

**Identifying Data Mining Techniques and Tools for Improving
Student's Academic Performance**

**Submitted in partial fulfillment of the
Requirement for the Master's degree**

**in
Information Technology**

**By
ANCY A
18/PCSA/102**

November 2019 – April 2020



**Stella Maris College (Autonomous)
17, Cathedral Road,
Chennai-600086.**

STELLA MARIS COLLEGE

**(Autonomous)
17, Cathedral Road,
Chennai-600086.**

Master of Science (Information Technology)

(Affiliated to University of Madras)



BONAFIDE CERTIFICATE

This is to certify that this is a bonafide record of the project done by

ANCY A

On

**Identifying Data Mining Techniques and Tools for Improving
Student's Academic Performance**

at

**Stella Maris College
November 2019 - April 2020**

**This project was done by her in partial fulfillment of the
requirements for the Master's degree in Information Technology**

Head of Department

Internal Guide

External Examiner

ACKNOWLEDGEMENT

First of all, I am deeply indebted to God the Almighty for being the source of my strength, guide, confidence and inspiration.

With deep gratitude I sincerely thank my institution Stella Maris College that has given me the opportunity and confidence to complete my research project in this institution.

I express my deep sense of gratitude to my principal **Dr.(Sr.) Rosy Joseph, fmm., MSc., M.Phil., Ph.D.**, Stella Maris College, for the facility provided by her in carrying out this work.

I would also like to thank my Head of the Department, **Ms. Birunda Antoinette Mary J. M.Sc., B.Ed., M.Phil.**, and faculties of Department of computer Science for their constant support throughout this project.

I would like to sincerely thank my project guide **Ms. Birunda Antoinette Mary J. M.Sc., B.Ed., M.Phil.**, for supporting and encouraging me without whose effort and guidance it wouldn't have been a successful project.

PLACE: Chennai

ANCY A

Date:

ABSTRACT

Data mining is the evolving process of identifying and extracting the hidden information from a data warehouse. Data Mining is widely used in business, medical, engineering and educational areas for analyzing existing data, identifying measures for improvement and also forecasting the future prospects. This study covers the application of data mining in education for predicting the academic performance of the Students. Educational Data Mining (EDM) plays a dominant role in the data mining era. There is an essential need to identify effective algorithms for predicting the Student's performance. The dataset used in this research is taken from the University of Minho, Mathematics department which consists of 33 attributes and 650 observations. The algorithms used in this study to predict the academic performance of the Student is Decision tree, Random Forest, Gradient Boosted tree, Logistic Regression and Multilayer Perceptron. Two Ensemble model is created, first model with trees by combining Decision tree, C5.0 and Random Forest along with cross validation, second model combining Random forest, Logistic Regression and Gradient Boosting. The tree ensemble gave more accurate result compared to other algorithms. The use of effective EDM techniques and tools would enable educators to improve the process by identifying any existing lacunae. EDM helps in developing a warning system for identifying weak Student's prior and give adequate training to improve the academic performance of the Students.

Keywords:

Student's Performance Prediction, Educational Data Mining, Data Mining Technique, Academic Performance, Decision tree, Random Forest, Gradient Boosted tree

Table of Contents

ABBREVIATION.....	III
LIST OF TABLES	IV
LIST OF FIGURES	V
1. INTRODUCTION	1
1.1 OVERVIEW.....	1
1.2 RELATED WORK	3
1.3 PROPOSED WORK.....	9
2. SYSTEM ANALYSIS	11
2.1 ABOUT R TOOL.....	11
2.2 REQUIREMENT SPECIFICATION.....	12
3. SYSTEM DESIGN	13
3.1 TAXONOMY OF SYSTEM.....	13
3.2 DATASET ATTRIBUTES	15
3.3 IMPLEMENTATION OF ALGORITHMS.....	18
3.4 REPORT.....	27
3.5 EXPERIMENTAL RESULTS	27
4. IMPLEMENTATION.....	31
4.1 FEATURE EXTRACTION	31
4.2 SAMPLE CODE	32
5. COMPARISON OF RESULTS OF THE ALGORITHMS.....	34
6. CONCLUSION.....	39
7.REFERENCES	40
8.APPENDIX.....	42

ABBREVIATION

List of Abbreviations

Abbreviations	Explanations
EDM	Educational Data Mining
MLP	Multilayer Perceptron
KDD	Knowledge Discovery in Databases
ID3	Iterative Dichotomiser 3
KNN	K-Nearest Neighbor
ANN	Artificial Neural Network
SVM	Support Vector Machine
NB	Naïve Bayes
LR	Logistic Regression
WEKA	Waikato Environment for Knowledge Analysis

LIST OF TABLES

Tab. No	Name of the Table	Page. No
Table 1	Survey of EDM Techniques used	5
Table 2	Student's Background Details	15
Table 3	Student's Social Activities.....	16
Table 4	Student's Course Work.....	16
Table 5	Important Attributes.....	17
Table 6	Individual Algorithm Accuracy	34
Table 7	Individual Algorithms Important Attributes	35
Table 8	Ensemble with Important Attributes.....	35
Table 9	Tree Ensemble	36

LIST OF FIGURES

Fig. No	Name of the Figure	Page. No
Figure 1	Data Mining Techniques	1
Figure 2	Data Mining Algorithms.....	2
Figure 3	Data Mining Techniques	13
Figure 4	Methodology of EDM	14

1. INTRODUCTION

1.1 OVERVIEW

Data Mining is a process of analyzing the important information from a large set of data and come up with a prediction model. Data Mining is also called as Knowledge Discovery in Databases (KDD). The data mining plays an important role in all the fields like medical, airline, banking sector, movies, scientific information and numerous new data types. Data mining can be used to solve real time problems. Educational Data Mining (EDM) is the emerging technique for developing a prediction model with the help of an available dataset, and transmit to predict Student's academic performance using machine learning technique. The prediction model acts like a warning system which is used to identify the weak Student's. EDM is a new field of research in data mining. The recent increase in online learning by the Students have led to the progression in development of EDM. The highly reputed educational institution mainly focuses on improving the performance of the Students in order to retain the standard rank of the institution, hence they train the Student's in such a way that they perform well in academics and extra-curricular activities. The data mining technique are classified as follows:

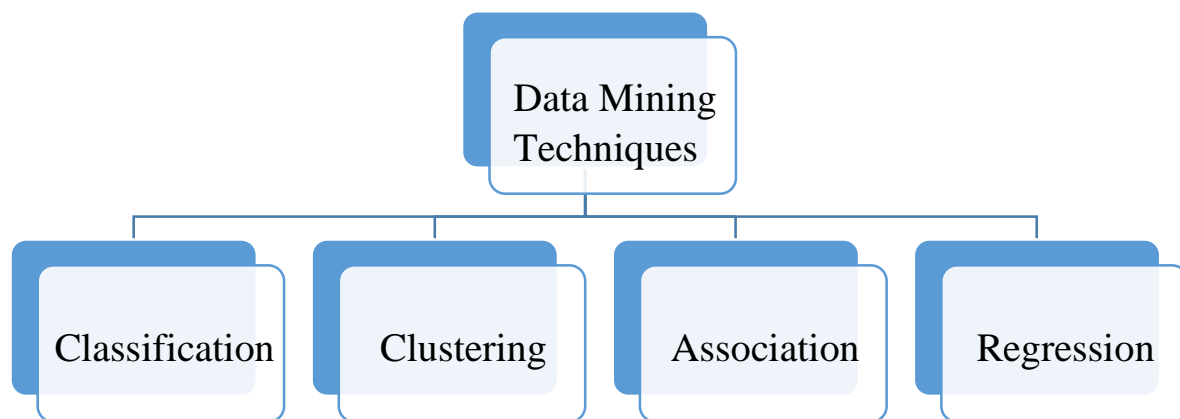


Figure 1 Data Mining Techniques

By using Educational data mining technique, the educational institutions can predict the performance of the Student and identify low performing Student's early enough to overcome their

difficulties in learning and improve their learning outcomes. Day by day the volume of the data is increasing, hence there are different data mining algorithms which are used for predicting the performance of the Students like supervised and unsupervised techniques to get the maximum accuracy.

The supervised method is categorized into Classification or Categorization and Regression. The unsupervised method is categorized into Clustering and Association. Some of the algorithms which are popularly used in prediction are Decision tree, Multilayer Perceptron, Logistic Regression, Random forest, Gradient boosted trees, ID3 and J48. This study comprises of implementing different data mining techniques which are used in predicting the academic performance of the Students. Two ensemble model is created, one model using Random Forest, Decision Tree and C5.0. Second model using Random Forest, Logistic Regression and Gradient Boosting. Major data mining techniques which are used for predicting the Student's performance are shown below:

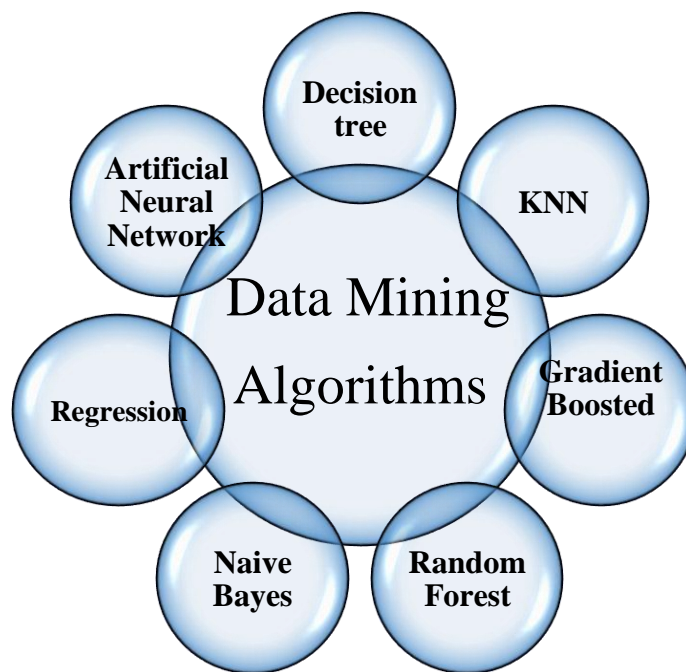


Figure 2 Data Mining Algorithms

1.2 RELATED WORK

Student's social activities and background details were used by Ching-Chieh Kiu [1] in the year 2018 for predicting the Student's academic performance. This study used several data mining technique has been applied on predicting the accuracy of each dataset, of which decision tree J48 has given the best accuracy rate. The decision tree model has achieved 95% accuracy rate compared to other data mining technique. The dataset was divided into three subsets, Student's background details, Student's social activities and Student's course work. The algorithms were implemented in WEKA tool. A comparative study on decision tree and random forest was done by Fergie Joanda and Reymon in the year 2018. The dataset was collected using questionnaires' from the computer science Student's. It had 249 instances out of which 3 different classes of data are used. The study proves that decision tree gives more accurate result of 66.9% compared to random forest which gave 61.44%. The comparison was implemented using WEKA tool [2].

A comparative study was done on Naïve Bayes, Decision Tree, K-Nearest Neighbor and Discriminant Analysis by Samuel, Nor Bahiah and Siti Mariyam in the year 2019. The study was done to identify the best data mining technique for predicting the Student's academic performance. It used 10 datasets from the University of California Irvine Repository. The decision tree out performed with the accuracy of 81.94% compared to other data mining techniques. The accuracy of other techniques was Naïve Bayes-73.61 %, KNN-80.56 % and Discriminant Analysis -77.78% accuracy rate. The tool used in the study was WEKA. In future hybrid metaheuristics algorithms will be for feature selection on the Student's data [3].

