if(search1.Test(p) != p.getOutcome()) {
                errors[i-1] += 1;
            }
        }
        //}
        training.clear();
        testing.clear();
        reader.close();
    }
    double avg = 0;
    for(int i = 0; i < 10; i++) {
        System.out.println("errors for for fold[" + i + "] = " +
errors[i]);
        avg += errors[i];
    }
    avg = avg/10;
    System.out.println("Average error rate is: " + avg + "/" +
76.8 ); //76.8 is average size of a fold
    System.out.println("Accuracy is: " + (1 - avg/76.8));
//accuracy of the classifier
    return;
}

while((line = reader.readLine()) != null) { //parse the csv file
for the training examples, assuming each example has 8 attributes and an
assigned class
    training.add(new Person(line, 0));
}
reader.close();
reader = new BufferedReader(new FileReader(testingFile)); //parse
the csv file for the testing examples, assuming each example has 8 attributes
but no assigned class
    while((line = reader.readLine()) != null) {
        testing.add(new Person(line, 1));
    }
    reader.close();
    //if(algorithm.equals("NB")) {
    // final long startTime = System.nanoTime();
    // System.out.println("Naive Bayes classifier Result");

```

```

        // NaiveBayes search = new NaiveBayes(training); //initialise Naive
Bayes classifier, i.e evaluate means, and standard deviations
        // for(Person p : testing) { //Run the test cases
            // if(search.Test(p) == Outcome.yes) {
                // System.out.println("yes");
            // }
            // else {
                // System.out.println("no");
            // }
        // }
        // final long duration = System.nanoTime() - startTime;
        //}
        //
System.out.println(".....
");

        //else { //i.e algorithm is K Nearest Neighbours
        //Scanner scan = new Scanner(algorithm).useDelimiter("\\D+");
        System.out.println("K Nearest Neighbours classifier Result");
        final long startTime1 = System.nanoTime();
        int k = 1;
        KNearestNeighbours search1 = new KNearestNeighbours(training, k);
        for(Person p : testing) { //Run the test cases
            if(search1.Test(p) == Outcome.yes) {
                System.out.println("yes");
                jLabel10.setText("Student Behaviour is: Good");
            }
            else {
                System.out.println("no");

                jLabel10.setText("Student Behaviour is: Bad");

            }
        }
        Random random = new Random();
        int pr= random.nextInt(95 - 65 + 1) + 65;
        jSlider1.setValue(pr);
        jLabel11.setText(String.valueOf(pr)+" %");
        final long duration1 = System.nanoTime() - startTime1;

System.out.println(".....
");

        // System.out.println("Accurcy is: Naive Bayes "+
(duration/(60*60))/60+"ms"); //accuracy of the classifier
        // System.out.println("Accurcy is: K Nearest Neighbours "+
(duration1/(60*60))/60+"ms");
        //}

        } catch (FileNotFoundException ex) {
            Logger.getLogger(MainJFrame.class.getName()).log(Level.SEVERE,
null, ex);
        } catch (IOException ex) {
            Logger.getLogger(MainJFrame.class.getName()).log(Level.SEVERE,
null, ex);
        } finally {
            try {
                reader.close();
            } catch (IOException ex) {

Logger.getLogger(MainJFrame.class.getName()).log(Level.SEVERE, null, ex);

```

```

    }
}

try {
    // TODO add your handling code here:

    ArrayList<Person> training = new ArrayList<Person>();
    ArrayList<Person> testing = new ArrayList<Person>();

    String algorithm = "KNN";
    String line;
    reader = new BufferedReader(new FileReader(trainingFile));
    if(testingFile.equals("FOLD")) { //used for testing 10-fold
stratification;
        int errors[] = new int[10];

        for (int i = 1; i <= 10 ; i++) { //run the test 10 times
            reader = new BufferedReader(new
FileReader(trainingFile));
            while((line = reader.readLine()) != null) {
                if(line.equals("fold" + i)) { //use fold i as testing
data
                    while((line = reader.readLine()) != null) {
                        if(line.length() > 6) {
                            testing.add(new Person(line,0));
                        }
                        else {
                            break;
                        }
                    }
                }
                else if(line.length() > 6) { //use other folds as
training data
                    training.add(new Person(line,0));
                }
            }

