

HYBRID MODELS FOR STUDENTS' ACADEMIC PERFORMANCE AND DROPOUT PREDICTION USING MACHINE LEARNING ALGORITHMS

ALARAPE, MOSHOOD ALABI
(14/52HG001)

HND (Offa) 1997, PGDCS (Minna) 2003, M.Sc. (Abeokuta) 2011

A THESIS SUBMITTED TO THE
DEPARTMENT OF COMPUTER SCIENCE,
FACULTY OF COMMUNICATION AND INFORMATION SCIENCES,
UNIVERSITY OF ILORIN, ILORIN. NIGERIA

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE
DEGREE OF DOCTOR OF PHILOSOPHY (Ph.D.) IN COMPUTER SCIENCE

MAY 2021

CERTIFICATION

This is to certify that the research reported in this thesis was carried out by ALARAPE, Moshood Alabi with Matriculation Number 14/52HG001 in the Department of Computer Science, Faculty of Communication and Information Sciences, University of Ilorin, Ilorin, Nigeria, for the award of Doctor of Philosophy (Ph.D.) Degree in Computer Science.

Dr. A. O. Ameen
(Supervisor)

Date

Dr. R. O. Oladele
(Departmental Postgraduate Coordinator)

Date

Dr. Oluwakemi. C. Abikoye
(Head of Department)

Date

Dr. Rafiat A. Oyekunle
(Internal Examiner)

Date

(External Examiner)

Date

DEDICATION

This research work is dedicated to the Almighty Allah, the Giver of opportunities.

DECLARATION

I, Moshood Alabi ALARAPE, a Ph. D student in the Department of Computer Science, University of Ilorin, hereby declare that this thesis entitled: “*Hybrid Models for Students’ Academic Performance and Dropout Prediction Using Machine Learning Algorithms*” submitted by me is based on my actual and original work. Any materials obtained from other sources or work done by any other persons or institutions have been duly acknowledged. In addition, the research has been approved by the University of Ilorin Ethical Review Committee.

Moshood Alabi ALARAPE

Date

ACKNOWLEDGEMENTS

All praises and thanks are due unto Allah, the Almighty, and the Giver of opportunities, without His permission and approval I would not have been able to get to this level.

My profound appreciation goes to my supervisor, Dr. A. O. Ameen for his mentorship, guidance, and support. I enjoyed the kind of relationship he established with me during my study. I pray to Allah to continue covering him with His unequal mercies.

My unparalleled gratitude goes to the Dean, Faculty of Communication and Information Sciences, Prof. R. G. Jimoh, the one whom Allah sent to me for the accomplishment of this programme. He was extremely useful for me throughout my study. I also want to register my profound appreciation to our success-loving mother and Head of Department, Computer Science, Dr. (Mrs.) O. C. Abikoye for her unflinching support, love, and encouragement and the immediate past Departmental PG Coordinator, Dr. (Mrs.) T. O. Oladele for her guidance and support. Also noteworthy are the contributions of Dr. K. S. Adewole for his all-time guidance and mentorship, Dr. (Mrs.) G. A. Akintola for her efforts in getting part of the data for the study, Dr. Bajeh for his love and concern at all time, Dr. (Mrs.) M. A. Mabayoje for her sisterly love, Dr. A. O. Babatunde, Dr. A. R. Ajiboye – for his foundational guide in the initial stage of the work, Dr. R. O. Oladele for the scrutiny of the work, Mr. H. A. Mojeed, Mr. A. O. Balogun and all other members of staff of the Department of Computer Science, University of Ilorin both academic and non-academic for their great support during the programme.

My special appreciation goes to my sister and former Head of Department of Information and Communication Science, Dr. (Mrs) R. A. Oyekunle for her effort on my work at the inception and her regular encouragement. I cannot also forget the brotherly support of Dr. Olawoyin L. A. of the Telecommunication Science department, Mr. Mohammed K. J. of the Educational Technology

department, and the security officers deployed to the CIS building, all in the University of Ilorin during the intense time of the programme.

I also want to enlist my employer, the Federal Polytechnic, Bida for the study leave and Tetfund opportunity given to me for the study. My sincere appreciation goes to all my colleagues in my working place for their prayers and various assistance during the programme. I am also greatly indebted to the following senior colleagues in my working place: Dr. O. I. Adejumo, Dr. I. A. Bolarinwa, and Dr. M. A. Asiru for their constant encouragement towards the success of the programme.

I am greatly indebted to my late parents, Alhaji Akeeb Alarape and Hajia Rodiyat Aweni Alarape for giving me the background training that strengthened me for this academic achievement. I ask Allah to forgive and have His mercies on them. My lovely and sincere appreciation is due to my loving zawjat, Umm Anas AbdulKareem Kabirat, and all my children: Anas Bamidele Alarape, Rodiyat Temitope Alarape, Ali Adesina Alarape, and Nusaybah Oluwalami Alarape for their steadfastness and endurance during my absence from home for the programme. I ask Allah to make this achievement of mine the beginning of His favours on all of us in the family. Also appreciated is the efforts of my wife junior sister, Hamida AbdulKareem for her assistance at home during my absence.

My thanks go to my loving sisters: Mrs. K. A. Okunlola and Mrs. S. O. Ogunsuyi, my brothers: Alhaji Alani Adeyemi (ALAN) and Alhaji Dauda Alarape for their moral support.

Finally, I sincerely thank all my colleagues in the Ph.D. programme for their support and cooperation during the programme. My special thanks go to my mentor and senior, Dr. T. O. Aro for his assistance at the initial stage of the programme. And to all others who had contributed in one way or the other to the success of this programme, too numerous to mention.

TABLE OF CONTENTS

Title Page	i
Certification	ii
Dedication	iii
Declaration	iv
Acknowledgements	v
Table of Contents	vii
List of Equations	xii
List of Figures	xiii
List of Tables	xv
Abbreviations and Acronyms	xvii
Abstract	xxi

CHAPTER ONE: INTRODUCTION

1.1	Background to the Study	1
1.2	Statement of the Problem	4
1.3	Aim and Objectives	6
1.4	Significance of the Study	6
1.5	Scope of the Study	7
1.6	Definition of Operational Term	8

CHAPTER TWO: LITERATURE REVIEW

2.1	Data Mining and Knowledge Discovery	10
2.2	Data Mining Methods	13
2.3	Feature Selection Techniques	17
2.3.1	Filter Models	18

2.3.2 Wrapper Models	19
2.3.2.1 Recursive Feature Elimination (RFE)	19
2.3.3 Embedded Models	20
2.4 Educational Data Mining	21
2.5 Students' Academic Performance Prediction	23
2.5.1 Factors Affecting Students' Academic Performance (SAP): Educational Psychology Perspective	23
2.5.2 Students' Academic Performance Datasets	24
2.5.3 Taxonomy Based on Features	25
2.5.4 Taxonomy of Methods Used for SAP and Dropout Predictions	30
2.5.4.1 Naïve Bayes	33
2.5.4.2 Decision Trees	39
2.5.4.3 Logistic Regression (LR)	42
2.5.4.4 Support Vector Machines (SVM)	44
2.5.4.5 Neural Network (NN)	52
2.5.4.6 k-Nearest Neighbour (kNN)	54
2.5.4.7 Random Forest (RF)	55
2.5.4.8 Clustering	57
2.5.4.9 Association Rule Mining	57
2.5.4.10 Nature-Inspired Algorithms (NIA)	58
2.5.4.11 Deep Learning	60
2.5.4.12 Hybrid Methods	61
2.6 Genetic Algorithm	63

2.7	Distribution of Selected Studies by Year of Publications	65
2.8	Related Works	79
2.9	Research Gaps	81
CHAPTER THREE: METHODOLOGY		
3.1	Preliminary Stages of the Hybrid Prediction Models Design	82
3.1.1	Data Collection and Preparation	82
3.1.2	Auto-Classifer Detection Stage	84
3.2	Design of the Proposed Hybrid Prediction Models	85
3.2.1	Feature Selection Layer	87
3.2.1.1	The Recursive Feature Elimination Strategy	87
3.2.1.2	The Genetic Search Algorithm Strategy	89
3.2.2	Classification Layer	92
3.2.3	Model Evaluation Layer	97
3.3	Evaluation Metrics	98
CHAPTER FOUR: RESULTS AND DISCUSSION		
4.1	Exploratory Data Analysis of the Preprocessed Data	100
4.1.1	Exploratory Data Analysis of UniDataSAP_num dataset	101
4.1.2	Exploratory Data Analysis of PolyDataSAP_num dataset	102
4.1.3	Exploratory Data Analysis of UniDataDropout_num dataset	104
4.1.4	Exploratory Data Analysis of PolyDataDropout_num dataset	105
4.2	Results of the Auto-Classifer Detection Stage	106
4.2.1	Analysis of Results from Experiment I	106
4.2.2	Analysis of Results from Experiment II	107

4.2.3 Analysis of Results from Experiment III	107
4.2.4 Analysis of Results from Experiment IV	108
4.2.5 Summary of Findings from the Auto-Classifer Detection Stage	109
4.3 Results of the Proposed Hybrid Prediction Models	111
4.3.1 SAP Hybrid Prediction Models (SAPHPM)	112
4.3.1.1 RFE_SVM-NB Models	112
4.3.1.2 GA-NB Models	113
4.3.2 Dropout Hybrid Prediction Models (DHPM)	117
4.3.2.1 RFE_NB-SVM Models	117
4.3.2.2 GA-SVM Models	118
4.4 Performance Analysis of the Proposed Hybrid Prediction Models	121
4.4.1 Analysis of SAPHPM	122
4.4.1.1 Comparison of the Two Categories of the Proposed Hybrid SAP Models	122
4.4.1.2 Comparison of the Two Categories of the Proposed Hybrid SAP Models with Single Classifiers	123
4.4.2 Analysis of DHPM	125
4.4.2.1 Comparison of the Two Categories of the Proposed Hybrid Dropout Models	125
4.4.2.2 Comparison of the Two Categories of the Proposed Hybrid Dropout Models With Single Classifiers	126
4.5 Comparative Analysis of the Proposed Hybrid Models with Existing Models	129
CHAPTER FIVE: SUMMARY, CONCLUSION AND RECOMMENDATIONS	
5.1 Summary	132
5.2 Conclusion	134

5.3	Recommendations	135
5.4	Contributions to Knowledge	136
5.5	Suggestions for Further Studies	137
REFERENCES		138
APPENDIX		152

LIST OF EQUATIONS

Equation 2.1: Probability of Class C_i given attributes X	33
Equation 2.2: Condition for label prediction in Naïve Bayes Algorithm	34
Equation 2.3: Support Vector Machine maximum margin equation	45
Equation 2.4: Expansion of Equation 2.3	45
Equation 2.5: Condition to be satisfied by two separating hyperplane (for positive side)	45
Equation 2.6: Condition to be satisfied by two separating hyperplane (for negative side)	45
Equation 2.7: Positive Hyperplane equation H_1	46
Equation 2.8: Negative Hyperplane equation H_2	46
Equation 2.9: Combination of equations H_1 and H_2 for all values	46
Equation 2.10: Lagrangian formulation equation to train SVM	47
Equation 2.11: Kernel function equation	47
Equation 3.1 Formula for Computing the Accuracy of a Classifier	98
Equation 3.2 Formula for Computing the Precision of a Classifier	98
Equation 3.3 Formula for Computing the Sensitivity of a Classifier	98
Equation 3.4 Formula for Computing the Specificity of a Classifier	98
Equation 3.5 Formula for Computing the F-measure of a Classifier	99
Equation 3.6 Formula for Computing the Geometric Mean of a Classifier	99
Equation 3.7 Formula for Computing the Youden's Index of a Classifier	99

LIST OF FIGURES

Figure 2.1(a) Learning in Data Mining	14
Figure 2.1(b) Classification in Data Mining	14
Figure 2.2 Feature Selection Process	18
Figure 2.3 Wrapper Feature Selection Process	19
Figure 2.4 Taxonomy based on Features	26
Figure 2.5 Taxonomy of Methods for SAP and Dropout Prediction	30
Figure 2.6 An illustration of SVM with two classes	46
Figure 2.7 Hard SVM	46
Figure 2.8 Soft SVM	48
Figure 2.9 Distribution of Selected Studies by Year of Publications	78
Figure 3.1 Algorithm for Naïve Bayes	85
Figure 3.2 Algorithm for Support Vector Machines	86
Figure 3.3 Procedures involved in the Auto-Classifer Detection Stage	87
Figure 3.4 Conceptual Framework for RFE-based Hybrid Prediction Models	88
Figure 3.5 Recursive Feature Elimination Algorithm	89
Figure 3.6 Conceptual Framework for GA-based Hybrid Prediction Models	90
Figure 3.7 Genetic Algorithm Procedure	91
Figure 3.8 Model Architecture of RFE_SVM-NB	92.
Figure 3.9 Algorithm for RFE_SVM-NB model	93
Figure 3.10 Model Architecture of GA-NB	94
Figure 3.11 Model Architecture of RFE_NB-SVM	94
Figure 3.12 Algorithm for GA-NB model	95