According to the study of Nongnuch Ketui, Warawut Wisomka and Kanitha Homjun in year 2019, Gradient boosted trees has given the best accuracy compared to other classification data mining techniques like Decision Tree, Weighted Decision Tree, Iterative Dichotomiser 3 (ID3) and Random Tree. WEKA is the data mining tool which is used for implementing the data mining techniques. A raw dataset was collected from the Rajamangala University of Technology Lanna Nan. The gradient boosted tree and decision tree gave good accuracy rate of 92.31% and 91.03% compared to other techniques. The Weighted Decision Tree 84.14%, ID3-89.66% and Random Tree-84.14% accuracy rate. The classification technique is widely used in predicting the Student's performance [4].

According to Romero, Cristóbal, et al [5] SMO gives more accurate result compared to other techniques like BayesNet, Naïve Bayes Simple and EM. The paper used four different datasets and each dataset produced its own accuracy. The algorithms were implemented using WEKA tool. The SMO produced 82.4% accuracy result and BayesNet produced accuracy result of 81.5% accuracy. The Naïve Bayes Simple produced an accuracy of 82.4% and EM has 80.7% accuracy rate. All the algorithms performed equally good. In the year 2014 [6], Hu, Ya-Han, Chia-Lun Lo, and Sheng-Pao Shih made a study on predict Student's' online learning using C4.5, CART and LGR. The WEKA tool was used for implementation. The dataset used was learning portfolio data and C4.5 produced more accurate result of 93.4%. Yu, Liang-Chih, et al [7] used Sentiment analysis in order to predict the Student's academic performance in the year 2018. The study used unstructured dataset. The WEKA tool was used for implementing the sentiment analysis which produced an 76% accuracy rate.

According to Deepika, K., and N. Sathvanaravana Support Vector Machine produces more accurate result compare to Linear Regression and Random forest. The study used Student's dataset of various academic disciplines of higher educational institutions in Kerala, India. The Linear Regression produced 89.96% accuracy and Random forest produced 89.98% and Support Vector Machine produced 91.43% accuracy result [8]. A comparative study was done in the year 2018 by Uzel and Vahide Nida on Multilayer Perceptron, Random Forest, Naïve Bayes, Decision Tree and Voting classifiers. The study used an educational dataset (xAPI) which is generated from an e-learning system includes 480 instances and 16 attributes. The Voting classifiers has highest accuracy of 80.6% [9].

The Artificial Neural Network outperformed the decision tree and Naïve Bayes. The study used dataset from Kalboard 360 e-learning system with 500 instances and 17 attributes. The Decision Tree has 71.1% accuracy, Naïve Bayes has 67.5% accuracy and ANN has highest accuracy of 78.1% [10]. According to Rawat, Keshav Singh, and I. V. Malhan a hybrid classification gives more accurate result for predicting the Student's academic performance. The study used data set of Department of Computer Science with 27 instances and 11 attributes. The Decision Tree produced 86.7% and KNN produced 87.5%, ANN produced 81.3% and NB produced 87.5%, Hybrid produced highest accuracy rate of 93.3% [11].

Table 1: Survey of EDM Techniques used

PAPER	EDM TECHNIQUE	TOOLS	DATASET	RESULT ACCURACY
[1] Kiu, Ching-Chieh. "Data Mining Analysis on Student's Academic Performance through Exploration of Student's Background and Social Activities." <i>2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA)</i> . IEEE, 2018.	1.Naïve Bayesian(NB) 2.Multilayer Perceptron 3. Decision Tree(DT) 4.J48 5. Random Forest	WEKA	Used 395 instances with 33 attributes that described performance in Mathematics subjects	DT-95% NB-76%
[2] Kaunang, Fergie Joanda, and Reymon Rotikan. "Student's Academic Performance Prediction using Data Mining." <i>2018 Third International Conference on Informatics and Computing (ICIC)</i> . IEEE, 2018.	1.Decision Tree(DT) 2. Random Forest(RF)	WEKA	Used 249 records with 3 different classes	DT -66.9% RF-61.14%
[3] Ajibade, Samuel-Soma M., Nor Bahiah Ahmad, and Siti Mariyam Shamsuddin. "An Heuristic Feature Selection Algorithm to Evaluate Academic Performance of Student" <i>2019 IEEE 10th Control and System Graduate Research Colloquium (ICSGRC)</i> . IEEE, 2019.	1.Naïve Bayes (NB) 2.Decision Tree (DT) 3. K-Nearest Neighbor (KNN) 4. Discriminant Analysis (DISC)	WEKA	Used 10 different datasets that are gotten from the University of California Irvine (UCI) Repository.	NB-73.61% DT-81.94 % KNN-80.56% DISC-77.78%

PAPER	EDM TECHNIQUE	TOOLS	DATASET	RESULT ACCURACY
<p>[4] Ketui, Nongnuch, Warawut Wisomka, and Kanitha Homjun. "Using Classification Data Mining Techniques for Student's Performance Prediction." <i>2019 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON)</i>. IEEE.</p>	<p>1.Decision Tree 2.Weighted Decision Tree(WDT) 3.Iterative Dichotomiser 3 (ID3) 4.Random Tree 5.Gradient Boosted Trees</p>	WEKA	<p>Education Division of Rajamangala University of Technology Lanna Nan (RMUTL Nan) for gave the raw dataset</p>	<p>DT-91.03% WDT-84.14% ID3-89.66% RT-84.14% GBT-92.31%</p>
<p>[5] Romero, Cristóbal, et al. "Predicting Student's final performance from participation in on-line discussion forums." <i>Computers & Education</i> 68 (2013): 458-472.</p>	<p>1.SMO 2.BayesNet 3.NaiveBayesSimple 4.EM</p>	Meerkat ED SNAPP	<p>Used four different Student's dataset</p>	<p>SMO -82.4% BNet -81.5% Naïve-82.4% EM -80.7%</p>
<p>[6] Hu, Ya-Han, Chia-Lun Lo, and Sheng-Pao Shih. "Developing early warning systems to predict Student's online learning performance." <i>Computers in Human Behavior</i> 36 (2014): 469-478.</p>	<p>1.C4.5 2. CART 3. LGR.</p>	WEKA	<p>Used learning portfolio data</p>	<p>C4.5-93.4% CART-76.9% LGR.-95%</p>

PAPER	EDM TECHNIQUE	TOOLS	DATASET	RESULT ACCURACY
[7] Yu, Liang-Chih, et al. "Improving early prediction of academic failure using sentiment analysis on self-evaluated comments." <i>Journal of Computer Assisted Learning</i> 34.4 (2018): 358-365.	1.Sentiment Analysis	WEKA	Used unstructured data	Sentiment Analysis-76%
[8] Deepika, K., and N. Sathvanaravana. "Analyze and Predicting the Student's Academic Performance Using Data Mining Tools." <i>2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)</i> . IEEE, 2018.	1.Linear Regression 2.Random forest 3.SVM	WEKA	Used Student's dataset of various academic disciplines of higher educational institutions in Kerala, India	LR-89.96% RF -89.98% SVM-91.43%
[9] Uzel, Vahide Nida, Sultan Sevgi Turgut, and Selma Ayşe Özel. "Prediction of Student's Academic Success Using Data Mining Methods." <i>2018 Innovations in Intelligent Systems and Applications Conference (ASYU)</i> . IEEE, 2018.	1.Multilayer Perceptron(MP) 2. Random Forest 3.NaïveBayes 4.Decision Tree 5.Voting classifiers(VC)	WEKA	Used an educational dataset which includes 480 instances and 16 attributes	MP -78.3 % RF-76.6% NB -67.7% DT -75.8% VC-80.6%

PAPER	EDM TECHNIQUE	TOOLS	DATASET	RESULT ACCURACY
[10] Siddiqui, Isma Farrah, and Qasim Ali Arain. "ANALYZING STUDENT'S ACADEMIC PERFORMANCE THROUGH EDUCATIONAL DATA MINING." <i>3C Tecnologia</i> (2019).	1.Decision Tree 2.Naïve Bayes 3.Artificial Neural Network	WEKA	Used dataset from Kalboard 360 e-learning system with 500 instances and 17 attributes	DT-71.1% NB -67.5% ANN -78.1%
[11] Rawat, Keshav Singh, and I. V. Malhan. "A Hybrid Classification Method Based on Machine Learning Classifiers to Predict Performance in Educational Data Mining." <i>Proceedings of 2nd International Conference on Communication, Computing and Networking</i> . Springer, Singapore, 2019.	1.Decision tree 2.KNN 3.Artificial neural network 4. Naïve Bayes 5.Hybrid	WEKA	Used data set of Department of Computer Science with 27 instances and 11 attributes	DT-86.7% KNN-87.5% ANN-81.3% NB-87.5% Hybird-93.3%

Decision tree has accuracy level from 66.9% to 95% and Random Forest has accuracy level from 61.14% to 89.98%. Naïve Bayes has accuracy level from 67.5% to 82.4% and KNN has accuracy level from 80.56% to 87.5%. ANN has accuracy level from 78.1% to 81.3% and Discriminant Analysis has accuracy level from 77.78%. ID3 has accuracy level from 89.66% Weighted Decision Tree has accuracy from 84.14%. Gradient Boosted Trees has accuracy level of 92.31% and Sentiment Analysis has accuracy level 76%. Linear Regression has accuracy level 89.96% and Support Vector Machine has accuracy level 91.43%. Multilayer perceptron has 78.3% accuracy and Voting classifier has 80.6% accuracy. CART has 76.9% accuracy and LGR has 95% accuracy.

1.3 PROPOSED WORK

In this experiment, the dataset used consists of 650 instances with 33 attributes that describes the performance in Mathematics subjects. The attributes of the dataset are divided into four subsets:

- 1) Student's background with 18 attributes
- 2) Student's social activities with 12 attributes
- 3) Student's coursework results with 2 attributes
- 4) Important values with 18 attributes

These subsets attributes will be used to predict final grade(G3). G3 is a Numeric datatype with range of 1 – 10 used to measure Student's performance on their final grade. The subset attributes will be evaluated under models: 2-level classification (Pass / Fail). Important attributes are selected using variable Importance function in order to enhance the accuracy of the algorithm. Ensembling is applied on the Important attributes. The algorithms which are used in study are listed below

DECISION TREE

It's one of the most powerful classification algorithm with tree like structure. Decision trees are a type of Supervised Machine Learning where the data is continuously split according to a certain parameter. A decision tree is a tree-shaped diagram used to determine a course of action. Each branch of the tree represents a possible decision, occurrence, or reaction. It has a root node, sub node and leaf node. The root node is the starting node of the tree followed by sub node which is used to make decisions and finally the leaf node which gives the end result of the classification.

RANDOM FOREST

Random forest is a supervised learning algorithm which is used for both classification as well as regression. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

LOGISTIC REGRESSION

Logistic regression is a statistical algorithm and it is mainly used for Binary classification problems (problems with two class values). Logistic regression is used to describe data and to explain the relationship between one dependent Binary variable and one or more Nominal, ordinal, interval or ratio-level independent variables.