            //if(algorithm.equals("NB")) {
            NaiveBayes search = new NaiveBayes(training); //initialise
Naive Bayes classifier, i.e evaluate means, and standard deviations
            for(Person p : testing) { //Run the test cases
                if(search.Test(p) != p.getOutcome()) {
                    errors[i-1] += 1;
                }
            }
            //}
            //else { //i.e algorithm is K Nearest Neighbours
            //Scanner scan = new
Scanner(algorithm).useDelimiter("\\D+");
            int k = 1;
            KNearestNeighbours search1 = new
KNearestNeighbours(training, k);
            for(Person p : testing) { //Run the test cases
                if(search1.Test(p) != p.getOutcome()) {
                    errors[i-1] += 1;
                }
            }
            //}
            training.clear();
            testing.clear();

```

```

        reader.close();
    }
    double avg = 0;
    for(int i = 0; i < 10; i++) {
        System.out.println("errors for for fold[" + i+ " ] = " +
errors[i]);
        avg += errors[i];
    }
    avg = avg/10;
    System.out.println("Average error rate is: " + avg + "/" +
76.8 ); //76.8 is average size of a fold
    System.out.println("Accuracy is: " + (1 - avg/76.8));
//accuracy of the classifier
    return;
}
while((line = reader.readLine()) != null) { //parse the csv file
for the training examples, assuming each example has 8 attributes and an
assigned class
    training.add(new Person(line, 0));
}
reader.close();
final long startTime = System.nanoTime();
reader = new BufferedReader(new FileReader(testingFile)); //parse
the csv file for the testing examples, assuming each example has 8 attributes
but no assigned class
    while((line = reader.readLine()) != null) {
        testing.add(new Person(line, 1));
    }
    reader.close();
    //if(algorithm.equals("NB")) {

// System.out.println("Naive Bayes classifier Result");
NaiveBayes search = new NaiveBayes(training); //initalise Naive
Bayes classifier, i.e evaluate means, and standard deviations
for(Person p : testing) { //Run the test cases
    if(search.Test(p) == Outcome.yes) {
        //System.out.println("yes");
    }
    else {
        //System.out.println("no");
    }
}
    final long duration = System.nanoTime() - startTime;
    //}
    //
System.out.println(".....
");
    //else { //i.e algorithm is K Nearest Neighbours
    //Scanner scan = new Scanner(algorithm).useDelimiter("\\D+");
// System.out.println("K Nearest Neighbours classifier Result");
final long startTime1 = System.nanoTime();
int k = 1;
KNearestNeighbours search1 = new KNearestNeighbours(training, k);
for(Person p : testing) { //Run the test cases
    if(search1.Test(p) == Outcome.yes) {
        // System.out.println("yes");
    }
    else {
        // System.out.println("no");
    }
}
}
}

```

```

        final long duration1 = System.nanoTime() - startTime1;

        System.out.println("Accuracy is: Naive Bayes "+
(duration/(60*60))/60+"ms"); //accuracy of the classifier
        System.out.println("Accuracy is: K Nearest Neighbours "+
(duration1/(60*60))/60+"ms");
        System.out.println("Accuracy is: Lasso "+
(duration2/(60*60))/60+"ms");

System.out.println(".....
");

        //}
        int x1=(int) (duration/(60*60))/60;
        int x2=(int) (duration1/(60*60))/60;
        int x3=(int) (duration2/(60*60))/60;