Figure 3.13	Algorithm for RFE_NB-SVM model	96
Figure 3.14	Model Architecture of GA-SVM	96
Figure 3.15	Algorithm for GA-SVM model	97
Figure 4.1	Chart showing the Instances-Attributes across the Four Datasets	100
Figure 4.2	Instances Versus Classes in UniDataSAP_num dataset	101
Figure 4.3	Percentage distribution of classes in UniDataSAP_num dataset	102
Figure 4.4	Instances Versus Classes in PolyDataSAP_num dataset	103
Figure 4.5	Percentage distribution of classes in PolyDataSAP_num dataset	103
Figure 4.6	Instances Versus Classes in UniDataDropout_num dataset	104
Figure 4.7	Percentage distribution of classes in UniDataDropout_num dataset	104
Figure 4.8	Instances Versus Classes in PolyDataDropout_num dataset	105
Figure 4.9	Percentage distribution of classes in PolyDataDropout_num dataset	105
Figure 4.10	Features Selected from UniDataSAP_num Using RFE_SVM Strategy	112
Figure 4.11	Features Selected from PolyDataSAP_num Using RFE_SVM Strategy	112
Figure 4.12	Features Selected from UniDataSAP_num Using GA Strategy	115
Figure 4.13	Features Selected from PolyDataSAP_num Using GA Strategy	116
Figure 4.14	Features Selected from UniDataDropout_num Using RFE_NB Strategy	117
Figure 4.15	Features Selected from PolyDataDropout_num Using RFE_NB Strategy	117
Figure 4.16	Features Selected from PolyDataDropout_num Using GA Strategy	120
Figure 4.17	Features Selected from PolyDataDropout_num Using GA Strategy	120

LIST OF TABLES

Table 2.1	Summary of SAP and Dropout Prediction Researches from 2007 to 2018	66
Table 3.1	Data Attributes and their descriptions	83
Table 3.2	GA Parameters for feature selection	91
Table 4.1	Results of Experiment I	106
Table 4.2	Results of Experiment II	107
Table 4.3	Results of Experiment III	108
Table 4.4	Results of Experiment IV	108
Table 4.5	Results of RFE_SVM-NB Model on SAP datasets	113
Table 4.6	Results of GA-NB Model on SAP datasets at population = 20	114
Table 4.7	Results of GA-NB Model on SAP datasets at population = 50	115
Table 4.8	Results of GA-NB Model on SAP datasets at population = 100	116
Table 4.9	Results of RFE_NB-SVM Model on Dropout datasets	118
Table 4.10	Results of GA-SVM Model on Dropout datasets at population = 20	119
Table 4.11	Results of GA-SVM Model on Dropout datasets at population = 50	120
Table 4.12	Results of GA-SVM Model on Dropout datasets at population = 100	121
Table 4.13	Results of Comparison of the Two Hybrid Models on UniDataSAP_num	122
Table 4.14	Results of Comparison of the Two Hybrid Models on PolyDataSAP_num	123
Table 4.15	Results of Comparison of the Two Hybrid Models and Single Classifiers on UniDataDropout_num	124
Table 4.16	Results of Comparison of the Two Hybrid Models and Single Classifiers on PolyDataDropout_num	124

Table 4.17 Results of Comparison of the Two Hybrid Models on UniDataDropout_num	125
Table 4.18 Results of Comparison of the Two Hybrid Models on PolyDataSAP_num	126
Table 4.19 Results of Comparison of the Two Hybrid Models with Single Classifiers on UniDataDropout_num	127
Table 4.20 Results of Comparison of the Two Hybrid Models with Single Classifiers on PolyDataDropout_num	128
Table 4.21 Results of Comparison of the Hybrid Models with Burman & Som (2019)	129
Table 4.22 Results of Comparison of the Hybrid Models with Amra & Maghari (2017)	130

ABBREVIATIONS AND ACRONYMS

ACO	Ant Colony Optimization
ADASYN	Adaptive Synthetic Sampling
ANFIS	Adaptive Neuro-Fuzzy Inference System
ANN	Artificial Neural Network
AuC	Area Under Curve
BP	Backward Propagation
CART	Classification and Regression Technique
CGPA	Cumulative Grade Point Average
CHAID	Chi-Square Automatic Interactive Detection
COA	Cuckoo Optimization Algorithm
CNN	Convolutional Neural Network
CS	Cuckoo Search
CSE	Computer Science and Engineering
DBM	Deep Boltzmann Machine
DBN	Deep Belief Network
DHPM	Dropout Hybrid Prediction Models
DM	Data Mining
DNN	Deep Neural Network
DT	Decision Tree
EDM	Educational Data Mining
ELM	Extreme Learning Machine
EN	Entropy Net

EP	Educational Psychology
ESAP	Emotional Skill Assessment Process
FNB	Fuzzy Naïve Bayes
FP	False Positive
FS	Feature Selection
GA	Genetic Algorithm
GA-NB	GA-based Hybrid Model
GA-SVM	GA-based Hybrid Model
GAP	Groups, Algorithms and Programming
GBGP	Grammar-Based Genetic Programming
GP	Genetic Programming
GPA	Grade Point Average
GGP	Grammar-based Genetic Programming
HEI	Higher Educational Institution
HSG	Higher School Grade
ICRM	Interpretable Classification Rule Mining
ID3	Interactive Dichotomizer 3
KEEL	Knowledge Extraction based on Evolutionary Learning
KNN	k-Nearest Neighbor
LMS	Learning Management System
LOOCV	Leave Out One Cross Validation
LSTM	Long Short Term Memory
LR	Logistic Regression

MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MATLAB	Matrix Laboratory
MCA	Masters in Computer Applications
MLP	Multi-Layer Perceptron
MNB	Multinomial Naïve Bayes
MOOC	Massive Open Online Course
NB	Naïve Bayes
NBC	Naïve Bayes Classifier
NIA	Nature Inspired Algorithms
NN	Neural Network
NNEP	Neural Network Evolutionary Programming
PCC	Percentage Correctly Classified
PG	Post Graduate
pHEI	Pre-Higher Educational Institution
PSO	Particle Swarm Optimization
R	Correlation Coefficient
RAE	Relative Absolute Error
RBF	Radial Basis Function
RBM	Restricted Boltzmann Machine
RF	Random Forest
RFE	Recursive Feature Elimination
RFE_SVM-NB	RFE-based Hybrid Model with NB classifier

RFE_NB-SVM	RFE-based Hybrid Model with SVM classifier
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristics
SAP	Students' Academic Performance
SAPHPM	Students' Academic Performance Hybrid Prediction Model
SEMD	Students Entry Management Database
SIMD	Single Instruction Multiple Data
SMO	Sequential Minimal Optimization
SMOTE	Synthetic Minority Over-sampling Technique
SVM	Support Vector Machines
S3PSO	Students' Performance Prediction based on Particle Swarm Optimization
TP	True Positive
USA	United States of America
UAR	Unweighted Average Recall
WEKA	Waikato Environment for Knowledge Analysis
XCS	eXtended Classifier System

ABSTRACT

Predicting Students' Academic Performance (SAP) and dropout are separate but very important concepts that must be measured accurately. Existing methods for measuring these concepts focused on the evaluation of a particular classifier and comparison between classifiers, with accuracy employed only to the detriment of other useful metrics leading to erroneous judgments about classifiers so, there is need for efficient classifiers. This study aimed at developing hybrid models for predicting SAP and dropout using machine learning algorithms. The objectives of the study were to: (i) investigate the performances of Support Vector Machine (SVM) and Naïve Bayes (NB) algorithms; (ii) design hybrid algorithms for SAP and dropout prediction from SVM and NB; (iii) design hybrid algorithms for SAP and dropout prediction from SVM or NB with GA (Genetic Algorithm); (iv) implement the hybrid models designed; and (v) conduct performance evaluation of the developed hybrid models in order to compare them with other existing models. The study collected real life datasets from the University of Ilorin, Ilorin, and Federal Polytechnic Bida in Nigeria. The datasets were analysed and preprocessed in MS-Excel. SVM and NB algorithms were applied to determine the best classifier for each of the SAP and dropout datasets; and the best kernel for SVM models. Recursive Feature Elimination (RFE) strategy for feature selection was first used with a baseline model as the top layer and the best of SVM and NB for classification at the bottom to produce RFE_SVM-NB and RFE_NB-SVM models. The GA-based models developed used GA for feature selection at the top layer and the best of SVM and NB for classification at the bottom. Accuracy (A_{cc}), Sensitivity (S_e), Specificity (S_p), Precision (P_r), F-measure (F_{score}), Geometric Mean (GM), and Youden's Index (YI) were used as performance metrics.

The findings of the study were that:

- (i) NB adjudged best with 0.8174(S_e), 0.9208(S_p), and 0.7852(P_r) on SAP datasets while the SVM with radial basis function kernel was adjudged best in the Auto-Classifer Detection stage;
- (ii) RFE_SVM-NB algorithm developed had 90.55%(A_{cc}), 0.8808(S_e) and 0.91(P_r) on PolyDataSAP_num dataset while RFE_NB-SVM model developed had 93.43%(A_{cc}), 0.8095(S_e) and 0.7727(P_r) on PolyDataDropout_num dataset;
- (iii) GA-NB and GA-SVM models were developed, GA-NB had 94.55%(A_{cc}), 0.9756(S_p), and 0.7197(P_r) on UniDataSAP_num dataset while GA-SVM had 92.7%(A_{cc}), 0.7143(S_e), and 0.7895(P_r) on PolyDataDropout_num dataset;
- (iv) RFE_SVM-NB model yielded the best values of 0.8808(S_e), 0.8952(F_{score}), 0.8957(GM) and 0.8352(YI) on PolyDataSAP_num dataset while RFE_NB-SVM model yielded 0.9091(S_e), 0.8333(F_{score}), 0.8884(GM) and 0.8409(YI) on UniDataDropout_num dataset; GA-NB model yielded 0.8344(S_e), 0.7531(F_{score}), 0.8868(GM) and 0.7769(YI) on PolyDataSAP_num while GA-SVM model yielded 0.7143(S_e), 0.75(F_{score}), 0.8305(GM) and 0.6798(YI) on PolyDataDropout_num dataset; and
- (v) GA-based models outperformed the existing models (94.55%(A_{cc}), 0.8301(GM), 0.6819(YI)) against (92.59%(A_{cc}), 0.8049(GM) and 0.6597(YI)) while the RFE-based models performance were comparable with the existing models (88.89%(A_{cc}), 0.7967(GM), 0.651(YI)) against (90.74%(A_{cc}), 0.7902(GM) and 0.6382(YI)).

The study concluded that the proposed models were of high performance with GA-based models overwhelmingly outperforming the existing models aside from the reduction in models' complexity. The study recommended that the models be used in educational institutions for predicting students' academic performances.

CHAPTER ONE

INTRODUCTION

1.1 Background to the Study

Education is regarded as the backbone of every nation's development. The development of any educational institution cannot be achieved without proper care of the students in such a school; which is mostly done by ensuring regular monitoring of the students' performances and solving their emotional challenges. This is usually contained in most of the students' information handbooks of many tertiary institutions as a responsibility of the class tutor or adviser, that is the lecturer/instructor in charge of each level of students, however, this important responsibility is always ignored and unstressed in the real sense. Where monitoring of students' performances is done it is always done subjectively by the person tutoring the students (Alexander, 1996, as cited in Asogbon, Samuel, Omisore, & Ojokoh, 2016).

This role of the course/level advisers is being addressed scientifically by the Educational Data Mining (EDM) researchers through the use of data mining techniques with the required objectivity to achieve an efficient result.

EDM is the application of data mining techniques to educational data with the objective of analyzing these data to resolve educational research issues (Barnes et al., 2009, as cited in Romero & Ventura, 2010). It has also been defined as an emerging interdisciplinary research area that deals with the development of methods to explore data originating from an educational context (Suhirman, Zain, & Herawan, 2014).

The sacred goal of every academic institution is to entrench learning and this cannot be said to have taken place except that the students are correctly evaluated. This evaluation which used to be in the form of class tests, assignments, practical/laboratory works, and semester examinations must

be done holistically to evolve the academic status of each of the students in a class in different courses taken in the semester (Ameen, Alarape, & Adewole, 2019). The performance of the student is, therefore, one of the crucial aspects of every educational institution (Solomon, Patil., & Agrawal, 2018) because without it there cannot be students' progression.