GRADIENT BOOSTING

Gradient boosting is one of the most powerful techniques for building predictive models. The idea used in gradient boosting is a weak learner can be modified to become better. It uses a collection of decision tree which is built sequentially one after the other based on the result of the first tree the next tree performance is improved. Trees are constructed in a greedy manner, choosing the best split points based on purity scores like Gini or to minimize the loss

MULTILAYER PERCEPTRON

In the Multilayer perceptron, there can be more than one linear layer (combinations of neurons). If we take the simple example the three-layer network, first layer will be the input layer and last will be output layer and middle layer will be called hidden layer. The input data is provided into the input layer and take the output from the output layer. The number of the hidden layer can be increased as much as needed, to make the model more complex according to our task.

ENSEMBLE WITH TREE

Ensembling is a technique of combining two or more algorithms of similar or dissimilar types called base learners. This is done to make a more robust system which incorporates the predictions from all the base learners. Ensemble methods allows to produce better predictions compared to a single model. Popular technique used in Ensembling is Boosting and Bagging. This study uses two different Ensemble model. One ensemble model is created using Random Forest, logistic Regression and Gradient Boosting, other created using Decision tree, Random Forest and C5.0. Three types of concepts are used in Ensembling to combine the result which are listed below.

AVERAGING

It's defined as taking the average of predictions from models in case of regression problem or while predicting probabilities for the classification problem.

Model1	Model2	Model3	AveragePrediction
45	40	65	50

MAJORITY VOTE

It's defined as taking the prediction with maximum vote / recommendation from multiple models predictions while predicting the outcomes of a classification problem.

Model1	Model2	Model3	VotingPrediction
1	0	1	1

WEIGHTED AVERAGE

Different weights are applied to predictions from multiple models then taking the average which means giving high or low importance to specific model output.

	Model1	Model2	Model3	WeightAveragePrediction
Weight	0.4	0.3	0.3	
Prediction	45	40	60	48

2. SYSTEM ANALYSIS

2.1 ABOUT R TOOL

R is a language and environment for statistical computing and graphics. R is a programming language developed by Ross Ihaka and Robert Gentleman in 1993. R possesses an extensive catalog of statistical and graphical methods. It includes machine learning algorithm, linear regression, time series, statistical inference to name a few. Most of the R libraries are written in R, but for heavy computational task, C, C++ and Fortran codes are preferred. R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- An effective data handling and storage facility
- A suite of operators for calculations on arrays, in particular matrices
- A large, coherent, integrated collection of intermediate tools for data analysis

- Graphical facilities for data analysis and display either on-screen or on hardcopy
- A well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities

Many users think of R as a statistics system. We prefer to think of it as an environment within which statistical techniques are implemented. R can be extended (easily) via packages. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics. R is used for Statistical inference, Data analysis and Machine learning algorithm. Data science is shaping the way companies run their businesses.

2.2 REQUIREMENT SPECIFICATION

HARDWARE

Windows 10 /8.1/8 /7 /Vista /XP /2000 operating system with at least 256 MB of RAM

SOFTWARE

R studio is the software tool which is used for implementing the machine learning algorithms. R is an open source software which has lots of statistical packages installed. It also has different packages for implementing specific algorithms. The R version which is used in this study is version 3.6.2

DATASET

The dataset used in this study is taken from the University of Minho, Mathematics department which consists of 33 attributes and 650 observations. This dataset includes G1, G2, G3 major attributes and G3 containing the average grade of G1 and G2.

3. SYSTEM DESIGN

3.1 TAXONOMY OF SYSTEM

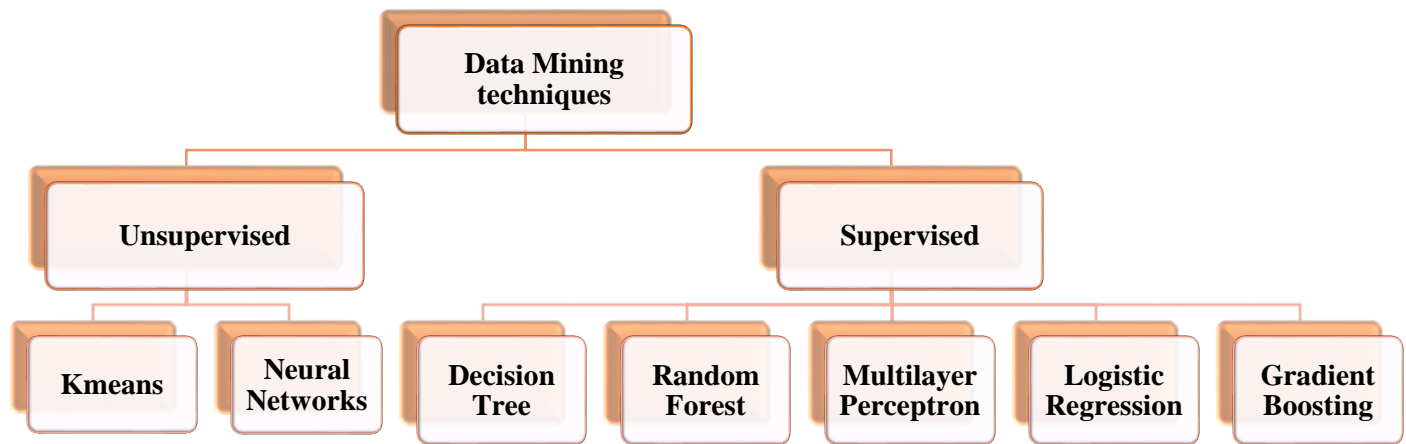


Figure 3 Data Mining Techniques

Data mining techniques comes in two forms: supervised and unsupervised techniques. The supervised technique is used when there is target variable. The target variables can have two or more possible outcomes, or even numeric values. Based on the target variables the data is trained for prediction. Unsupervised technique does not focus on predetermined attribute and finds the hidden structure and relation among the data. It's mainly used for grouping the data of similar types which doesn't have a target variable. This study covers supervised technique since the dataset has a target variable

METHODOLOGY FOR EDM

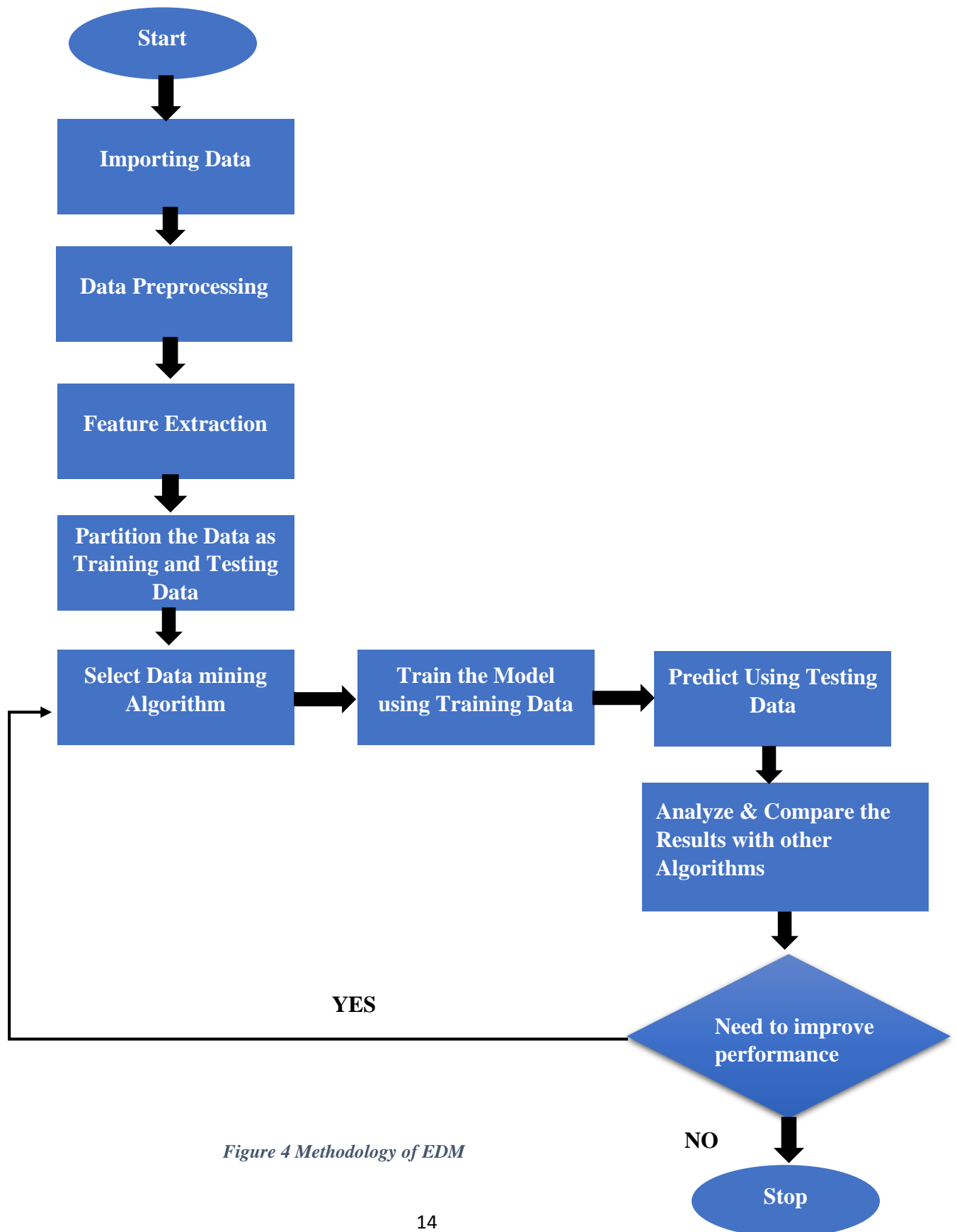


Figure 4 Methodology of EDM

3.2 DATASET ATTRIBUTES

STUDENT'S BACKGROUND			
Attribute	Description	Type	Value
sex	gender of Student's	Binary	male female
school	school of Student's		Mousinho da Silveira Gabriel Pereira
address	type of Student's home address		rural urban
Pstatus	cohabitation status of parent		living together apart
famsize	size of family		≤ 3 > 3
schoolsup	extra educational school support		yes no
famsup	educational support from family		yes no
Mjob	job of mother	Nominal	- at home - civil services - teacher - health care related - other
Fjob	job of father		
reason	reason to choose this school		- close to home - school reputation - course preference - other
guardian	guardian of Student's		- father mother other
Medu	education of mother	Numeric	0# none 1# primary education 2# 5th to 9th grade 3# secondary education 4# higher education
Fedu	education of father		
famrel	quality of family relationships		very bad (1) to excellent (5)
age	age of Student's		15 - 22
traveltime	travel time from home to school		< 15 min 15 to 30 min 30 min. to 1 hour > 1 hour
studytime	weekly study time		< 2 hours 2 to 5 hours 5 to 10 hours > 10 hours
failures	number of failures in past class		n if $1 \leq n < 3$, else 4

Table 2 Student's Background Details

STUDENT'S SOCIAL ACTIVITIES			
Attribute	Description	Type	Value
activities	extra-curricular	Binary	yes no
higher	plans for higher education		
internet	home internet access		
nursery	nursery school attended		
paidclass	extra paid classes		
romantic	in romantic relationship		
absences	absences from school	Numeric	very low (1) to very high (5)
health	status of current health		
freetime	free time after school		
goout	outing with friends		
Dalc	consume alcohol in weekday		
Walc	consume alcohol in weekend		0 - 93

Table 3 Student's Social Activities

STUDENT'S COURSEWORK RESULT			
Attribute	Description	Type	Value
GI	1st grade period	Numeric	0 - 20
G2	2nd grade period		