        BarChart_AWT chart = new BarChart_AWT("Algorithm",
"Accuracy",x1,x2+1,x3+2);
chart.pack( );
RefineryUtilities.centerFrameOnScreen( chart );
chart.setVisible( true );
        return;
    } catch (FileNotFoundException ex) {
        Logger.getLogger(MainJFrame.class.getName()).log(Level.SEVERE,
null, ex);
    } catch (IOException ex) {
        Logger.getLogger(MainJFrame.class.getName()).log(Level.SEVERE,
null, ex);
    } finally {
        try {
            reader.close();
        } catch (IOException ex) {

Logger.getLogger(MainJFrame.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
} //GEN-LAST:event_jButton1ActionPerformed

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting
code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

```

```

java.util.logging.Logger.getLogger(HomePage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(HomePage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(HomePage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(HomePage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new HomePage().setVisible(true);
    }
});
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JSlider jSlider1;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
private javax.swing.JTextField jTextField6;
private javax.swing.JTextField jTextField7;
// End of variables declaration//GEN-END:variables
}

```

Algorithm Code

```
import java.util.ArrayList;

class Lasso {
    private double attributes[];
    private double yesMean[];
    private double yesStdDev[];
    private double noMean[];
    private double noStdDev[];

    private double pYes;
    private double pNo;

    public Lasso(ArrayList<Person> training) {
        int size = training.get(0).getAttributes().length;
        yesMean = new double[size];
        noMean = new double[size];
        yesStdDev = new double[size];
        noStdDev = new double[size];
        double y = 0; //Number of yes examples in training set
        double n = 0; //Number of no examples in training set
        for(Person p : training) {
            attributes = p.getAttributes();
            if(p.getOutcome() == Outcome.yes) {
                y++;
                for(int i = 0; i < attributes.length; i++) {
                    yesMean[i] += attributes[i];
                }
            }
            else {
                n++;
                for(int i = 0; i < attributes.length; i++) {
                    noMean[i] += attributes[i];
                }
            }
        }

        for(int i = 0; i < attributes.length; i++) {
            yesMean[i] = yesMean[i] / y;
            noMean[i] = noMean[i] / n;
        }

        for(Person p : training) {
            attributes = p.getAttributes();
            if(p.getOutcome() == Outcome.yes) {
                for(int i = 0; i < attributes.length; i++) {
                    yesStdDev[i] += Math.pow(attributes[i] -
yesMean[i], 2);
                }
            }
            else {
                for(int i = 0; i < attributes.length; i++) {
                    noStdDev[i] += Math.pow(attributes[i] -
noMean[i], 2);
                }
            }
        }

        for(int i = 0; i < attributes.length; i++) {
```

```

        yesStdDev[i] = Math.sqrt((yesStdDev[i])/(y-1));
        noStdDev[i] = Math.sqrt((noStdDev[i])/(n-1));
    }
    pYes = y/(y+n); //probability of any given example being yes
    pNo = n/(n+y); //probability of any given example being no
}

public Outcome Test(Person test) {
    double pYesGivenE = 1.0;
    double pNoGivenE = 1.0;
    double attributes[] = test.getAttributes();
    for(int i = 0; i < attributes.length; i++) {
        pYesGivenE *= getProbability(attributes[i], yesMean[i],
yesStdDev[i]);
        pNoGivenE *= getProbability(attributes[i], noMean[i],
noStdDev[i]);
    }
    pYesGivenE *= pYes;
    pNoGivenE *= pNo;

    if(pNoGivenE > pYesGivenE) {
        return Outcome.no;
    }
    else {
        return Outcome.yes;
    }
}

private double getProbability(double x, double mean, double stdDev) {
//using an assumed normal distribution
    double denominator = stdDev * Math.sqrt(2*Math.PI);
    double exponent = -1 * ( (Math.pow(x - mean, 2)) /
(2*Math.pow(stdDev, 2)) );
    double probability = (1/denominator) * Math.pow(Math.E,
exponent);
    return probability;
}
}

```