The students' academic programmes are designed in such a way that they go through two semesters in a session and the whole programme may consist of four or five sessions. Students are assessed semesterly in a semester-based programme and each semester is a link to the next semester till the completion of the programme. It, therefore, gives the opportunity to use a previous semester performance of a student to predict his performance in the next or even on graduation.

Educational institutions are greatly characterized by the dumping of large databases arising from the computerization of many transactions on daily basis. These data repositories include the lecturers/instructors databases, students' registration databases, students' academic databases, students' admission/enrolment databases, and so on. All these databases are known as educational databases and the data in them could be mined to reveal important information about lecturers, instructors, students, and learning activities for management planning, scheduling, and decision making. It was the researches in this area that gave rise to an aspect of data mining called Educational Data Mining.

The hidden knowledge extracted from these educational data can help the educators to improve teaching techniques, understand learners, improve the learning process and it could also be used by learners to improve their learning activities (Romero & Ventura, 2010). The information obtained from the result of mining educational data can be used for any educational purpose as required by the administrators. For instance, Witten (2017) posited that the mining of data generated from courses such as the online data mining course can help predict students' future

learning success, assess the effect of educational support, and advance scientific knowledge about learning. Perhaps individual learning patterns can be discovered and used to personalize the presentation of materials.

Different machine learning algorithms have been used in mining educational data, these include Classification and Regression Technique (CART), ID3 (Interactive Dichotomizer 3), SVM (Support Vector Machine), NB (Naïve Bayes), C4.5, C5.0, k-NN (k-Nearest Neighbor), k-means, and so on. The most explored study area in the field of Educational Data Mining is the prediction of students' academic performance (SAP). A lot of studies in EDM had used a combination of factors relating to students' academic performance to predict students' performance on data collected from universities, colleges, and even secondary schools. Most of the studies only predict the success or failure of students; other studies were on the prediction of students' dropout rates. The series of literature reviewed in this study is evidence of this. The findings in most of these studies like Dekker, Pechenizkiy & Vleeshouwers (2009) and Asif, Merceron, Ali & Haider (2017) are helpful in the advancement of the students' learning and behavior towards their academic success and the progress of the institution.

The most important and very interesting ones are the ones addressing the dropout rates of students which have a direct impact on the institutions ranking and revenue generation, the study of which can assist the management of institutions to take preventive measures to avert students' dropout. The focus of those studies is to devise methods and early solutions towards the full retention of students once admitted. Dropout students are of two types, one is a dropout which is due to academic failure while the other is a dropout that is due to other reasons like financial incapacitation, sickness, and so on. This agrees with the view of the father of researches in psychology on students' dropout and retention management, Vincent Tinto in 1975. He proposed,

in his theory of dropout, that students' dropout can be viewed as a longitudinal process of interactions between the individual (student) and the academic and social systems of the college during which a person's experiences in those systems (as measured by his normative and structural integration) continually modify his goal and institutional commitments in ways which lead to persistence and/or to varying forms of dropout (Tinto, 1975).

According to Romero & Ventura (2007), the quest for efficient data mining strategies for educational data mining goals; of which predicting students' academic performance is paramount, is unquenchable and that is the focus of this research study which seeks to develop efficient hybrid Students' Academic Performance (SAP) and dropout prediction models using Support Vector Machines, Naïve Bayes algorithms and Genetic Algorithm (GA).

1.2 Statement of the Problem

The prediction of students' performances requires a lot of objectivity to attract the required accuracy for the targeted goal of efficient prediction to be achieved. This is because the wrong classification of students' performances could have a devastating effect on the students, tutor/level advisers, parents, and other stakeholders in the educational sector. Some of these devastating effects are the cases where students commit suicide owing to academic failure as reported in some of the Nigerian dailies (Agbedo, 2019) and the application of wrong strategies for enhancing students' performances by the stakeholders, most especially in dropout cases.

In addition, most studies in EDM focused on the comparison of the performances of Data Mining (DM) algorithms for Students' Academic Performance (SAP) and dropout predictions as separate tasks (Rovira, Puertas & Igual (2017); Strecht, Cruz, Soares, Mendes-Moreira & Abreu (2015); Osofisan, Adeyemo & Oluwasusi (2014) and Romero, Ventura, Espejo & Hervás (2008)). Others

are restricted to the investigation of the performance of a particular data mining algorithm on educational data (Al-barrak & Al-razgan (2016); Ogunde & Ajibade (2014) and Ogor (2007)).

The majority of these studies based their decision on classifiers' performances on predictive accuracy alone to the detriment of other metrics and it lends way to misleading judgement about classifier (Akosa, 2017). Application of a single data mining algorithm to education data as evident in the literature is also insufficient to achieve the required efficiency in the mining of the data and hence the combination of some of these algorithms either as an ensemble (Abikoye, Omokanye & Aro, 2017) or in form of hybrid (Sagale & Kale, 2014; Shakya & Sigdel, 2017) have always proof better than the single application of these algorithms as manifested in other domains. Although there are scanty studies that hybridize two or more algorithms for SAP and dropout prediction (Altaher & Barukab, 2018 and Chen *et al.*, 2019) and aside from this, the hybridization of the widely used classification algorithms in the area have not been attended to.

All these points to the need for hybridization of techniques for SAP and dropout prediction using popular DM algorithms like Naïve Bayes and Support Vector Machines.

This study, therefore, addresses the problem of over-reliance on predictive accuracy as a performance metric and the unquenchable search for novel and more efficient classifiers for predicting SAP and dropout by developing hybrid models of SAP and dropout using Naïve Bayes, Support Vector Machine and Genetic Search Algorithm (GA) for feature optimization. The choice of SVM and Naïve Bayes algorithms for this study is due to their being widely used in the EDM area coupled with their outstanding performances as reported in the literature (Agarwal, Pandey, & Tiwari, 2012; Strecht *et al.*, 2015; Amra & Maghari, 2017; Burman & Som, 2019).

1.3 Aim and Objectives

This study aimed at developing hybrid models for predicting Students' Academic Performance and Dropout using machine learning algorithms.

The specific objectives are to:

- (i) investigate the performances of Support Vector Machine (SVM) and Naïve Bayes (NB) algorithms using real life datasets;
- (ii) design hybrid algorithms for SAP and dropout prediction from SVM and NB;
- (iii) design hybrid algorithms for SAP and dropout prediction from SVM or NB with GA (Genetic Algorithm);
- (iv) implement the hybrid models designed;
- (v) conduct performance evaluation of the developed hybrid models in order to compare them with other existing models.

1.4 Significance of the Study

The need for a more efficient technique of mining educational data cannot be over-emphasized as being perceived by different researchers (Suhirman *et al.*, 2014 and Romero & Ventura, 2010). When a very efficient prediction model is available, it will ease the students' performance classification and help to avoid uncalled-for negative consequences of the wrong classification. Identifying useful patterns in the educational datasets is a very worthwhile exercise because it reveals the consistent and valuable hidden knowledge in these datasets that helps in meaningful decision-making processes by the management of educational institutions. It also facilitates the required guide and supports to students by parents, lecturers/instructors, and other stakeholders in the educational institutions for the advancement of the students and the institutions.

The care for students, being the most important component of every educational institution, demands that their academic progress and development should be central to the institution. It is commonly agreed in an educational institution that when there is no student to train, it means that the existence of other components is bound to vanish. The degree of success or failure of students can however determine their availability or retention in an institution. Any study that will help in this direction will therefore go a long way to benefit institutions on the course of carrying out their main goal of learning and training.

Mining educational data can also play an important role in addressing the issues of students' failure, dropouts, and retention, lack of student monitoring strategies on the part of the educators, lecturers'/instructors' poor teaching methodology, and lack of educational infrastructures in schools.

Specifically, predicting students' academic performance will enhance the sustainability of students throughout their programme of study while identifying the dropout students will also provide earlier alerts for educators to focus on the strategies that will enhance the retention of such students and most times prevent them from becoming spill-overs in the system. All in all, the success of any employed strategies in an institution will bring a lot of stability and improvement to the education system.

1.5 Scope of the Study

This study focuses on the tertiary institutions' students' academic performance models for the design of the hybrid algorithms and the models were purely based on students' academic and demographic features. It does not consider any socio-economic and behavioural features in the modeling of SAP and dropout predictions, although the models can accommodate these features in future researches.

1.6 Definition of Operational Terms

The definitions of some technical terms used in this study are given as follows.

- **Attributes:-** These are set of quantities in a dataset that has a particular (fixed) or predefined value(s) – which may be nominal or numeric in a broader sense. They represent the properties or features of an object (concept) in a dataset. They are always put in columns of a table depicting the values in a dataset.
- **Cross Validation:-** This is a statistical technique of splitting a whole dataset for mining into two where one set is used for training and the other for testing. In cross validation, therefore, the decision is on the fixed number of holds or partitions of the data (Witten and Frank, 2005). Most of the researches in EDM uses 10-fold cross validation or 3-fold cross validation.
- **Dropouts:-** These are categories of students who have been out of their studies without completing their programme due to academic failure or other reasons like financial incapability, sickness, and so on. The focus of this study is on dropout due to academic failure as the only measurable indicator from dropout datasets.
- **Instances:-** An instance is the same as the concept (or object) in a dataset. Each instance is characterized by a set of attributes or features that describe it. In essence, they are always represented as the records in a dataset and are the inputs to a machine learning algorithm in any data mining process.
- **Models:-** A model is a representation of a thing, event, or action. In this study, the expected models are the predictive models obtainable from the application of the various machine learning algorithms to the training dataset.
- **Pedagogical Support:-** This refers to the methods of teaching that are applied in a particular circumstance by an educator for learning to take place. It refers to the nature of teaching

support that is given to a certain category of students by the educators based on the outcome of prediction from the students' data mined.

- **Psychometrics:-** This is a field of study concerned with the theory and technique of psychological measurement. The National Council on Measurement in Education (NCME), simply defined psychometrics as psychological measurement. Generally, it refers to the field in psychology and education that is devoted to testing, measurement, assessment, and related activities.
- **Training Data:-** This is an observed or acquired dataset from an unknown universal dataset for some reasons. It is used to learn the properties and knowledge of the universal dataset. Training sets are assumed independently and identically sampled from the universal dataset (Chao, 2011). This set of data is used by one or more learning algorithms to produce classifiers.
- **Accuracy Paradox:** This is a situation when a high value of accuracy does not correspond to a high-quality model due to the fact that there is a disproportionate relationship between the minority and majority classes as we could have in an imbalanced data (Barros, Neto, Silva, & Guedes, 2019).

CHAPTER TWO

LITERATURE REVIEW

This chapter discusses the various methods of the study through the exposition of the various scholarly efforts that have been put in by other researchers, in order to establish research gaps in the area.

2.1 Data Mining and Knowledge Discovery

Data mining (DM) and Knowledge Discovery in Databases (KDD) are often confused to mean the same thing. This is not so, as the process of discovering knowledge from databases consists of stages of which data mining is one. These stages, according to Han, Kamber, & Pei (2012) are as follows.

1. **Data Cleaning:-** This is the first stage of removing noise and inconsistent data from the original data source.
2. **Data Integration:-** This is the second stage where data from various sources (multiple data sources) may be combined as demanded by the target analysis.
3. **Data Selection:-** This is the stage where data relevant to the analysis task are retrieved from the database.
4. **Data Transformation:-** This is where the selected datasets are formatted or consolidated in the form suitable for mining. This is done by performing summarization or aggregation operations for instance, on the data. Other data transformation operations include normalization of numeric attributes, discretization (that is conversion of a continuous numeric variable into a series of categories with minimal loss of information), conversion of a numeric

variable through either encoding of labels or use of dummy variable and combining levels to reduce the number of levels in categorical variables (Alyahyan & Düşteğör, 2020).

5. **Data Mining:-** Here, intelligent methods or algorithms are applied to the transformed data to extract useful data patterns.
6. **Pattern Evaluation:-** This is the stage where interesting patterns representing knowledge; based on some chosen measures are identified.
7. **Knowledge Presentation:-** This is where visualization and knowledge representation techniques are used to present mined knowledge to the user.

Stages 1 through 4 are regarded as data pre-processing stages in KDD.