Table 4 Student's Course Work

IMPORTANT ATTRIBUTES			
Attribute	Description	Type	Value
failures	number of failures in past class	Numeric	n if $1 \leq n < 3$, else 4
studytime	weekly study time	Numeric	1# < 2 hours 2# 2 to 5 hours 3# 5 to 10 hours 4# > 10 hours
G2	2nd grade period	Numeric	0 – 20

Attribute	Description	Type	Value
absences	absences from school	Numeric	very low (1) to very high (5)
goout	outing with friends	Numeric	very low (1) to very high (5)
Wal-c	consume alcohol in weekend	Numeric	0 – 5
Dalc	consume alcohol in weekday	Numeric	0 – 5
traveltime	travel time from home to school	Numeric	0# < 15 min 1# 15 to 30 min 2# 30 min. to 1 hour 3# > 1 hour
famrel	quality of family relationships	Numeric	very bad (1) to excellent (5)
Fedu	education of father	Numeric	0# none 1# primary education 2# 5th to 9th grade 3# secondary education 4# higher education
reason	reason to choose this school	Nominal	- close to home - school reputation - course preference - other
guardian	guardian of Student's	Nominal	- father mother other
Fjob	job of father	Nominal	- at home - civil services - teacher - health care related - other
Mjob	job of mother	Nominal	
higher	plans for higher education	Binary	yes no
internet	home internet access	Binary	yes no
paidclass	extra paid classes	Binary	yes no

Table 5 Important Attributes

3.3 IMPLEMENTATION OF ALGORITHMS

➤ Decision tree

```
library(rpart)
library(rpart.plot)
stud <- read.table(file.choose(), header = TRUE, sep=';')
shuffle_index <- sample(1:nrow(stud))
stud <- stud[shuffle_index, ]
create_train_test <- function(data, size = 0.8, train = TRUE) {
  n_row = nrow(data)
  total_row = size * n_row
  train_sample <- 1: total_row
  if (train == TRUE) {
    return (data[train_sample, ])
  } else {
    return (data[-train_sample, ])
  }
}
data_train <- create_train_test(stud, 0.8, train = TRUE)
data_test <- create_train_test(stud, 0.8, train = FALSE)
tree<-rpart(class~sex+school+Pstatus+failures+studytime+Medu+Fedu+traveltime+
Fjob+Mjob+famrel+reason+Pstatus+famsize+schoolsup+famsup+age+guardian, data=d
ata_train, method='class')
rpart.plot(tree,extra=106,cex=.8,roundint = FALSE)
pred<-predict(tree, data_test, type = 'class')
```

The libraries which are required to implement decision tree are rpart and raprt.plot. First the data is split into training and testing data with 80% for training and 20% for testing. The decision tree model is built using training data and with the help of rpart function, the method is class since its classification model. The rpart.plot() is used for plotting decision tree and predict function is used for predicting the test data with the trained tree model. Based on the prediction, confusion matrix is derived to find out the accuracy of the trained decision tree model.

```
print(paste('Accuracy for testing', accuracy_Test))
## [1] "Accuracy for testing 0.792307692307692"
```

➤ Random Forest

```
library(randomForest)
stud <-read.table(file.choose(),header = TRUE,sep=';')
index<-sample(1:nrow(student),size=data1_set_size)

training<-student[index,]
testing<-student[-index,]

rf<-randomForest(class~sex+school+address+Pstatus+failures+studytime+Medu+Fedu+traveltime+Fjob+Mjob+famrel+reason+Pstatus+famsize+schoolsup+famsup+age+guardian, data=training, mtry=10, ntree=300, importance=TRUE)

result<-data.frame(testing$class, predict(rf, testing[,1:32], type="response"))

prediction<-predict(rf,testing,type="class")

ConfusionMatric<-table(prediction,testing$class)
ConfusionMatric
accuracy<-sum(diag(ConfusionMatric))/sum(ConfusionMatric)
```

The library which is required to implement Random forest is randomForest. First the data is split into training and testing data with 80% for training and 20% for testing. The random forest model is built using training data and with the help of randomForest function, ntree is the number of trees that needs to be built in the forest and mtry is number of attributes need to construct the tree. Predict function is used for predicting the test data with the trained model. Based on the prediction, confusion matrix is derived to find out the accuracy of the trained Random forest model.

```
print(paste("Accuracy of the test",accuracy))
## [1] "Accuracy of the test 0.769230769230769"
```

➤ Multilayer Perceptron

```
library(RSNNS)
stud <- read.csv(file.choose(), header = TRUE)
stud$school <- as.numeric(stud$school)
stud$address <- as.numeric(stud$address)
stud$sex <- as.numeric(stud$sex)
stud$famsize <- as.numeric(stud$famsize)
stud$Pstatus <- as.numeric(stud$Pstatus)
stud$Mjob <- as.numeric(stud$Mjob)
stud$Fjob <- as.numeric(stud$Fjob)
stud$reason <- as.numeric(stud$reason)
stud$guardian <- as.numeric(stud$guardian)
stud$schoolsup <- as.numeric(stud$schoolsup)
stud$famsup <- as.numeric(stud$famsup)
stud <- stud[sample(1:nrow(stud), length(1:nrow(stud))), 1:ncol(stud)]
StudentValues <- stud[, 1:18]
StudentTargets <- decodeClassLabels(stud[, 19])
stud <- splitForTrainingAndTest(StudentValues, StudentTargets, ratio=0.20)

model <- mlp(stud$inputsTrain, stud$targetsTrain, size=10, learnFuncParams=c(
0.1), maxit=50, inputsTest=stud$inputsTest, targetsTest=stud$targetsTest)
Predictions <- predict(model, stud$inputsTest)
confusionMatrix(stud$targetsTrain, fitted.values(model))
```

The library which is required to implement Multilayer Perceptron is RSNNS. First the non-numeric attributes are converted to numeric in order to perform MLP. Secondly data is split into training and testing data with 80% for training and 20% for testing. The Multilayer Perceptron model is built using training data and size is the number of hidden layers and maxit is number of iterations. Predict function is used for predicting the test data with the trained model. Based on the prediction, confusion matrix is derived to find out the accuracy of the trained MLP model.

```
print(paste('Accuracy for test', accuracy_Test))
## [1] "Accuracy for test 0.784615384615385"
```

➤ Gradient Boosting

```
library(tidyverse)
library(caret)
create_train_test <- function(data, size = 0.8, train = TRUE) {
  n_row = nrow(data)
  total_row = size * n_row
  train_sample <- 1: total_row
  if (train == TRUE) {
    return (data[train_sample, ])
  } else {
    return (data[-train_sample, ])
  }
}

data_train <- create_train_test(stud, 0.8, train = TRUE)
data_test <- create_train_test(stud, 0.8, train = FALSE)

model <- train(class~sex+school+address+Pstatus+failures+studytime+Medu+Fedu+
traveltime+Fjob+Mjob+famrel+reason+Pstatus+famsize+schoolsup+famsup+age+guard
ian, data = data_train, method ="xgbTree", trControl = trainControl("cv", num
ber=10))

model$bestTune

predicted.classes <-predict(model,data_test)
```

The library which is needed to implement Gradient Boosting is tidyverse and caret. First the data is split into training and testing data with 80% for training and 20% for testing. The Gradient Boosting model is built using training data and method is xgbTree is used along with 10fold cross validation. BestTune is used for tuning the trained model. Predict function is used for predicting the test data with the trained model. Based on the prediction, confusion matrix is derived to find out the accuracy of the trained Gradient Boosting model.

```
mean(predicted.classes == data_test$class)

## [1] 0.7923077
```

➤ Logistic Regression

```
library(caTools)
library(e1071)
set.seed(200)
pass_train<-sample(1:nrow(pass),0.7*nrow(pass))
fail_train<-sample(1:nrow(fail),0.7*nrow(fail))
train_pass<-pass[pass_train,]
train_fail<-fail[fail_train,]
train<-rbind(train_pass,train_fail)
table(train$class)
test_pass <- pass[-pass_train, ]
test_fail <- fail[-fail_train, ]
test<- rbind(test_pass, test_fail)  # row bind the pass and fail
table(test$class)

mymodel<-glm(class~sex+school+address+Pstatus+failures+studytime+Medu+Fedu+tr
aveltime+Fjob+Mjob+famrel+reason+Pstatus+famsize+schoolsup+famsup+age+guardia
n, family='binomial', data=train,maxit=100)
retest<-predict(mymodel,test,type="response")
confmatrix<-table(Actual_Value=test$class,Predicted_Value=retest>=0.5)
```

The library which is needed to implement Logistic Regression is caTools and e1071. First the data is split into training and testing data with 80% for training and 20% for testing. The Logistic Regression model is built using glm method. The model is trained using training data with maxit as 100 iterations. Logistic Regression is a binomial classification hence family is given as binomial. Predict function is used for predicting the test data with the trained model. Based on the prediction, confusion matrix is derived to find out the accuracy of the trained Logistic Regression model.

```
accuracy<-sum(diag(confmatrix))/sum(confmatrix)
print(paste("Accuracy of the test",accuracy))
## [1] "Accuracy of the test 0.871794871794872"
```

➤ Ensemble

```
#RANDOM FOREST

model_rf<-train(trainSet[,predictors],trainSet[,response],method='rf',trControl=fitControl,tuneLength=3)

testSet$pred_rf<-predict(object = model_rf,testSet[,predictors])

confusionMatrix(testSet$class,testSet$pred_rf)


#LOGSTIC REGRESSION

model_lr<-train(trainSet[,predictors],trainSet[,response],method='glm',trControl=fitControl,tuneLength=3)

testSet$pred_lr<-predict(object = model_lr,testSet[,predictors])

confusionMatrix(testSet$class,testSet$pred_lr)


#GRADIENT BOOSTING

model_gb<- train(trainSet[,predictors],trainSet[,response], method = 'gbm', trControl = fitControl,tuneLength=3)

testSet$pred_gb<-predict(object = model_gb,testSet[,predictors])

confusionMatrix(testSet$class,testSet$pred_gb)

testSet$pred_rf_prob<-predict(object = model_rf,testSet[,predictors],type='prob')

testSet$pred_gb_prob<-predict(object = model_gb,testSet[,predictors],type='prob')

testSet$pred_lr_prob<-predict(object = model_lr,testSet[,predictors],type='prob')


#Taking average of predictions

testSet$pred_avg<-(testSet$pred_rf_prob$PASS+testSet$pred_gb_prob$PASS+testSet$pred_lr_prob$PASS)/3

testSet$pred_avg<-as.factor(ifelse(testSet$pred_avg>0.5, 'PASS', 'FAIL'))

confusionMatrix(testSet$class,testSet$pred_avg)

mean(testSet$pred_avg == testSet$class)


testSet$pred_majority<-as.factor(ifelse(testSet$pred_rf=='PASS' & testSet$pred_gb=='PASS', 'PASS', ifelse(testSet$pred_rf=='PASS' & testSet$pred_lr=='PASS', 'PASS', ifelse(testSet$pred_gb=='PASS' & testSet$pred_lr=='FAIL', 'FAIL', 'FAIL'))))
```

```

confusionMatrix(testSet$class, testSet$pred_majority)
mean(testSet$pred_majority == testSet$class)

#Taking weighted average of predictions
testSet$pred_weighted_avg<-(testSet$pred_rf_prob$PASS*0.25)+(testSet$pred_gb_
prob$PASS*0.25)+(testSet$pred_lr_prob$PASS*0.5)

#Splitting into binary classes at 0.5
testSet$pred_weighted_avg<-as.factor(ifelse(testSet$pred_weighted_avg>0.5, 'PASS', 'FAIL'))
mean(testSet$pred_weighted_avg == testSet$class)

```

The libraries which are used for building the ensemble are caret, randomForest, xgboost and tidyverse. Gradient Boosting, Random Forest and logistic Regression are used for building ensemble model. The train function is used for training all the three algorithms and the method is changed based on the model that will be implemented. Each model's prediction is added as a new column in the testing data sequentially. Finally average is taken between the three models predictions and end result is produced. Majority voting is done for all the prediction columns in the test data and end result is produced. Different weights are applied to prediction columns in the test data and average is taken to produces the end result.