The field of data mining evolved because data are being created every second from several transactions of life which are being added to the existing ones and the data in these databases are created and stored for a particular purpose which the organization focuses their use on; they, however, do not think of making use of those data for other viable purposes that may be hidden in them from the users. Examples of these databases are the staff payroll databases, environmental databases, students' academic databases, agricultural databases, medical and biological databases, sales analysis/business databases to mention a few. It had been found out that further analyses of these idle data in the databases can reveal other useful knowledge which can contribute to the advancement of the organization. The field of human endeavour that salvage man from losing all this hidden but important knowledge in the databases is called the Knowledge Discovery in Databases (KDD). What then does this knowledge discovery from databases (KDD) and Data Mining (DM) mean?

Knowledge Discovery in Databases (KDD) according to Han *et al.* (2012) consists of an interactive sequence of steps, namely: data cleaning, data integration, data selection, data transformation, data

mining, pattern evaluation, and knowledge presentation. The most important of these stages is the data mining stage which is a research field upon which knowledge discovery largely depends.

Data mining on the other hand is the process of finding interesting patterns (or knowledge) from a large amount of data stored in databases (Han & Kamber, 2006). It is defined as a non-trivial extraction of implicit, formerly hidden, and potentially useful information from large amounts of data (Gorunescu, 2011). Data mining is therefore an experimental science that involves digging into a database to extract potentially useful knowledge which is previously unknown to the system where it is being generated. The information extracted from the process is then used in making important decisions in the users designated environment. Data mining uses machine learning, statistics and visualization techniques to actualize these goals.

Data mining according to Han & Kamber (2006) and Gorunescu (2011) are of two types; there is descriptive data mining (for example as in clustering and association rule) and predictive data mining (for instance, classification and regression). Descriptive data mining is the mining of data that focuses on bringing out previously unknown patterns in a dataset in a descriptive form using descriptive statistics and so on; a good example of descriptive data mining can be text data mining. Predictive data mining focuses on mining interesting patterns in a dataset through the generation of predictive models, an example is the prediction of students' academic performance.

Data mining has been applied to various problems solving areas of life such as seed and plant selection in Agriculture, Financial Data Analysis, Supermarket Basket Analysis, Customer Relationship Management, Criminal Investigation, Disease Diagnosis, and selecting embryos for artificial insemination among others (Han & Kamber (2006); Witten & Frank (2005)).

2.2 Data Mining Methods

There are four broad methods employed in data mining. These methods can be applied in certain problem cases with specialized approaches matching the choice of method. They are classification, prediction, clustering, and relationship mining. Brief discussions on each of these methods are offered as follows.

I Classification

Classification is the placing of a specific object in a predefined set of categories, based on similarity in object properties (Yassein, Helali & Mohomad, 2017). This is to say that in classification, objects are classified, categorized, or labeled based on predefined parameters. This is what makes classification be referred to as a supervised learning method.

It is the most commonly applied data mining method that employs a set of pre-classified examples from a dataset to develop a model that fits a whole population of records in a large dataset. The goal of classification is to classify a particular dataset into the predefined classes or groups as demanded by the client or the miner.

According to Gorunescu (2011), there are four fundamental components upon which the classification process is based. These are:

(a) **Class:-** This is a categorical variable representing the ‘label’ of an object after its classification.

It depends solely on the predictors for its value.

(b) **Predictors:-** These are the independent variables of the model, they represent the attributes or features of the data to be classified. The final classification is made based on them.

(c) **Training dataset:-** This is the set of data used for ‘training’ the model to recognize the appropriate class based on available predictors. It comprises values from the class and the predictors. A classifier or model is always constructed after the training of a dataset.

(d) **Testing dataset:-** These are new set of data that will be classified by the model (classifier) constructed in (c) after training to evaluate the classification accuracy (that is, model performance).

In essence, classification problems can be stated as follows.

*“Given a set of training data points along with associated training labels,
determine the class label for an unlabeled test instance”* (Aggarwal, 2014)

This formulation gives all classification algorithms two typical phases, namely: the ‘training phase’ – where a model is constructed from the trained instances and, ‘testing’ phase – where the model is used to assign a label to an unlabeled test instance. These two phases are depicted in Figures 2.1(a) and 2.1(b) respectively.

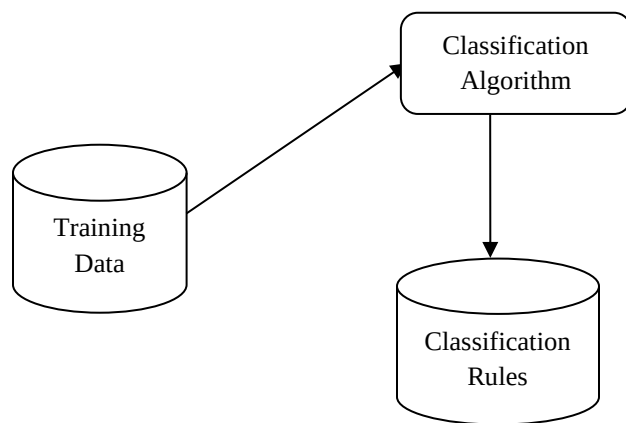


Figure 2.1(a): Learning

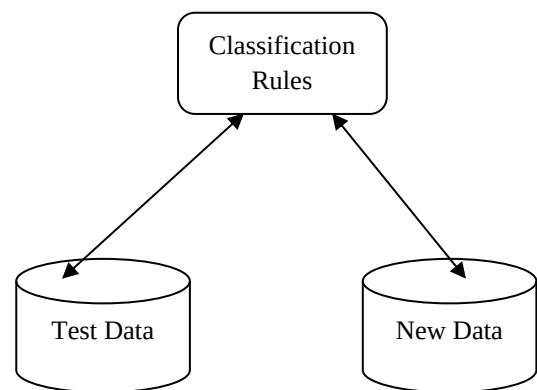


Figure 2.1(b): Classification

Adapted from Han & Kamber (2006)

In Figure 2.1(a) the training data are analyzed by a classification algorithm and the learned model or classifier is represented in the form of classification rules while in Figure 2.1(b) the test data are used to estimate the accuracy of the classification rules. If the accuracy is considered acceptable, these rules can be applied to the classification of new data tuples.

There are many classification algorithms used in data mining, among which are: the Decision Trees (DT), the Naïve Bayes Classifier (NBC), Artificial Neural Network (ANN), and the Support Vector Machine (SVM).

II Prediction

Prediction is very similar to classification in that classification also attempts to make a prediction, however, prediction entails the identification of distribution trends in the dataset available (Han *et al.*, 2012).

When making a prediction, the value of an attribute to be predicted is always not known and that is why it is referred to as the response variable to independent variables or attributes that are already known. That is to say that it is the response variable that is forecasted in prediction.

Complex techniques like logistic regression, decision trees, and neural networks are used in forecasting future values in many real-world problems because they are simply predictions. Some algorithms can handle both classification and regression problems such as the CART (Classification and Regression Trees) algorithm (Suhirman *et al.*, 2014).

III Clustering

Clustering can be said to be the identification of similar classes of objects. It is a descriptive and unsupervised learning method. One can identify dense and sparse regions in object space and discover overall distribution patterns and correlations among data attributes (Suhirman *et al.*, 2014 and Han *et al.*, 2012). Clustering is sometimes used as a preprocessing task for attribute selection and classification.

Cluster analysis helps to identify outliers (that is the objects that are far away from the groupings) and these outliers may be an interesting discovery in a particular data mining task.

According to Han *et al.* (2012) clustering is also called data segmentation in some applications simply because it partitions large datasets into groups based on their similarity.

There are several clustering algorithms used in data mining, among them are the k-means and Expectation Maximization (EM) algorithms.

It is important to state that many of the clustering algorithms can be categorized as classifiers because what is achieved at the end of clustering is a kind of classification. The main difference between clustering and classification is that clustering does not involve predefined class labels.

IV Relationship Mining

Relationship Mining is used to discover the relationship between variables in a dataset with a large number of variables. There are four types of relationship mining, namely: association rule mining, correlation mining, sequential pattern mining, and causal data mining. The most common of the four is association rule mining.

In association rule mining, association rule tends to determine which parameters (or attributes) in a dataset are closely associated or related to each other and what is the existing relationship between them. In other words, the association rule can be said to be a form of correlation finding technique that could be used to expose the relationship between variables in a set of data for mining (Suhirman *et al.*, 2014). A very good example of an association rule algorithm is the Apriori algorithm. According to Baker & Inventado (2017), the goal of association rule mining is to find the if-then rules such that if a variable is found it implied a particular value of the other variable. Correlation mining seeks to find the positive or negative correlation of one variable with another in a given dataset.

In sequential pattern mining, the target is to find the temporal association between variables, which is revealing the trend in the association among different variables of a dataset.

In causal data mining, on the other hand, the focus is on the search for whether an unobserved event is a cause for another observed event in a pool of data.

2.3 Feature Selection Techniques

The use of the wrapper feature selection technique for selecting relevant attributes is part of the methodology of this study. Wrapper feature selection is used because the models being proposed focus on the application of the learning algorithms under study which is supported by wrapper feature selection.

Feature selection or attribute selection is the process of selecting a subset of relevant features (variables or predictors) from the original feature set without any form of transformation and maintaining the physical meaning of the original features (Tang, Alelyani, & Liu, 2014).

There are many reasons why feature selections are done, these includes:

- i. Simplification of models for easy interpretation by users.
- ii. Shorter training times and improved learning performance.
- iii. Avoidance of curse of dimensionality thereby decreasing the required storage.
- iv. Enhancement of generalization by reducing overfitting.

In feature selection, the central focus is to identify features that are relevant or redundant so as to decide whether to retain such attributes or not. There are two basic activities in a feature selection process, these are subset generation and subset evaluation as vividly shown in Figure 2.2.

The subset generation stage of a feature selection process entails the creation of a group of features that are put forward to the best feature subset strategy for necessary evaluation. If the subset of features so generated meet the condition of goodness, the said feature subset is validated, otherwise

the iteration goes back to create another subset of features that is subjected to the same condition. This continues until a satisfactory feature subset is realized and validated.

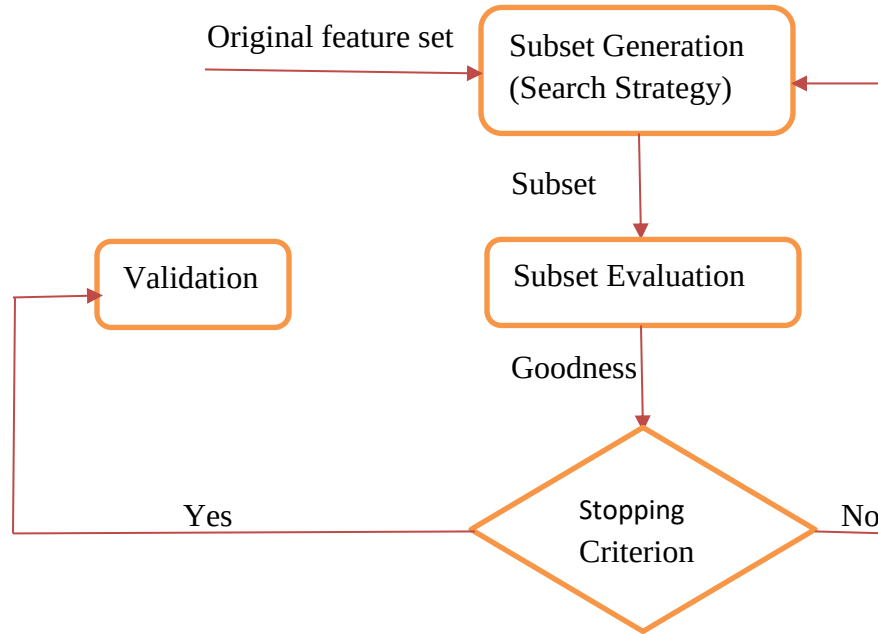


Figure 2.2: Feature Selection Process (Adapted from Homsapaya & Sornil, 2017)

According to (Tang *et al.*, 2014), feature selection algorithms can be categorized into three broad types, namely: supervised, unsupervised and semi-supervised. The supervised feature selection methods are again categorized into filter models, wrapper models, and the embedded models. These three supervised types are discussed below to give an insight into the one to be used in this study.

2.3.1 Filter Models

The filter model separates feature selection from classifier learning and as a preprocessing step; so that the bias of a learning algorithm does not interact with the bias of the feature selection algorithm. It relies on measures of general characteristics of the training data such as distance, consistency, dependency, information, and correlation (Tang *et al.*, 2014).

2.3.2 Wrapper Models

The major disadvantage of the filter approach is that it ignores the effect of the selected feature subset on the performance of the learning algorithm. The wrapper model uses the predictive accuracy of a chosen machine learning algorithm to determine the quality of selected features.

Given a predefined classifier, a typical wrapper model will perform the following steps:

Step 1: Searching a subset of features.

Step 2: Evaluating the selected subset of features by the performance of the classifier, and

Step 3: Repeating Steps 1 and 2 until the desired quality is reached (Tang *et al.*, 2014).

In the wrapper model, the classifier is considered a perfect black box (Guyon & Elisseeff, 2003).

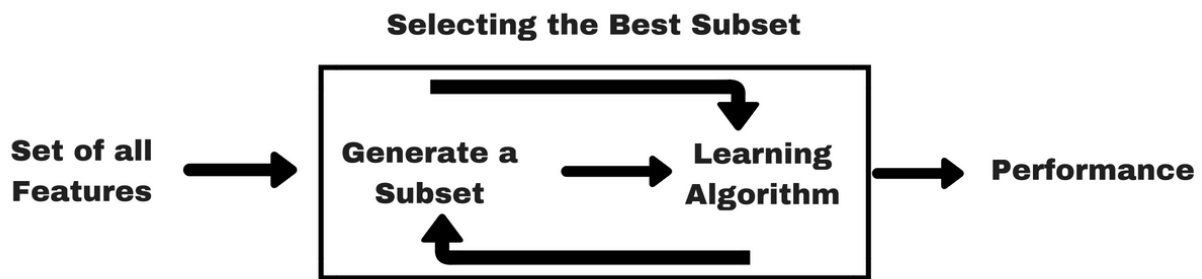


Figure 2.3: Wrapper feature selection process (Adapted from Kaushik, 2016)

The simplified approaches used in wrapper feature selection as shown in Figure 2.3 can either be ‘Forward Selection’ or ‘Backward Elimination’. The shortcoming of this method is that it is very expensive to run for a dataset with a large number of features.

2.3.2.1 Recursive Feature Elimination (RFE)

Recursive Feature Elimination (RFE) technique is a wrapper feature selection technique that fits a model and removes the weakest feature (or features) until the specified number of features is reached. In this study, the underline model is either the SVM or Naïve Bayes algorithm.

The RFE algorithm ranks all the features according to their importance with respect to the particular data mining algorithm that acts as a baseline method. It is a type of backward elimination method that works on a feature ranking system.

Given an external estimator that assigns weights to features (e.g., the coefficients of a linear model), the goal of Recursive Feature Elimination (RFE) is to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on the initial set of features and the importance of each feature is obtained either through a 'coef_' attribute or through a 'feature_importances_' attribute. Then, the least important features are pruned from the current set of features. That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached (Scikit-learn, 2019).

This technique is employed for the hybrid models proposed in this study by using the best of Support Vector Machines (SVM) or Naïve Bayes (NB) algorithm at the base of the wrapper method for feature selection on the SAP and dropout datasets depending on the one that emerged best in the Auto-Classifer Detection stage.

The major advantage of the RFE technique is that it allows the influence of the base algorithm on the selection procedures and the major drawback is the long time it takes to do feature selection since it trains and tested on the model every time a feature is to be selected. It is therefore not advisable for use on a high-dimensional dataset.

RFE is being chosen among all the wrapper methods because of its greedy optimization nature in finding the best performing feature subset (Kaushik, 2016).

2.3.3 Embedded Model

According to Tang *et al.* (2014), the embedded model was proposed to bridge the gap between the filter model and the wrapper model by incorporating statistical criteria as the filter model does, to

select several candidate feature subsets with a given cardinality and by chosen the subset with the highest classification accuracy as in wrapper models.

2.4 Educational Data Mining

Educational Data Mining (EDM) has been expressed as the convergence of methods from several fields which include data mining, machine learning, statistics, visualization, psychometrics, and computational modeling (Baker & Yacef, 2009). EDM is an emerging field from the main research field of data mining which helps in resuscitating several data repositories that are lying fallow in different educational institutions due to the high level of computerization of organizational functions.

The main thing being done in EDM is data mining; it is the dataset being worked upon that differentiates it from other aspects of general data mining.

Educational Data Mining (EDM) being an emerging area of research from the broader Data Mining (DM) researches have unique approaches, these approaches are derived from the combination of data mining methods with techniques in psychometrics.

The data mining approaches applied to educational data as proposed by Baker (2011) ranges from prediction, clustering, relationship mining, discovery with models, and distillation of data for human judgement. These approaches had been employed in so many studies like the studies of Chopra & Lal (2012) and Upadhyay & Katiyar (2014). A brief explanation of each of these approaches is as follows.

a) Prediction:- This is exactly as being discussed under general data mining methods. However, it is important to emphasize here that this approach is useful to the EDM researchers in two ways, namely: to predict the outcome of a data mining process and to study the influence of some

attributes (predictors) on the final educational outcomes (this is what is referred to as reverse engineering by Andonie & Kovalerchuk (2004).

b) Clustering:- This method is also common in general data mining tasks. The goal of clustering is to find data points that are naturally belonging to the same group and making the whole dataset to be in groups of similar concepts. EDM researchers mostly employ this method in the groupings of students based on several factors, for instance, learning behaviours, performance status, and so on. A very good example is in the study of Asif, Merceron, *et al.* (2017).

c) Relationship Mining:- This is also a general data mining method. It is used in EDM to discover the relationship between variables or attributes in a large educational dataset. In addition to the explanation given earlier, it is important to note that relationships found through relationship mining must satisfy two criteria of statistical significance and interestingness (Baker, 2011). This is to strengthen the level of confidence in the found relationship and as well to determine which findings are most distinctive and well-supported by the data.

d) Discovery with models:- This is an approach peculiar to educational data mining and it is a process of generating new models or findings from a particular model. This is mostly done by using psychometric principles to bring out more findings from the resultant data mining models in an EDM process.

e) Distillation of Data for Human Judgement:- This is also very specific to EDM researches. According to Baker (2011), data is distilled for two key purposes; for identification and classification. When data is distilled for identification, information visualization tools are employed to present the education data to allow easy identification of well-known patterns from the data by human beings. However, when data is distilled for classification, certain parts of a

dataset are displayed in visual or text format and labeled by human coders. It is these labels that are later used as the basis for the development of a prediction model.

2.5 Students' Academic Performance Prediction

Predicting students' academic performances has been the preoccupation of interesting research fields such as Educational Psychology (EP), Learning Analytics (LA), and the field of Educational Data Mining (EDM).

Prediction of students' academic performances based on their previous records of performances is a very important and usual thing for an academic institution to provide adequately for its students. These predictions are done with the aid of psychometrics in psychology, analytic tools in learning analytics, and the use of data mining techniques in EDM.

2.5.1 Factors Affecting Students' Academic Performance from Educational Psychology Perspectives

Researches in Educational Psychology (EP) had greatly helped the EDM researchers in generating variables for study when mining education data. These variables are the factors responsible for students' academic performances. Some of them as exposed in the EP literature, are presented as follows.

Hijazi & Raza-Naqvi (2006) agreed with Tinto (2006) that student performance is the product of socio-economic, psychological, and environmental factors. These factors were expatiated by others as classroom management, students motivation, a balance between moral and social development of a student, students' performance monitoring strategies with emphasis on their different nature, parents involvement and comprehensive assessment (Seifert & Sutton, 2009), family stress, communication, learning facilities and proper guidance by both parents and teachers

(Mushtaq & Khan, 2012), age, guardian, father's socio-economic status and daily study hours (Ali, Haider, Munir, Khan, & Ahmed, 2013), students attendance in classes (Aden, Yahye, & Dahir, 2013), prior academic success, incentives and expectations (Nayebzadeh, Dehnavi, Nejad, Mir, & Sadrabadi, 2013), truancy and physical environment were stated in Fareo (2013), availability of teaching resources, teacher quality and performance, socio-economic status, entry grade, motivation, and attitudes were presented in Enu, Agyman, & Nkum (2015) and a host of other factors.

2.5.2 Students' Academic Performance Datasets

SAP and dropout datasets are either purely institution-based or traditional survey-based depending on the mode of acquisition. The datasets used in SAP and dropout studies reviewed in this study are from any of the three data sources, namely: the students' record system, the learning management system, and the traditional survey. The students' record system and learning management system fall under the institution-based data sources.

The students' record system is private to the individual institution and it contains the direct data values relating to students' academic performance within the institution databases. The Learning Management System (LMS) is a data source that allows the extraction of software-generated data values relating to students' performance from the system. Data values of attributes that are connected to students' academic performance are mostly obtained from the students using fact-finding tools like the questionnaire and the interview. Even though some facts to be obtained from students using traditional survey can also be acquired from students' profiles as entered by them at the beginning of a session, the specialized survey can still be carried out to obtain such data in case they are not available in the students' profiles.

SAP dataset is mostly privately available and is rarely made public. There are studies among the reviewed ones that made use of the three data sources for the SAP dataset such as the study of Adejo & Connolly (2018). Some others made use of the LMS data source for their dataset like the study of Romero *et al.* (2008). The students' record system that usually contained the exact data values indicating the academic performance of the students in every institution is a widely used data source for all SAP and dropout studies. Datasets used for SAP and dropout studies can be large if the researcher considered obtaining several years data and they can contain several features depending on the objective of the research.

2.5.3 Taxonomy Based on Features

The existing SAP and dropout studies employed diverse features. A taxonomy of these features as presented in Figure 2.4 shows that the features can be classified into either internal features of students in the case study or external features linked to the students owing to their relationship with their environments. The internal features can be divided further into two, namely: personal and psychological features. The external features are also classified into four categories as academic features, demographic features, economic features, and social features.

1. Personal features

These are features that are directly reflecting the characteristics of the students in the case studies without any link to the students' outside world. They are features that are directly describing the students. Features like age, average study hour, disability status, health status, sex/gender are included in this category. Almost all the reviewed studies on SAP and dropout used some of these variables except the studies of Ogor (2007); Ogunde & Ajibade (2014) and Hamsa, Indiradevi & Kizhakethottam (2016) that did not use them.

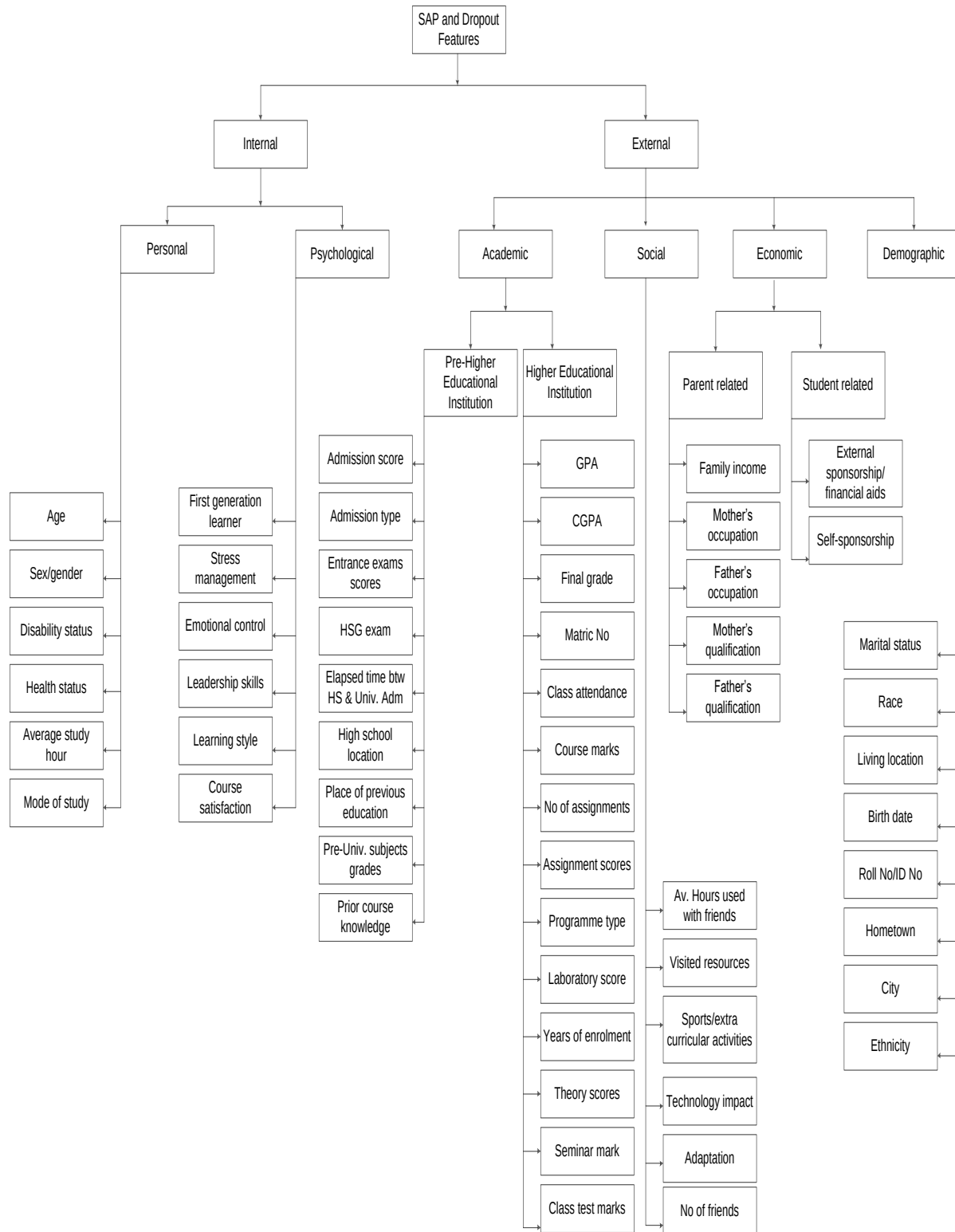


Figure 2.4: Taxonomy based on features

2. Psychological features

These are features that describe the mental development of the students used in a case study in the SAP and dropout researches. They are mostly emphasized in the dropout studies because of their direct effect on the non-academic dropout rate. They include course satisfaction, first-generation learner, learning style, perceived academic emotion, perceived academic control, and stress management ability among others. These features were used in the studies of Respondek, Seufert, Stupnisky & Nett (2017); Ruby & David (2015) and Mishra, Kumar & Gupta (2014).