Accuracy of Weighted average

```
## [1] 0.9534884
```

Accuracy of Majority Voting

```
## [1] 0.9534884
```

Accuracy of Average

```
## [1] 0.9534884
```


➤ Ensemble with Tree algorithms

```
model_cd<- train(trainSet[,predictors],trainSet[,response], method="C5.0",
trControl=fitControl,tuneLength=3)

#Predicting using C5.0 model

testSet$pred_cd<-predict(object = model_gb,testSet[,predictors])
confusionMatrix(testSet$class,testSet$pred_cd)

#Predicting using Decision tree model

model_dt<-train(trainSet[,predictors],trainSet[,response],method="rpart",t
rControl=fitControl,tuneLength=3)

testSet$pred_dt<-predict(object = model_dt,testSet[,predictors])
confusionMatrix(testSet$class,testSet$pred_dt)

model_rf<-train(trainSet[,predictors],trainSet[,response],method='rf',ntre
e=450,trControl=fitControl,tuneLength=3)

#Predicting using randomforest model

testSet$pred_rf<-predict(object = model_rf,testSet[,predictors])
confusionMatrix(testSet$class,testSet$pred_rf)

testSet$pred_rf_prob<-predict(object = model_rf,testSet[,predictors],type=
'prob')

testSet$pred_gb_prob<-predict(object = model_gb,testSet[,predictors],type=
'prob')

testSet$pred_lr_prob<-predict(object = model_lr,testSet[,predictors],type=
'prob')

#Taking average of predictions

testSet$pred_avg<-(testSet$pred_rf_prob$PASS+testSet$pred_gb_prob$PASS+tes
tSet$pred_lr_prob$PASS)/3

#Splitting into binary classes at 0.5

testSet$pred_avg<-as.factor(ifelse(testSet$pred_avg>0.5, 'PASS', 'FAIL'))
confusionMatrix(testSet$class,testSet$pred_avg)

testSet$pred_majority<-as.factor(ifelse(testSet$pred_rf=='PASS' & testSet$
pred_gb=='PASS', 'PASS', ifelse(testSet$pred_rf=='PASS' & testSet$pred_lr=='
PASS', 'PASS', ifelse(testSet$pred_gb=='PASS' & testSet$pred_lr=='FAIL', 'FAI
L', 'FAIL'))))
```

```

confusionMatrix(testSet$class, testSet$pred_majority)

#Taking weighted average of predictions

testSet$pred_weighted_avg<-(testSet$pred_rf_prob$PASS*0.25)+(testSet$pred_gb_
prob$PASS*0.25)+(testSet$pred_lr_prob$PASS*0.5)

#Splitting into binary classes at 0.5

testSet$pred_weighted_avg<-as.factor(ifelse(testSet$pred_weighted_avg>0.5, 'PASS', 'FAIL'))

mean(testSet$pred_weighted_avg == testSet$class)

```

The library which is used for implementing tree Ensembling is caret which contains the train function. Decision tree, Random Forest and C5.0 are used for building the tree ensemble. The train function is used for training all the three algorithms and the method is changed based on the model that will be implemented. Each model's prediction is added as a new column in the testing data sequentially. Finally average is taken between the three models predictions and end result is produced. Majority voting is done for all the prediction columns in the test data and end result is produced. Different weights are applied to prediction columns in the test data and average is taken to produces the end result.

Accuracy of Weighted average

```
## [1] 0.9612403
```

Accuracy of Majority Voting

```
## [1] 0.9612403
```

Accuracy of Average

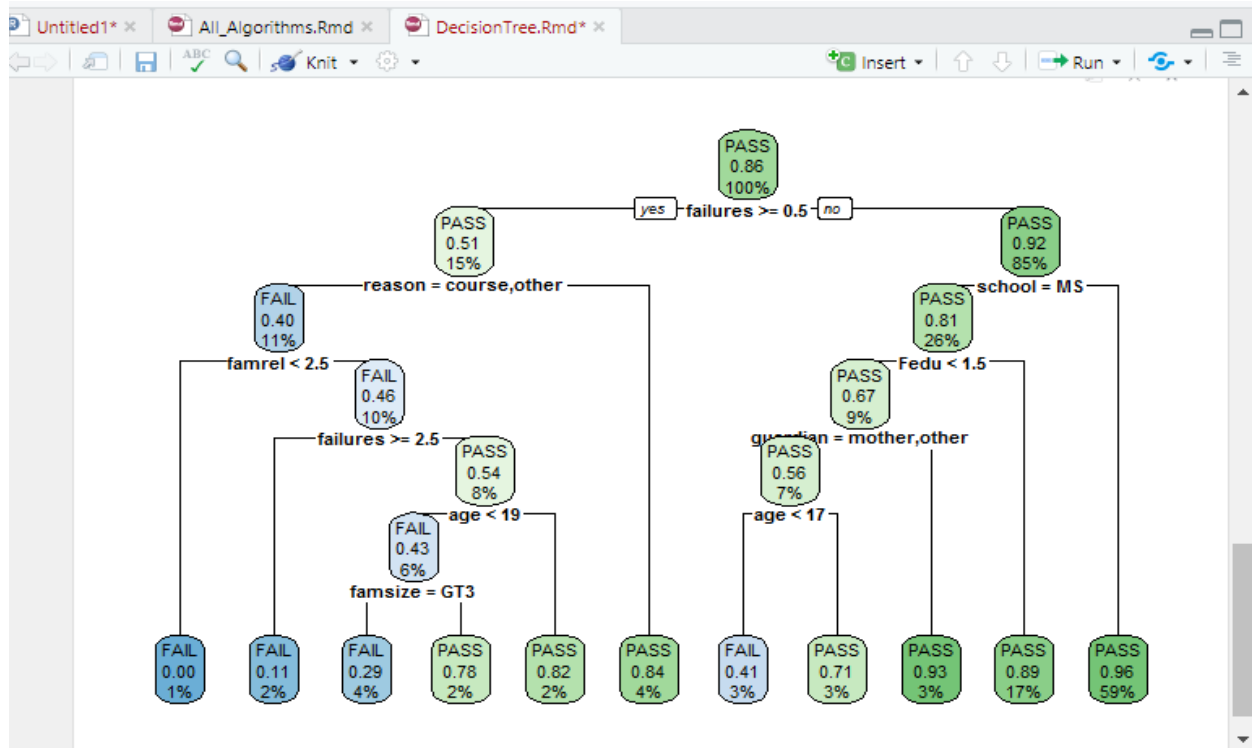
```
## [1] 0.9612403
```

3.4 REPORT

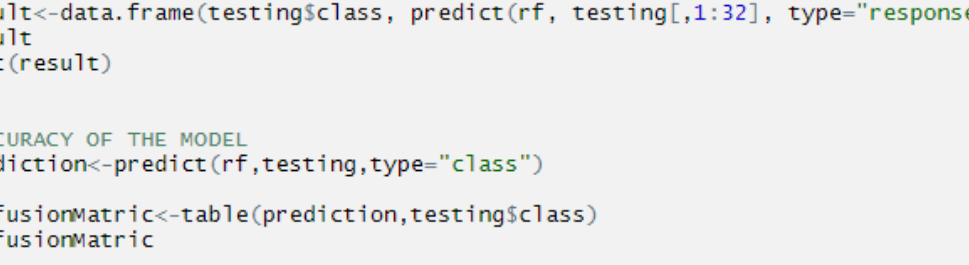
The implementation of decision tree with background details gave 73% accuracy, with social activities attributes 78% accuracy, with course work attributes 93% accuracy and with important attributes it gave 93% accuracy. The implementation of Random forest with background details gave 76% accuracy, with social activities attributes 78% accuracy, with course work attributes 91% accuracy and with important attributes it gave 91% accuracy. The implementation of Logistic Regression with background details gave 87% accuracy, with social activities attributes 83% accuracy, with course work attributes 93% accuracy and with important attributes it gave 92% accuracy. The implementation of Gradient Boosting with background details gave 79% accuracy, with social activities attributes 74% accuracy, with course work attributes 93% accuracy and with important attributes it gave 92% accuracy. The implementation of Logistic Regression with background details gave 83% accuracy, with social activities attributes 83% accuracy, with course work attributes 90% accuracy and with important attributes it gave 93% accuracy.

3.5 EXPERIMENTAL RESULTS

➤ Output of the decision tree



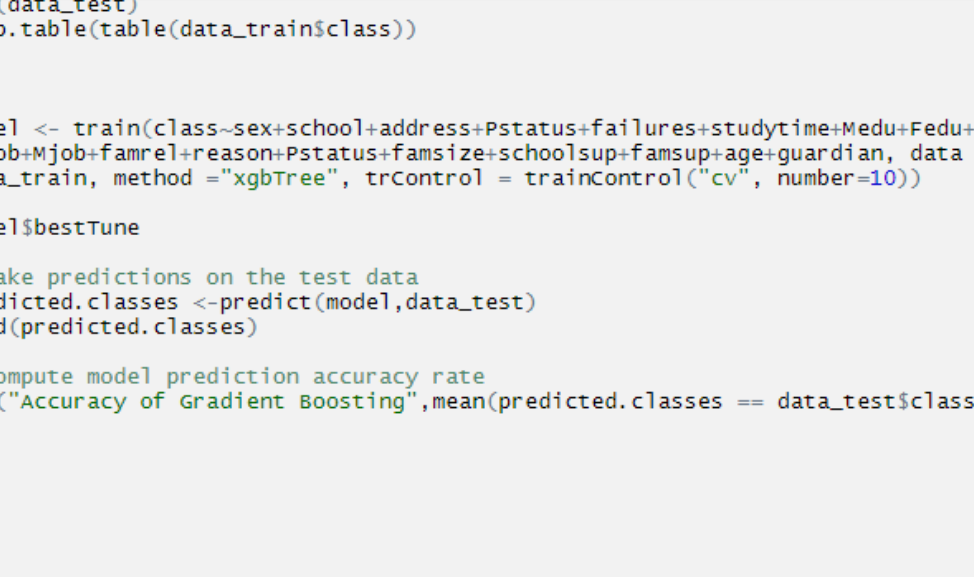
➤ Output of Random forest



```
41 result<-data.frame(testing$class, predict(rf, testing[,1:32], type="response"))
42 result
43 plot(result)
44
45
46 #ACCURACY OF THE MODEL
47 prediction<-predict(rf,testing,type="class")
48
49 ConfusionMatrix<-table(prediction,testing$class)
50 ConfusionMatrix
51
52 accuracy<-sum(diag(ConfusionMatrix))/sum(ConfusionMatrix)
53 print(paste("Accuracy of the test",accuracy))
54
55 #STEP5:VARIABLE IMPORTANCE
56 varImpPlot(rf,sort = T,main="variable Importance",n.var=5)
57
58 var.imp <- data.frame(importance(rf,type=2))
59 # make row names as columns
60 var.imp$variables <- row.names(var.imp)
61 var.imp[order(var.imp$MeanDecreaseGini,decreasing = T),]
62
63 ...
64
```