3. Academic features

Features in the datasets that are purely related to the academic activities of the students in the case study are referred to as academic features. They are classified into the pre-Higher Educational Institution (pHEI) academic features and the Higher Educational Institution (HEI) academic features.

a) Pre-Higher Educational Institution (pHEI) academic features

These are features obtained from the students due to their activities before their entrance into the Higher Educational Institutions. They include the admission score, admission type, entrance exams scores, Higher School Grade (HSG) exam, place of previous education, pre-university subjects' grades or scores, and so on. SAP studies that tend to predict the student's final grade on graduation at the university using the pre-university data or features make use of these features often. These features were used in the studies of Ahmad, Ismail & Aziz (2015); Ogunde & Ajibade (2014); Kabakchieva (2012), and Dekker *et al.* (2009).

b) Higher Educational Institution (HEI) academic features

These are features that are obtained from the academic activities of students while in the higher educational institution. They are the major set of features used in SAP and dropout studies as both

the predictors and the predicted variables because of their great importance. Final grade at graduation, Grade Point Average (GPA), Cumulative GPA (CGPA), Matric No, course marks, course grades, class test marks, assignment scores, class attendance, laboratory scores, theory scores, mode of study, seminar marks are among the commonly used features in this category. Every SAP and dropout study must use some of these features most especially attributes like the GPA, the CGPA, and the final grade as the class variable. Studies like Rovira *et al.* (2017), Al-barrak & Al-razgan (2016), and Osofisan *et al.* (2014) had exclusively limited themselves to the use of these features for the prediction of students' academic performance.

4. Demographic features

These features are demographic data related to students in the case study as it affects their academic performance. They include marital status, race, nationality, living location, date of birth, ethnicity, hometown, city, and so on. These features were used to predict the students' academic performance in the works of Aulck, Velagapudi, Blumenstock & West (2016); Kaur & Singh (2016); Lakkaraju *et al.* (2015), and Strecht *et al.* (2015) among others.

5. Economic features

The economic features are those features that indicate the economic status and support that are available to students in a case study of any SAP and dropout research. It can be classified into two: parent-related economic features and student-related economic features.

a) Parent-related Economic features

These are features that are related to the students' economic status through their parents. These include family income or family support, parent's economic status which depends on the mother's occupation and father's occupation as separate attributes, parent's education which is broken into

the mother's qualification and father's qualification. Studies that had used these features include Adejo & Connolly (2018); Saa (2016); Aziz *et al.* (2013) and Pal (2012).

b) Student-related Economic features

These are features that describe the nature of the economic support available to students through other means apart from their parents. It includes sponsorship or financial aid from third parties and self-sponsorship. These features were employed in Adejo & Connolly (2018) and Gilbert (2017).

6. Social features

Features that describe the relationship of the students in the case study with others in their environment, most especially as it affects their academic performances in the higher institution are called the social features. The social features employed in the existing literature include the number of friends, average hours used with friends, sporting and extra-curricular activities, visited resources, technology impact, adaptation, and so on. These features were used in Adejo & Connolly (2018), Amrieh, Hamtini & Aljarah (2016), and Ruby & David (2015).

The list of these features is inexhaustive, however, the taxonomy presented here will assist future researchers in SAP and dropout predictions to classify future attributes embedded in the study of SAP and dropout prediction.

When the attributes so far used in SAP and dropout studies were considered as it is being presented in the taxonomy of features in Figure 2.4, it was discovered that the Grade Point Average (GPA) is always being used as the dependent variable in all the studies while the predicting variables continue to vary based on different studies' priority.

2.5.4 Taxonomy of Methods Used for SAP and Dropout Predictions

This section presented a classification of the various methods used for SAP and dropout predictions in the literature with emphasis on the data mining classifiers and the examples belonging to each classification discussed. SAP predictions have been one of the research focuses of the four fields of human endeavour, namely: Educational Psychology, Learning Analytics, Survival Analysis, and Data Mining as can be seen in Figure 2.5.

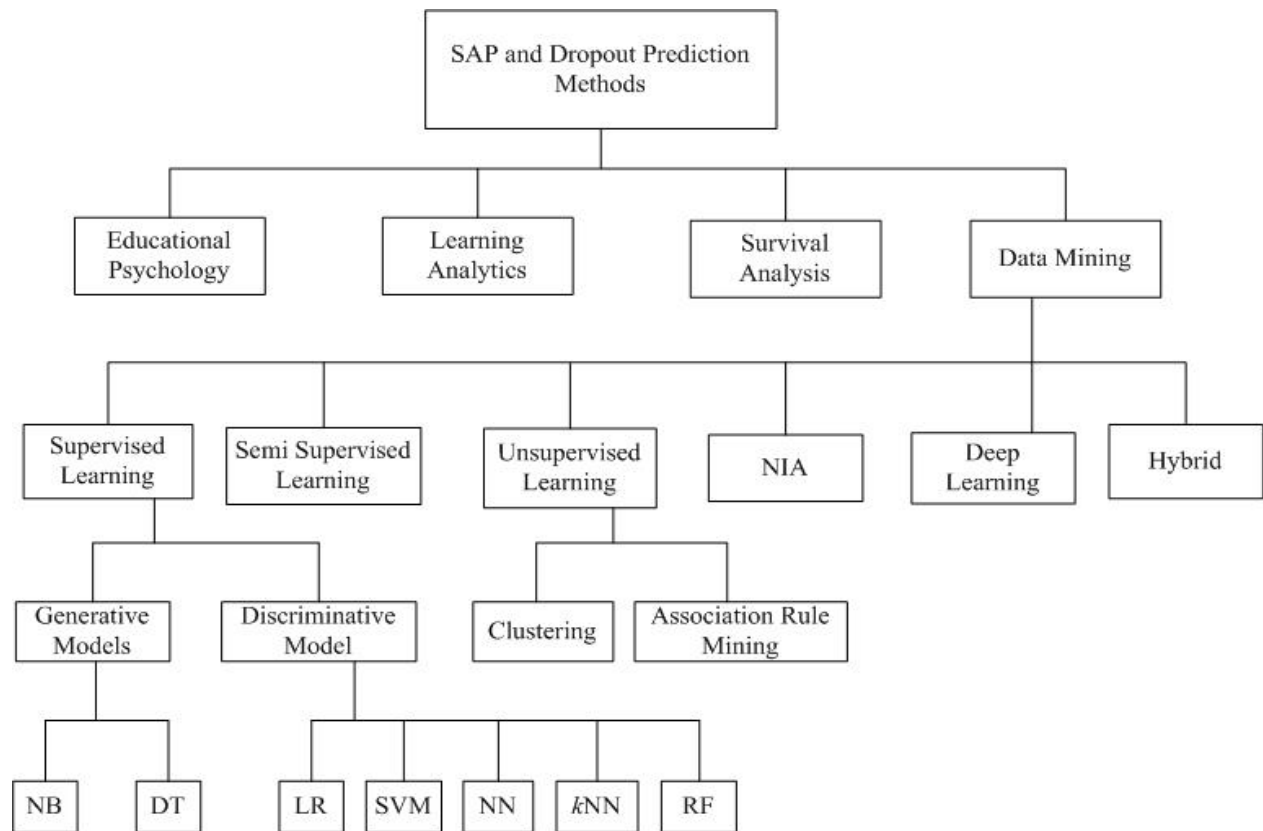


Figure 2.5: Taxonomy of methods for SAP and dropout prediction

In the Educational Psychology field, researchers postulate theories relating to the effects of certain psychological factors on the final outcome of student performance, a good example of the studies in this area is the study of Respondek *et al.* (2017) which investigated the effect of perceived academic control and academic emotion on dropout intention and students' academic achievements.

Learning Analytics (LA) is defined as the measurement, collection, analysis, and reporting of data about learners and their contexts, for understanding and optimizing learning and the environments in which learning occurs (LAK, 2011). It takes a deeper step in analyzing students' performance in learning using a host of techniques when compared to the adopted method by educational psychologists. The techniques used in LA include Statistics, Business Intelligence, Web Analytics, Operational Research, Artificial Intelligence, Social Network Analysis, and Information Visualization. There are several studies in this area, however, the works of Daud *et al.* (2017) which used advanced learning analytics to predict student performance with experiments conducted using 776 students records from the University of Pakistan, and Lu *et al.* (2018) that applied Learning Analytics for early prediction of students' final grades in a blended Calculus course.

The Survival Analysis (SA) approach is an approach that involves the application of a collection of statistical methods that contain the time of a particular event of interest as the outcome variable to be estimated (Ameri, Fard, Chinnam, & Reddy, 2016). In this approach, there are three fundamental terminologies, namely: the censoring, the survival function, and the hazard function. In the context of students' performance prediction, censoring refers to when a student dropout and had failed during the study period. The survival function is the probability that a student will perform well during a specified time, t , while the hazard function is the instantaneous probability for the event to occur given that the student had performed well in time, t (Khamis *et al.*, 2017). Among the various studies that use this approach are the studies of Ameri *et al.* (2016) which applied the time-dependent Cox proportional hazard model to predict student dropouts. The survival analysis approach was also employed in the study of Khamis *et al.* (2017) which compared four survival analysis models, the Kaplan Meier analysis, and three parametric survival analysis

models, namely: the exponential model, Weibull model, and the Gamma model. All these four SA models were applied to students' enrolment data of a university in Malaysia to predict the student's performance. It was discovered that the Weibull model outperformed the Kaplan Meier model in the study. Patacsil (2020) applied a combination of a survival analysis technique and ensemble models on 2,401 students' enrollment data of Pangasinan State University for early prediction of student dropout. The highest accuracy was obtained when bagging is applied to the J48 algorithm in comparison with other tree-based algorithms, although Random Forest had the highest dropout precision and graduated recall values.

Data Mining (DM) contributed its quota to SAP and dropout predictions in a newly emerging field of Educational Data Mining. The DM approaches employed so far can be classified into three, namely: the supervised learning approach, semi-supervised, and the unsupervised learning approach. Nature-Inspired Algorithm (NIA) were also combined with some of these DM techniques for better optimization. Deep learning had also been used to predict SAP and dropout, and various hybrid methods that offer opportunities to combine one or more DM algorithms had also been explored to predict students' performance even though scanty at present.

A supervised learning approach is one in which the class labels are known from the datasets employed. The techniques or methods used in predicting SAP and dropout that is supervised learning are either generative or discriminative models (Ameen *et al.*, 2019). A generative model is a statistical model of the joint probability distribution $p(x, y)$ on input x and output y while a discriminative model is a statistical model that defines the conditional probability $P(y|x)$ of output y given input x (Andrew & Michael, 2001). Discriminative models, unlike the generative model, solve classification problems directly rather than solving a more general problem as an intermediate step (Nivre, 2003).

Naïve Bayes (NB) and Decision Trees (DT) are both examples of generative models commonly used while the Support Vector Machine (SVM), Logistic Regression (LR), Neural Network (NN), k-Nearest Neighbor (kNN) and Random Forest (RF) are examples of commonly used discriminative models as earlier presented in Ameen *et al.* (2019). The specific data mining methods that are being used in SAP and dropout predictions in each category of the two supervised learning models are further discussed in the subsequent sections.