[1] "Accuracy of the test 0.769230769230769"

➤ Output of Gradient Boosting



The screenshot shows an RStudio window with a script editor and a console. The script editor contains R code for training a Gradient Boosting model and evaluating its accuracy. The console shows the output of the accuracy calculation.

```

49 dim(data_test)
50 prop.table(table(data_train$class))
51
52
53
54 model <- train(class~sex+school+address+Pstatus+failures+studytime+Medu+Fedu+traveltime
+Fjob+Mjob+famrel+reason+Pstatus+famsize+schoolsup+famsup+age+guardian, data =
data_train, method = "xgbTree", trControl = trainControl("cv", number=10))
55
56 model$bestTune
57
58 # Make predictions on the test data
59 predicted.classes <- predict(model,data_test)
60 head(predicted.classes)
61
62 # Compute model prediction accuracy rate
63 cat("Accuracy of Gradient Boosting",mean(predicted.classes == data_test$class))
64
65
66
67
68
69 ...
70

```

Accuracy of Gradient Boosting 0.7923077

➤ Output of Multilayer Perceptron

```
50
51 model <- mlp(
52   stud$inputsTrain, stud$targetsTrain, size=10, learnFuncParams=c(0.1),
53   maxit=50, inputsTest=stud$inputsTest, targetsTest=stud$targetsTest)
54
55 model
56 weightMatrix(model)
57 extractNetInfo(model)
58 par(mfrow=c(2,2))
59 plotIterativeError(model)
60
61 predictions <- predict(model, stud$inputsTest)
62
63 plotRegressionError(predictions[,2], stud$targetsTest[,2])
64
65
66 confusionMatrix(stud$targetsTrain, fitted.values(model))
67 table_mat<-confusionMatrix(stud$targetsTest, predictions)
68 table_mat
69 accuracy_Test <- sum(diag(table_mat)) / sum(table_mat)
70 print(paste('Accuracy for test', accuracy_Test))
71
72
73 | ``
```

➤ Output of Logistic Regression

```
44
45 table(test$class)
46 table(train$class)
47
48 mymodel<-glm(class~sex+school+address+Pstatus+failures+studytime+Medu+Fedu+traveltime+F
49   job+Mjob+famrel+reason+Pstatus+famsize+schoolsup+famsup+age+guardian,
50   family='binomial', data=train, maxit=100)
51 mymodel
52 summary(mymodel)
53
54 retest<-predict(mymodel, test, type="response")
55
56 ROCRPred<-prediction(retest, test$class)
57 ROCRPref<-performance(ROCRPred, "tpr", "fpr")
58 plot(ROCRPref, colorize=TRUE, print.cutoffs.at=seq(0.1, by=0.1))
59
60 confmatrix<-table(Actual_value=test$class, Predicted_value=retest>=0.5)
61 confmatrix
62 accuracy<-sum(diag(confmatrix))/sum(confmatrix)
63 print(paste("Accuracy of the test", accuracy))
64
65 | ``
```

```
[1] "Accuracy of the test 0.871794871794872"
```

```
66
```

➤ Output of Ensemble

MAJORITY VOTING

```
testSet$pred_majority<-as.factor(ifelse(testSet$pred_rf=='PASS' & testSet$pred_gb=='PASS','PASS',ifelse(testSet$pred_rf=='PASS' & testSet$pred_lr=='PASS','PASS',ifelse(testSet$pred_gb=='PASS' & testSet$pred_lr=='FAIL','FAIL','FAIL'))))
```

```
confusionMatrix(testSet$class,testSet$pred_majority)
```

```
## Confusion Matrix and Statistics
##
##      Reference
## Prediction FAIL PASS
##      FAIL    14     6
##      PASS     0    109
##
##               Accuracy : 0.9535
##               95% CI : (0.9015, 0.9827)
##               No Information Rate : 0.8915
##               P-Value [Acc > NIR] : 0.01065
##
##               Kappa : 0.7977
##
##  Mcnemar's Test P-Value : 0.04123
##
##               Sensitivity : 1.0000
##               Specificity : 0.9478
##               Pos Pred Value : 0.7000
##               Neg Pred Value : 1.0000
##               Prevalence : 0.1085
##               Detection Rate : 0.1085
##               Detection Prevalence : 0.1550
##               Balanced Accuracy : 0.9739
##
##               'Positive' Class : FAIL
##
```

➤ Output of Tree Ensemble

MAJORITY VOTING

```
testSet$pred_majority<-as.factor(ifelse(testSet$pred_rf=='PASS' & testSet$pred_gb=='PASS','PASS',ifelse(testSet$pred_rf=='PASS' & testSet$pred_lr=='PASS','PASS',ifelse(testSet$pred_gb=='PASS' & testSet$pred_lr=='FAIL','FAIL','FAIL'))))
```

```
confusionMatrix(testSet$class,testSet$pred_majority)
```

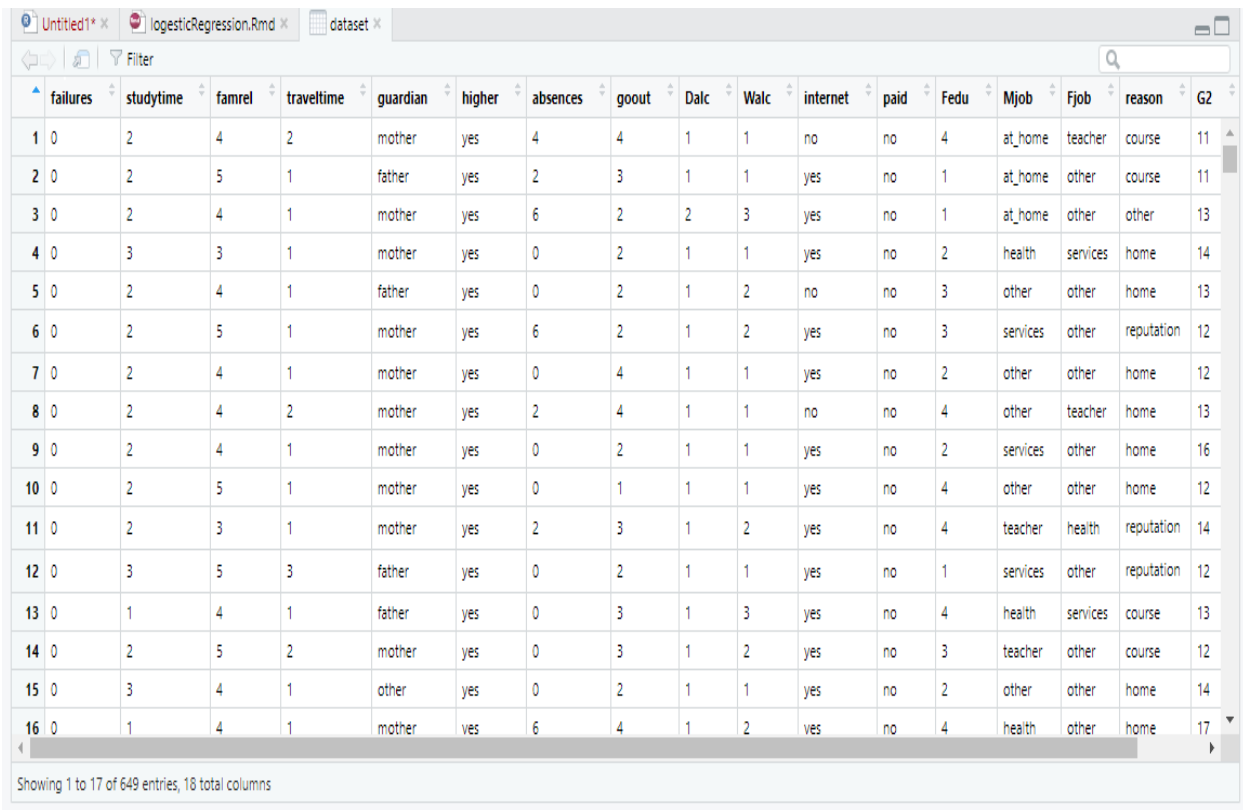
```
## Confusion Matrix and Statistics
##
##      Reference
## Prediction FAIL PASS
##      FAIL    17     3
##      PASS     2    107
##
##               Accuracy : 0.9612
##               95% CI : (0.9119, 0.9873)
##               No Information Rate : 0.8527
##               P-Value [Acc > NIR] : 6.419e-05
##
##               Kappa : 0.849
##
##  Mcnemar's Test P-Value : 1
##
##               Sensitivity : 0.8947
##               Specificity : 0.9727
##               Pos Pred Value : 0.8500
##               Neg Pred Value : 0.9817
##               Prevalence : 0.1473
##               Detection Rate : 0.1318
##               Detection Prevalence : 0.1550
##               Balanced Accuracy : 0.9337
##
##               'Positive' Class : FAIL
##
```

4. IMPLEMENTATION

4.1 FEATURE EXTRACTION

The dataset used in this study is taken from the University of Minho, Mathematics department which consists of 33 attributes and 650 observations. This dataset includes G1, G2, G3 major attributes and G3 containing the average grade of G1 and G2. The response attribute class is created using G3 attribute. This class attribute contains only the pass and fail values, the Student's who got more than 10 in G3 they are placed into pass class and Student's who got less than 10 in G3 are placed in fail category. The dataset is divided into three groups: Background details, Social activities and Course work. The Background details contain Sex, School, Address, Pstatus, famsize, schoolsup, famup, Mjob, Fjob, Fedu, Medu, Reason, Guardian, Famrel, Age, Travel time, Study time and Failures. The Social activities contain Activities, Higher, Internet, Nursery, Paid Class, Romantic, Absences, Health, Free time, Gout, Dalc and Walc. The course work contains G1 and G2 attribute. From the 33 attributes 14 attributes are selected as important attributes by using the Variable Importance function in order to improve the accuracy of the algorithms.