2.5.4.1 Naïve Bayes (NB)

Naïve Bayes classifier is based on the Bayes' probability theory which assumes that all attributes of a given class in a dataset are independent. This means that it assumes that they all contribute equally to the outcome of the classification task performed on the dataset. This is what is termed the strong (naïve) independence assumptions. Naïve Bayes classifier is a descriptive and predictive approach to predict the class membership for a target tuple (Pandey & Pal, 2011).

The Naïve Bayes Classifier works as follows:

- i. Given a training set of tuples A and their associated class labels $C_1, C_2, \dots, C_m$. If X represents each of the attributes in A which is to be classified.
- ii. We find the probability of each of the classes, $P(C_i)$ – class prior probabilities.
- iii. Compute the conditional probabilities for each of the components of X with respect to the classes C_i , that is $P(X|C_i)$.
- iv. Using the probabilities in (iii) above, we obtain the posterior probabilities $P(C_i|X)$ for each of the classes. By Bayes theorem; this is given as:

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \dots\dots\dots 2.1$$

where $P(X)$ is the prior probability of each of the components of X.

- v. The Naïve Bayes prediction will be in favour of the class whose posterior probability $P(X|C_i)P(C_i)$ is maximum since $P(X)$ is constant for all classes. The classifier predicts that the label of tuple X is the class C_i if and only if

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \quad \text{for } 1 \leq j \leq m, j \neq i, \dots, 2.2$$

Hence, it is called the maximum posteriori probability (Han *et al.*, 2012).

The strengths and weaknesses of the Naïve Bayes algorithm as reported in the literature are as follows.

Strengths:

1. Simple, fast, and very effective.
2. Resilience to noise and missing data.
3. Require relatively few samples for training.
4. Easy to update on the arrival of new data.
5. Not ideal for large datasets.

Weaknesses:

1. It relies on an often faulty assumption of equally important and independent features.
2. Estimated probabilities are less reliable than the predicted classes and therefore there are chances of loss of accuracy.
3. Zero frequency can lead to difficult predictions. (Dugong, 2011)

Among the studies that applied the Naïve Bayes method in predicting SAP and dropout is the work of Pal (2012) which used the Naïve Bayes method to manage the students' dropout rate in an engineering college.

Aziz *et al.* (2013) is another study that uses a Naïve Bayes classifier, it presented a framework for predicting the performance of the first-year students in a Computer Science course using merged data from two academic databases of a university. The presented framework consists of four stages,

namely: data collection, data transformation, pattern extraction, and prototype development. The output of the pattern extraction stage was used to determine the last stage of prototype development. The study determined whether the parameters used contributed to the student's academic performance or not and also found out the relationship between independent parameters and the dependent parameter, that is GPA.

Aziz, Ismail, Ahmad, & Hassan (2015) also applied Naïve Bayes classifier (NBC) on an integrated dataset extracted from the students' academic databases of the Faculty of Information and Communication (FIC), Unisza, and the Student Entry Management Database (SEMD) to analyze the academic performance of students. The study uses GPA as the dependent variable with labels: Poor, Average, and Good to predict the students' performance using other five independent attributes. Their results showed that the NBC had the highest accuracy of 57.4% when 3 fold cross validation was used and the most influential predictor of students' performance among the five independent parameters used was family income with 56.8% probability. The study discovered that at 3-fold cross validation; the predictive model gives better classification for the 'Average' students' category and failed to predict the poor student category.

Kaur, Singh, & Josan (2015) applied the Naïve Bayes algorithm together with other algorithms like MLP, SMO, J48 to predict slow learners among students and displaying it based on predictive data mining models constructed using classification algorithms. Dataset from a high school was used and tested with various classification algorithms (that is, Multi-Layer Perceptron, Naïve Bayes, SMO, J48, and REPTree). Performance measures such as TP Rate, FP Rate, precision, recall, F-measure, ROC area, and accuracy were used to test and validate the resulting models. It was observed that the Multi-Layer Perceptron outperforms all other classifiers in the study with an accuracy of 75%.

Ahmad *et al.* (2015) used NB, DT, and Rule-based algorithms to classify students' academic performance with a few to compare the three methods. A new feature of learning behavioural features was introduced by Amrieh *et al.* (2016) to predict students' academic performance using Naïve Bayesian, ANN, and Decision Tree. The study applied ensemble methods of Bagging, Boosting, and Random Forest to improve the performance of the three classifiers. The dataset used was filtered using a filter feature selection method to extract the most important ranked features from the database of Kalboard of LMS. The experimental result of their study shows that the accuracy, precision, recall, and F-measure observed for ANN were high with and without the behavioural features and when tested and validated the measures were all high in values for the prediction models.

Kaur & Singh (2016) applied Naïve Bayes and J48 decision tree algorithms on a set of dummy data containing nine attributes. Their results showed that the Naïve Bayes classifier (with a predictive accuracy of 63.59%) outperformed the J48 algorithm (with 61.53% accuracy). The comparisons between the techniques were mainly based on the predictive accuracy of the classifiers.

Mueen, Zafar & Manzoor (2016) also applied three different classification algorithms, Naïve Bayes, Neural Network, and Decision Tree, to undergraduate students' data. Their results showed that the Naïve Bayes algorithm outperformed the other two algorithms with the highest predictive accuracy of 86%.

Asif, Merceron *et al.* (2017) used a host of DM algorithms with the inclusion of Naïve Bayes and added a pragmatic policy to analyze the undergraduate students' achievements at the end of a four-year study programme and the typical progressions of these students. The application of the two methods was combined to arrive at two groups of students (cohorts 1 and 2) of low and high-

achieving students. The study combined three approaches of educational data mining namely: prediction, clustering, and distillation of data for human judgment. The prediction approach used ten classifiers, six variants of decision tree algorithms, Neural Networks, Naïve Bayes, a rule-based classifier called the Rule Induction with Information Gain (RI-IG), and I-Nearest Neighbor classifier. X-means algorithm was used for clustering. The distillation of data for human judgement in the study was enhanced by the pragmatic policy on the two cohorts to examine the revealed progressions as in the clusters. It was observed that the pragmatic policy was able to detect the two groups of students in the two cohorts under study.

Makhtar, Nawang, Nor & Shamsuddin (2017) used the Naïve Bayes algorithm with 10 fold cross validation to predict the relationship between subjects that affects students' academic performance in Sijil Pelajaran Malaysia institution. In their results, Naïve Bayes has a predictive accuracy of 80.94% among the four iterations carried out to top other selected classifiers like random tree, nearest neighborhood, multiclass classifier, and conjunctive rule. Naïve Bayes and k-Nearest Neighbor (k-NN) algorithms were also applied to the dataset of secondary general certificate students of Gaza strip by Amra & Maghari (2017). Their results showed that Naïve Bayesian classification had the highest predictive accuracy of 93.17%. Agrawal, Vishwakarma, & Sharma (2017) applied four classifiers, namely: decision tree, random forest, Naïve Bayes and rule induction on students' dataset containing historical records collected online from two schools in Portugal. Their results showed that decision tree outperformed other classifiers at different folds of cross validation of 10, 20, 30, 40, and 50, however, the performances of all the classifiers used including the Naïve Bayes were in the same range and were very close.

Sultana, Khan, & Abbas (2017) employed the Naïve Bayes algorithm with a decision tree, logistic regression, and neural network to predict potential dropouts among electrical engineering and

computer students of two universities in Pakistan using both cognitive and non-cognitive features. The dataset used in the study was collected from students using questionnaires for both cognitive and non-cognitive features. The result of the study revealed that the Naïve Bayes algorithm outperformed all others by achieving a predictive accuracy of 65% in the best combination of the features. However, the study limitation is that the manner of data collection from students in the survey may impact negatively on the results of prediction due to its liability to error.

Another study that used the Naïve Bayes algorithm is the study of Razaque *et al.* (2018) which applied the Naïve Bayes algorithm on the Computing students dataset. The performances of Naïve Bayes on the data clusters labeled C1 to C4 showed the best predictive accuracy of 98.8% in cluster C3.

In the study of Ayyaz, Asif, & Zahoor (2020), the Naïve Bayes algorithm was used together with Random Forest, SVM, and Deep Learning to predict the performance of primary school students. The data collected for the study has 10 features relating to the student, teacher, and the school where the data was gathered. The experiment was carried out using RapidMiner and the results showed that Naïve Bayes had an accuracy of 75% to be the second-best after Deep Learning with 77% accuracy.

Naïve Bayes classifier was also applied in Vamshidharreddy, Saketh, & Gnanajeyaraman (2020) together with Decision Tree (DT) and SVM, all in separate, to a dataset containing some demographic and academic features with a view to predicting students' academic performance. The study reported that NB outperformed the other two techniques by having an accuracy of 77%, SVM (38%) and DT (71%). The limitation of this study is that it considers only the predictive accuracy to determine the best performing classifier though the results presented in the study showed that DT appears the best among the three selected algorithms. This is because the DT

values of Accuracy, Precision, Recall, and F1 score which are 0.71, 0.74, 0.72, and 0.73 respectively are higher in three out of the four evaluation criteria than the NB while the SVM was the least of all.

2.5.4.2 Decision Trees (DT)

Decision trees are widely used data mining techniques. They are built in such a way that they start with an attribute at the root node (first node) from where it is further split into leave(s) and these leaves can also be split into other leaf nodes (internal node with one incoming edge and one or more outgoing edges) based on certain criteria. The tree terminates when the last node in the tree has no outgoing edges. This is to say that a decision tree is a supervised classification technique that builds a top-down tree-like model from a given dataset using the attributes in it. The last node of a decision tree (that is the leaf node) is the predicted class label of a particular instance (Shalev-Shwartz & Ben-David, 2014). All decision tree classifiers have two phases, namely: the growth phase or building phase and the pruning phase. The tree is built in the first phase by splitting the training set based on local optimal criteria in a recursive way until all or most of the records belonging to each of the partitions have the same class label. However, there may be overfitting, the effect of which is eliminated by the second phase called pruning. Pruning improves the tree by removing noise and outliers to increase classification accuracy (Yadav, Bharadwaj & Pal, 2011). There are several variants of decision trees like the ID3 (Interactive Dichotomizer 3), C4.5 (or J48), C5.0, and CART (Classification and Regression Tree).

Studies that have employed various decision tree algorithms in SAP predictions include Ogor (2007) where the C5.0 algorithm - a variant of DT, was applied on students' datasets using the Clementine software to predict students' academic performance. The rules generated by the C5.0 decision tree algorithm were used to develop a prototype software written in Microsoft Visual

Basic programming to monitor and evaluate students' academic performance. Some of the attributes used in the work are AgeClass, CreditAttemptClass, Campus, CreditPassClass, the CGPAClass, and so on. The resulting system was able to predict accurately the students' performance as demonstrated in the visualization results of the predicted CGPA plotted against the actual graduating GPA.

Ogunde and Ajibade (2014) employed the ID3 decision tree algorithm to find the relationship between the final graduation grades of students and their entry results. The study used the data collected from the Academic department of Redeemer's University, Nigeria consisting of sex, student entry grades in secondary school (O'level), entrance examination scores, and grades obtained at graduation (B.Sc) for all graduates between 2008/2009 and 2011/2012 sessions. ID3 algorithm was applied on the dataset using WEKA and the rules generated from the application were used to create the knowledge base part of the prediction system developed in Java. The prediction system was able to predict the final grades of students on graduation even at the point of their entry into the university.

A weighted modified ID3 algorithm developed by Joseph & Devadas (2015) was used to predict students' performance using a dataset obtained from the first batch of students of the Department of Computer Science and Engineering (CSE) at the College of Engineering, Munnar. The results showed that the modified weighted ID3 (with 76% prediction accuracy) outperformed C4.5 (45.83% accuracy), ID3 (52.08% accuracy), and CART (56.25% accuracy). The six attributes used in the study are the previous semester mark, class test mark, assignment performance, attendance in class, lab work, and End Semester mark. The study focused on the test for the efficiency of the modified algorithm.

Yadav *et al.* (2011) applied CART, ID3, and C4.5 – the three commonly used decision trees; on a dataset of 48 students of Masters in Computer Application (MCA) in WEKA mining environment. The results showed that CART has the highest correctly classified instances (56.25% accuracy) among the three classifiers although it has the highest execution time. The work of Yadav & Pal (2012) was similar to that of Yadav *et al.* (2011) in that the same algorithms were implemented but to improve the engineering students' performance. The three DT algorithms used by Yadav *et al.* (2011) were implemented separately with Chi-Square Automatic Interactive Detection (CHAID) algorithm by Saa (2016) to produce predictive models using multiple features of students' performance. The study applied C4.5, ID3, CART, and CHAID algorithms to the dataset used in both RapidMiner and WEKA environments using 10-fold cross-validation to verify and validate the outcomes of the four algorithms. The results of the study show that CART had the best accuracy of 40% among the four classifiers. The study also applied a Naïve Bayes classifier for the sake of comparison, which still performed lower than CHAID, having a predictive accuracy of 36.40%. The study used predictive accuracy as the only measure to compare the algorithms.