SCREENSHOT



	failures	studytime	famrel	traveltime	guardian	higher	absences	goout	Dalc	Walc	internet	paid	Fedu	Mjob	Fjob	reason	G2
1	0	2	4	2	mother	yes	4	4	1	1	no	no	4	at_home	teacher	course	11
2	0	2	5	1	father	yes	2	3	1	1	yes	no	1	at_home	other	course	11
3	0	2	4	1	mother	yes	6	2	2	3	yes	no	1	at_home	other	other	13
4	0	3	3	1	mother	yes	0	2	1	1	yes	no	2	health	services	home	14
5	0	2	4	1	father	yes	0	2	1	2	no	no	3	other	other	home	13
6	0	2	5	1	mother	yes	6	2	1	2	yes	no	3	services	other	reputation	12
7	0	2	4	1	mother	yes	0	4	1	1	yes	no	2	other	other	home	12
8	0	2	4	2	mother	yes	2	4	1	1	no	no	4	other	teacher	home	13
9	0	2	4	1	mother	yes	0	2	1	1	yes	no	2	services	other	home	16
10	0	2	5	1	mother	yes	0	1	1	1	yes	no	4	other	other	home	12
11	0	2	3	1	mother	yes	2	3	1	2	yes	no	4	teacher	health	reputation	14
12	0	3	5	3	father	yes	0	2	1	1	yes	no	1	services	other	reputation	12
13	0	1	4	1	father	yes	0	3	1	3	yes	no	4	health	services	course	13
14	0	2	5	2	mother	yes	0	3	1	2	yes	no	3	teacher	other	course	12
15	0	3	4	1	other	yes	0	2	1	1	yes	no	2	other	other	home	14
16	0	1	4	1	mother	yes	6	4	1	2	yes	no	4	health	other	home	17

```

40
41 test_pass <- pass[-pass_train, ]
42 test_fail <- fail[-fail_train, ]
43 test<- rbind(test_pass, test_fail) # row bind the pass and fail
44
45 table(test$class)
46 table(train$class)
47
48 mymodel<-glm(class~sex+school+address+Pstatus+failures+studytime+Medu+Fedu+traveltime+Fjob+Mjob+famrel+reason+Pstatus+famsize+schoolsup+famsup+age+guardian, family='binomial', data=train,maxit=100)
49
50 mymodel
51 summary(mymodel)
52
53 retest<-predict(mymodel,test,type="response")
54
55 ROCRPred<-prediction(retest,test$class)
56 ROCRPPref<-performance(ROCRPred,"tpr","fpr")
57
58 plot(ROCRPPref,colorize=TRUE,print.cutoffs.at=seq(0.1, by=0.1))
59
60 confmatrix<-table(Actual_Value=test$class,Predicted_Value=retest>=0.5)
61 confmatrix
62 accuracy<-sum(diag(confmatrix))/sum(confmatrix)
63 print(paste("Accuracy of the test",accuracy))
64
65 ```
66

```

4.2 SAMPLE CODE

```

chisq.test(class,Fjob)

chisq.test(class,Mjob)

chisq.test(class,paid)

chisq.test(class,guardian)

cor(Fedu,G3)

cor(traveltime,G3)

cor(absences,G3)

cor(Walc,G3)

cor(Dalc,G3)

#STEP2:DATA PREPROCESSING

stud<-na.omit(stud)

stud$class[stud$G3>=10]<- 'PASS'

stud$class[stud$G3<10]<- 'FAIL'

```



```

#CONVERTIG TO NUMERIC

stud$school<-as.numeric(stud$school)

stud$address<-as.numeric(stud$address)

stud$sex<-as.numeric(stud$sex)

stud$famsize<-as.numeric(stud$famsize)

stud$Pstatus<-as.numeric(stud$Pstatus)

stud$Mjob<-as.numeric(stud$Mjob)

stud$Fjob<-as.numeric(stud$Fjob)

as.factor(stud)

create_train_test <- function(data, size = 0.8, train = TRUE) {

  n_row = nrow(data)

  total_row = size * n_row

  train_sample  <- 1: total_row

  if (train == TRUE) {

    return (data[train_sample, ])

  } else {

    return (data[-train_sample, ])

  }}

data_train <- create_train_test(stud, 0.8, train = TRUE)

data_test <- create_train_test(stud, 0.8, train = FALSE)

rf<-randomForest(class~failures+studytime+G2+higher+absences+goout+Walc+famrel+reason+Fedu+internet+Fjob,data=data_train, mtry=12, ntree=500, importance=TRUE)

mymodel<-glm(class~failures+studytime+G2+higher+absences+goout+Walc+famrel+reason+Fedu+internet+Fjob, family='binomial', data=data_train,maxit=100)

model <- mlp(stud$inputsTrain, stud$targetsTrain, size=5, learnFuncParams=c(0.1), maxit=50, inputsTest=stud$inputsTest, targetsTest=stud$targetsTest)

```

5. COMPARISON OF RESULTS OF THE ALGORITHMS

Individual Algorithm Accuracy					
S.no	Algorithm	Result	Background Details	Social Activities	Course
1	Decision Tree	Precision	0.93	1	0.85
		Recall	0.83	0.79	0.92
		F-Measure	0.88	0.88	0.88
		Accuracy	0.79	0.78	0.93
2	Random Forest	Precision	0.96	0.96	0.87
		Recall	0.92	0.86	0.95
		F-Measure	0.94	0.91	0.93
		Accuracy	0.76	0.83	0.93
3	Logistic Regression	Precision	0.92	0.96	0.87
		Recall	0.9	0.86	0.95
		F-Measure	0.91	0.91	0.91
		Accuracy	0.87	0.83	0.93
4	Gradient Boosting	Precision	0.97	0.97	0.82
		Recall	0.96	0.91	0.95
		F-Measure	0.96	0.94	0.93
		Accuracy	0.79	0.74	0.93
5	Multilayer Perceptron	Precision	1	0.99	0.86
		Recall	0.79	0.8	0.9
		F-Measure	0.88	0.88	0.88
		Accuracy	0.83	0.83	0.9

Table 6 Individual Algorithm Accuracy


Individual Algorithms Accuracy with Important attributes 					
S.no	Algorithm	Accuracy	Precision	Recall	F-measure
1	Decision Tree	0.93	0.88	0.93	0.91
2	Random Forest	0.915	0.89	0.97	0.92
3	Logistic Regression	0.923	0.88	0.93	0.9
4	Gradient Boosting	0.923	0.88	0.93	0.91
5	Multilayer Perceptron	0.93	0.9	0.92	0.91

Table 7 Individual Algorithms Important Attributes


Ensembling with Important attributes 		
S.no	Algorithm	Accuracy
1	Random Forest	0.937
2	Logistic Regression	0.945
3	Gradient Boosting	0.953
Averaging	Precision	0.88
	Recall	1
	F-Measure	0.93
	Accuracy	0.953
Majority Voting	Precision	0.88
	Recall	0.94
	F-Measure	0.91
	Accuracy	0.953
Weighted Average	Precision	0.88
	Recall	1
	F-Measure	0.93
	Accuracy	0.953

Table 8 Ensemble with Important Attributes


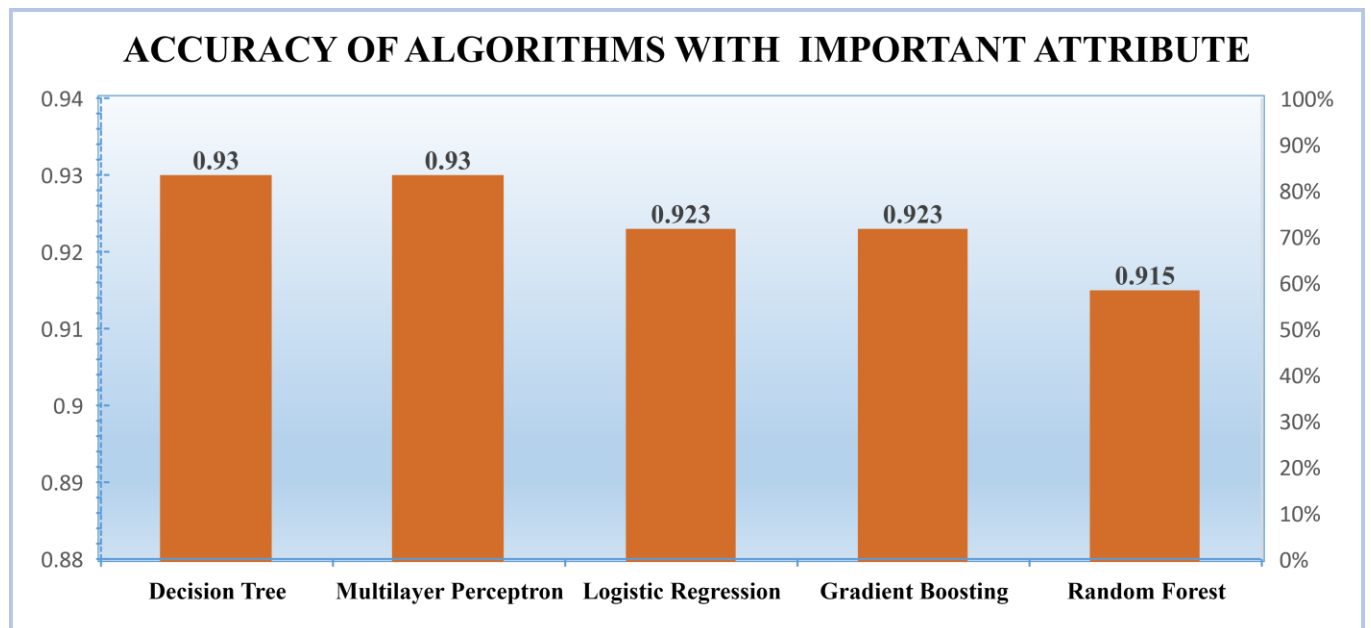
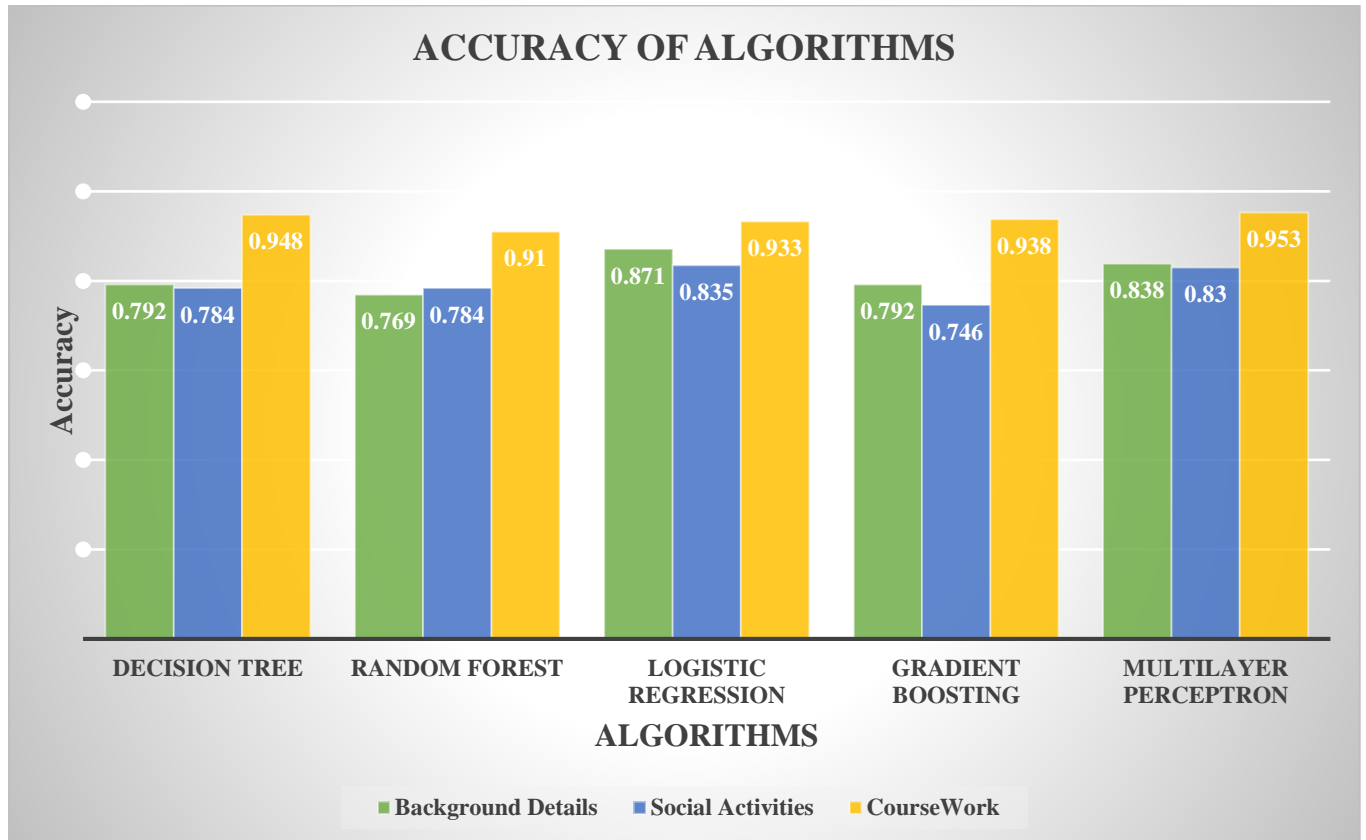
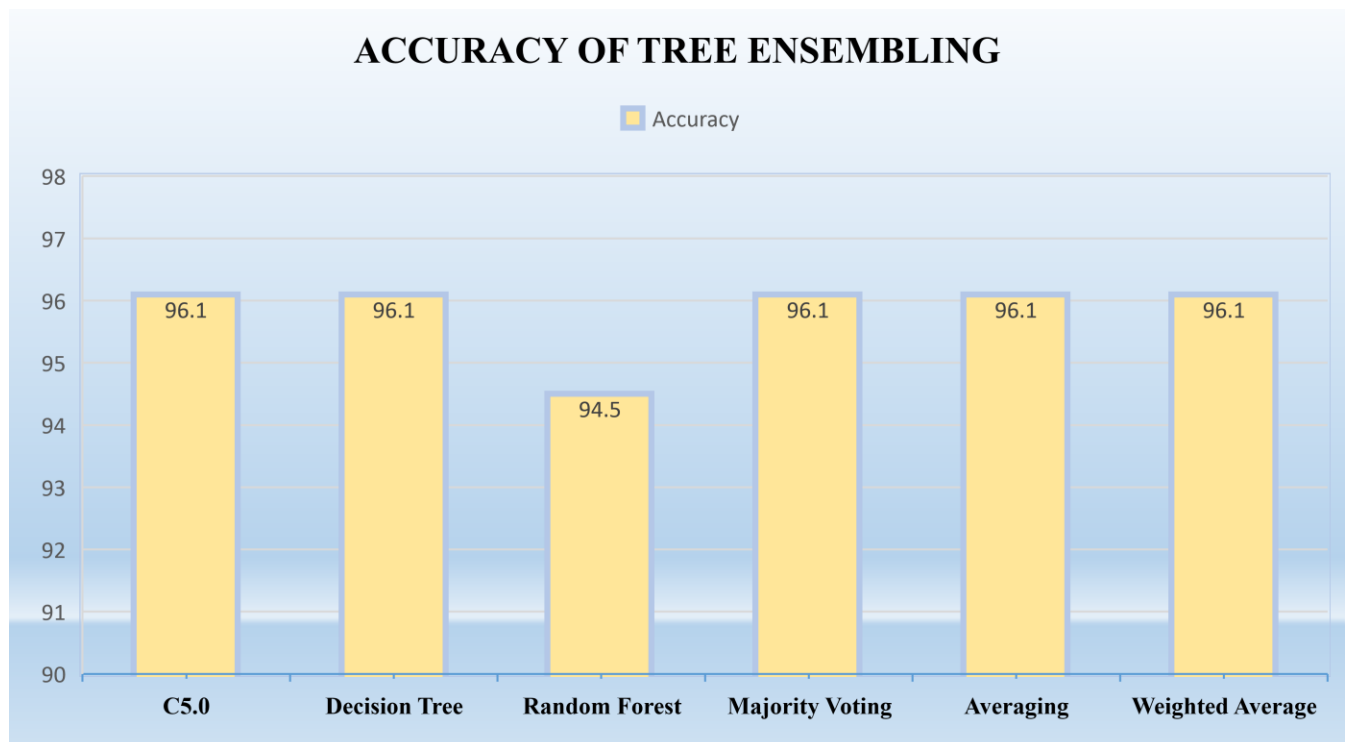
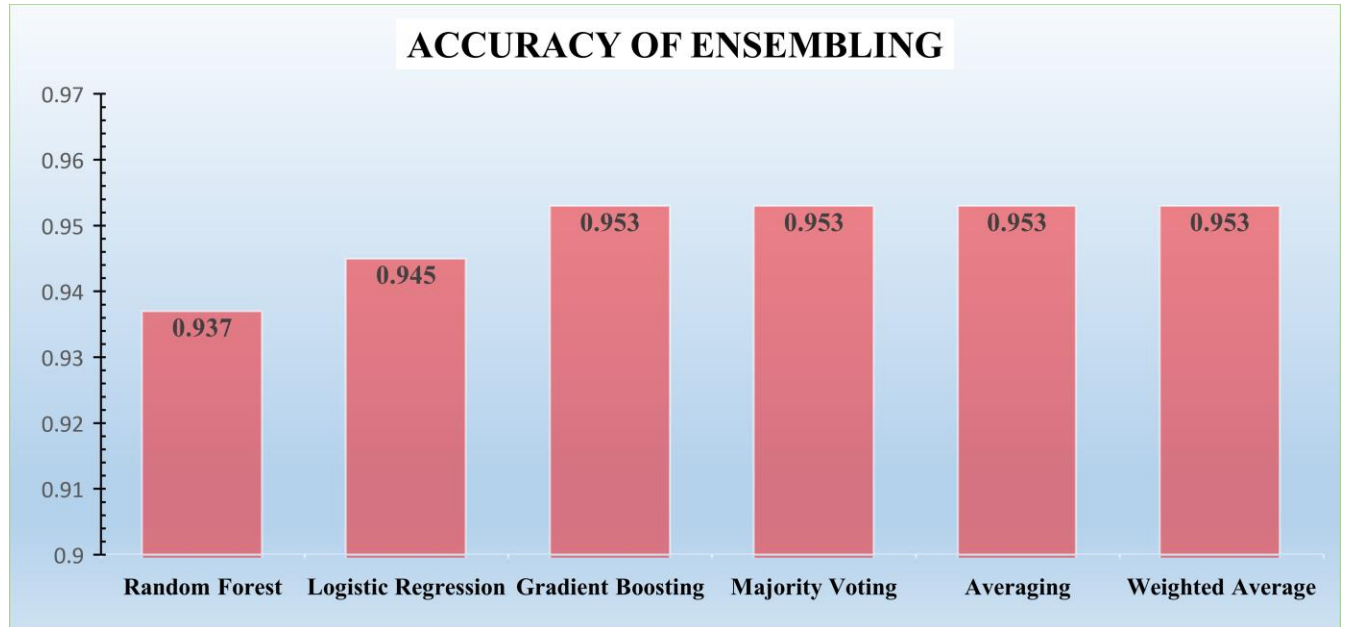
Ensembling Trees with important attributes 		
S.no	Algorithm	Accuracy
1	C5.0	0.961
2	Decision Tree	0.961
3	Random Forest	0.945
Averaging	Precision	0.87
	Recall	1
	F-Measure	0.93
	Accuracy	0.961
Majority Voting	Precision	0.86
	Recall	0.97
	F-Measure	0.91
	Accuracy	0.961
Weighted Average	Precision	0.87
	Recall	1
	F-Measure	93
	Accuracy	0.961

Table 9 Tree Ensemble

ANALYSIS OF RESULTS





6. CONCLUSION


Educational data mining is the interesting field of research for educationalist. With the help of EDM the educational institutions can be benefitted by identifying the weak Student's and give adequate training for improving the performance of the Students. The classification technique is the popular data mining technique used in Educational Data Mining. This study implemented five classification algorithms which are Decision tree, Random Forest, Logistic Regression, Gradient Boosting and Multilayer Perceptron. The Logistic Regression out performed other algorithms with an accuracy of 87% using Background, Social with an accuracy of 83.5% and Course work attributes with 93%. And Multilayer Perceptron and Decision tree out performed other algorithms with an accuracy of 93% using Important attributes. Ensembling with Gradient Boosting, Random Forest and Logistic Regression gave accuracy of 95%. The Tree Ensembling using Decision tree, Random forest and C5.0 gave accuracy of 96%. As the age increases the failure rate also increases, the female Student's face less failures compared to male. If the study time is greater than equal to 3 hours the Students can escape from failure.



7. REFERENCES

- [1] Kiu, Ching-Chieh. "Data Mining Analysis on Student's Academic Performance through Exploration of Student's Background and Social Activities." *2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA)*. IEEE, 2018.
- [2] Kaunang, Fergie Joanda, and Reymon Rotikan. "Student's' Academic Performance Prediction using Data Mining." *2018 Third International Co-nference on Informatics and Computing (ICIC)*. IEEE, 2018.
- [3] Ajibade, Samuel-Soma M., Nor Bahiah Ahmad, and Siti Mariyam Shamsuddin. "An Heuristic Feature Selection Algorithm to Evaluate Academic Performance of Student's." *2019 IEEE 10th Control and System Graduate Research Colloquium (ICSGRC)*. IEEE, 2019.
- [4] Ketui, Nongnuch, Warawut Wisomka, and Kanitha Homjun. "Using Classification Data Mining Techniques for Student's Performance Prediction." *2019 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON)*. IEEE.
- [5] Romero, Cristóbal, et al. "Predicting Student's' final performance from participation in on-line discussion forums." *Computers & Education* 68 (2013): 458-472.
- [6] Hu, Ya-Han, Chia-Lun Lo, and Sheng-Pao Shih. "Developing early warning systems to predict Student's' online learning performance." *Computers in Human Behavior* 36 (2014): 469-478.
- [7] Yu, Liang-Chih, et al. "Improving early prediction of academic failure using sentiment analysis on self_evaluated comments." *Journal of Computer Assisted Learning* 34.4 (2018): 358-365.
- [8] Deepika, K., and N. Sathvanaravana. "Analyze and Predicting the Student's Academic Performance Using Data Mining Tools." *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2018.

- [9] Uzel, Vahide Nida, Sultan Sevgi Turgut, and Selma Ayşe Özel. "Prediction of Student's' Academic Success Using Data Mining Methods." *2018 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE, 2018.
- [10] Siddiqui, Isma Farrah, and Qasim Ali Arain. "ANALYZING STUDENT'S' ACADEMIC PERFORMANCE THROUGH EDUCATIONAL DATA MINING." *3C Tecnologia* (2019).
- [11] Rawat, Keshav Singh, and I. V. Malhan. "A Hybrid Classification Method Based on Machine Learning Classifiers to Predict Performance in Educational Data Mining." *Proceedings of 2nd International Conference on Communication, Computing and Networking*. Springer, Singapore, 2019.

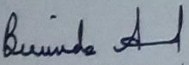
8.APPENDIX

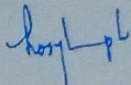

STELLA MARIS COLLEGE (AUTONOMOUS)
CHENNAI - INDIA
Department of Computer Science

 Second International Conference on
**ADVANCEMENTS IN COMPUTING
TECHNOLOGIES** 

January 10 & 11, 2020

This is to certify that Mr./Ms./Dr. ANCY A
has presented a paper titled Identifying data mining techniques and tools for
Improving student's academic performance at the
Second International Conference on Advancements in Computing Technologies.


Ms. Birunda Antoinette Mary J.
Convenor, ICACT 2020
Head, Department of Computer Science


Dr. Sr. Rosy Joseph fmm
Principal