Some variants of decision trees together with a variant of ensemble method like Random Forest were employed by Asif, Hina, and Haque (2017) to predict students' graduation performance in their final year. Al-barrak & Al-razgan (2016) predicted the students' final grade (GPA) using the J48 algorithm – a variant of DT by applying the J48 algorithm on King Saud University Computer Science college students of the year 2012 dataset using WEKA mining tool. The results of the mining showed that among all the courses; Software Engineering came out as the most important mandatory course in the determination of students' final grade, it also showed that Java2 course among all programming languages was closely related to the final GPA than Java1. It was found out that the addition of students' grade in the Data Structure course confirmed the influential nature

of Java2 and when the J48 algorithm was run repeatedly for six times, each run for each semester; the resultant tree showed that Java1 is the most influential course for the determination of final GPA.

In Dekker *et al.* (2009), two variants of DT – CART & C4.5 along with other DM algorithms were employed to develop predictive models that predict the Electrical Engineering students' dropouts. Students' performance prediction models were also generated from DT combined with a fuzzy genetic algorithm in Hamsa et al. (2016). The study used the stream of data from Bachelor and Master Degree students in Computer Science and Electronic and Communications. Internal marks, sessional marks, and admission scores were included in the parameters selected for the study. The results revealed that there were more students at risk class when a decision tree is used and the fuzzy genetic algorithm gives more passed students.

Aye (2020) applied J48, and three other DM algorithms on the X-API-Edu-Data gathered from a Learning Management System of the University of Jordan which contains 480 instances and 16 features. The results showed that the J48 algorithm outperformed all others with 75.83% accuracy and the lowest Mean Absolute Error of 0.2155.

2.5.4.3 Logistic Regression (LR)

Logistic regression is a linear model that models categorical response variables. It gives estimates of the probability that a dependent variable will have a given value. It is only used when the output variable of the model is defined as a categorical binary, though the inputs can be quantitative. The conclusion is made based on the final value of the probability of the output after having calculated the regression probability for all inputs. This will make us know the more probable categorical value whether the one with value 0 or the one with value 1 (Kantardizc, 2011).

Logistic regression was used for SAP predictions in studies like that of Agarwal, Pandey, & Tiwari (2012). The authors employed eight classifiers, namely: Logistic Regression (LR), multi-layer perceptron, Support Vector Machine (SVM), RBFNetwork, Voted Perceptron, Single Instruction Multiple Data (SIMD), Winnow, and simple logistic to predict students' performance. In their study, they applied the mentioned classification algorithms on a dataset from a community college database. The dataset has four attributes and 2000 records of students' performance details. The open-source machine learning tool, WEKA, was used to generate models from the dataset. The results showed that the Support Vector Machine is the best classifier among others in the study with maximum accuracy and minimum Root Mean Square Error (RMSE). The comparison was based on predictive accuracy only. LR was also used together with DT and NN in the study of Simeunović & Preradović, (2014) to predict students' success in their studies.

Aulck *et al.* (2016) modeled student dropout using data gathered from the registrar databases of the University of Washington in the USA. Regularized logistic regression, k-nearest neighbour, and Random Forest were applied on a dataset containing 32,538 students' records to predict the binary dropout on the features like race, gender, resident status, GPA, and so on totaling 784 additional features. Their results showed that the strongest prediction was obtainable in predicting eventual student attrition from a balanced dataset of over 32,500 students with regularized logistic regression. The strongest individual predictors of attrition from the study were the GPA in Math, English, elementary, and psychology courses.

The logistic regression technique was also applied by Baker, Berning, Gowda, Zhang, & Hawn (2020) to predict K-12 students' dropouts using a dataset that contains 4,864 student records from a district in Texas. A total of 231 features selected for use in the study were directly from the school district's data warehouse. The results of their experiments revealed that the proposed model

can distinguish a student who would drop out from the one who would not with 0.76 as the AuC value. It was also found out that a broad range of factors are associated with dropout, and in particular found dress code violations as having a strong relationship with dropout.

Sequential LR together with Input-Output Hidden Markov Model (IOHMM) were applied by Mubarak, Cao, & Zhang (2020) to the 2013-2014 academic year dataset as provided by the Open University Learning Analytics (OULA) containing demographic information about students and their behavior with a Virtual Learning Environment (VLE) for early prediction of dropout students in an online course. The models were evaluated using the performance metrics of precision, recall, F-measure, Accuracy, and Area under the ROC curve. The study revealed that students' activities are more interactive in the early weeks of the course than the middle or final week for all activities. It was also discovered that the proposed models outperformed the four baseline methods in terms of precision and accuracy.

2.5.4.4 Support Vector Machines (SVM)

Support Vector Machine (SVM) is an algorithm based on the idea of 'kernel trick', which is a way of solving non-linear separation problems by mapping the original non-linearly separable points into a higher-dimensional space where a linear classifier is subsequently generated for use (Gorunescu, 2011). The problems of computational complexity and overfitting as inherent in other learning algorithms are solved by SVM through the support vectors and maximum margin hyperplane. In other words, it is an algorithm that finds a special kind of linear model, the maximum margin hyperplane which is the hyperplane that gives the greatest separation between the classes in a two-class dataset whose classes are linearly separable (Witten & Frank, 2005). Both classification and regression tasks are supported by SVM and it can handle multiple continuous and categorical variables effectively.

According to Rajeshinigo & Jebamalar (2017), a support vector machine is a classifier that performs classification tasks by constructing hyperplanes (linear models) in a multidimensional space used in separating cases of different class labels. New examples are mapped into the same space and then predicted to belong to a category based on which side of the margin they fall.

An SVM classifier that builds a maximum margin decision hyperplane to separate two classes (squares and circles) is depicted in Figure 2.6.

According to Shalev-Shwartz & Ben-David (2014), when data is linear, it means that the component classes are linearly separable, otherwise, they are non-linearly separable. Hence there are two possible cases as follows.

(a) When the case is linearly separable data, the SVM works as follows.

- Let the dataset D be given as $(X_1, y_1), (X_2, y_2) \dots (X_{(D)}, y_{(D)})$ where X is the set of training tuples with associated class labels $y_i, y_i \in (+1, -1)$.
- The separating hyperplane that has maximum margin is then sought for. It is usually written

$$\text{as } W.X + b = 0 \quad \dots\dots\dots 2.3$$

$$\text{or } w_0 + w_1x_1 + w_2x_2 = 0 \text{ (replacing } b \text{ with } w_0) \quad \dots\dots\dots 2.4$$

where 'w' is a vector containing the weight of the points X and 'b' is the constant.

so that the two separating sides will have tuples satisfying either

$$w_0 + w_1x_1 + w_2x_2 > 0 \quad \dots\dots\dots 2.5$$

$$\text{or } w_0 + w_1x_1 + w_2x_2 < 0 \quad \dots\dots\dots 2.6$$

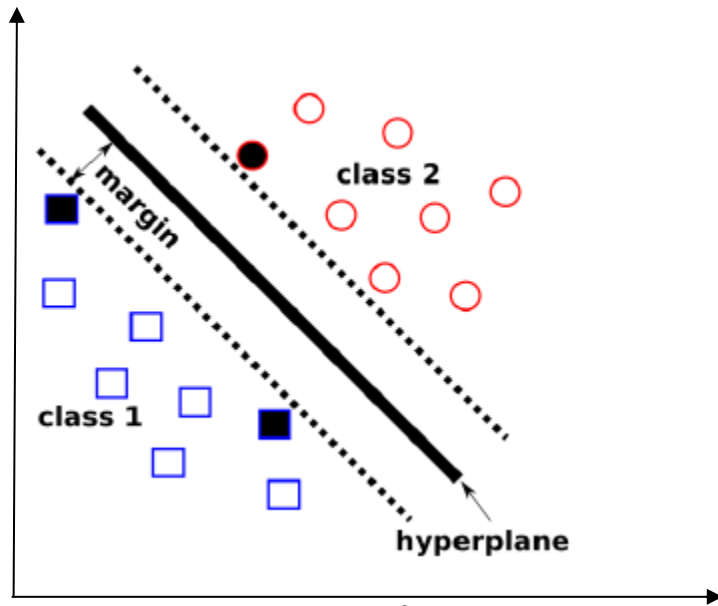


Figure 2.6: An illustration of SVM with two classes

This gives rise to two hyperplanes defining the two sides of the margin and can be re-written as

$$H_1: w_0 + w_1x_1 + w_2x_2 \geq 1 \quad \text{for } y_i = +1 \quad \dots\dots\dots 2.7$$

$$H_2: w_0 + w_1x_1 + w_2x_2 \leq -1 \quad \text{for } y_i = -1 \quad \dots\dots\dots 2.8$$

Combining H_1 and H_2 we have the equation

$$y_i(w_0 + w_1x_1 + w_2x_2) \geq 1 \quad \forall i \quad \dots\dots\dots 2.9$$

All training tuples that satisfy equation 2.9 above are called the ‘support vectors’ and are equally close to the Maximum Margin Hyperplane (MMH).

This type of SVM is regarded as ‘hard SVM’ by Shalev-Shwartz & Ben-David (2014) which can be formulated as a quadratic optimization problem. They present the Hard SVM formulation as the algorithm in Figure 2.7. This is because there is no in-sample classification error since the case is linearly separable.

input: $(x_i, y_i). \dots (x_m, y_m)$ solve: $(w_0, b_0) = \underset{(w, b)}{\operatorname{argmin}} \ w\ ^2 \text{ s.t. } \forall i, \quad y_i((w \cdot x_i) + b) \geq 1$ Output: $\hat{w} = \frac{w_0}{\ w_0\ }, \quad \hat{b} = \frac{b_0}{\ w_0\ }$

Figure 2.7: Hard SVM (Shalev-Shwartz & Ben-David, 2014))

In Figure 2.7, $\|w_0\|$ refers to the norm of w_0 . The output of the hard-SVM is the separating hyperplane with the largest possible margin.

To use the trained SVM to classify test tuples, the Lagrangian formulation is invoked as the decision boundary for classification thus:

$$d(X^T) = \sum_{i=1}^l y_i(\alpha_i X_i X^T + b_0) \dots\dots\dots 2.10$$

where y_i is the class label of support vector X_i , X^T is a test tuple, α_i and b_0 are numeric parameters that are determined by optimization or SVM algorithm and 'l' is the number of support vectors (Han *et al.*, 2012).

(b) When the data is non-linearly separable, the approach described for linearly separable data can be extended to create non-linear SVMs. This involves two steps according to Han *et al.* (2012), they are the transformation of the original input data into a higher dimensional space using a non-linear mapping and then the search for a linear separating hyperplane in the new space. This returns the problem to the same quadratic optimization problem.

In the non-linear transformation of the original input data into a higher-dimensional space, the use of 'kernel trick' is employed to cater for some problems. The kernel function $K(X_i, X_j)$ is applied to the original input data to replace the dot product, that is

$$K(X_i, X_j) = \phi(X_i) \cdot \phi(X_j) \dots\dots\dots 2.11$$

The types of kernel functions that can be employed include the following.

- Polynomial kernel of degree h: $K(X_i, X_j) = (X_i + X_j + 1)^h$
- Gaussian radial basis function kernel: $K(X_i, X_j) = e^{-((X_i - X_j))/2\sigma^2}$
- Sigmoid kernel: $K(X_i, X_j) = \tanh(kX_i + X_j - \delta)$

We can then proceed to find the maximal margin hyperplane. The soft SVM (SVM for non-linearly separable data) formulation is given in Figure 2.8 since there occur in-sample classification errors,

ε_i (Auria & Moro, 2011). It is called ‘soft’ because the hard rule of the linearly separable is relaxed by violating the constraints, the extent of which is measured by $1 - \varepsilon_i$.

input: $(x_1, y_1) \dots (x_m, y_m)$ parameter: $\lambda > 0$ solve: $\min_{w, b, \varepsilon} \left(\lambda \ w\ ^2 + \frac{1}{m} \sum_{i=1}^m \varepsilon_i \right)$ s.t. $\forall i, y_i((w \cdot x_i) + b) \geq 1 - \varepsilon_i$ and $\varepsilon_i \geq 0$ Output: w, b

Figure 2.8: Soft SVM (Shalev-Shwartz & Ben-David, 2014)

where λ is the control parameter, ε_i is the non-negative slack variable for in-sample classification error.

The strengths and weaknesses of the SVM algorithm as reported in the literature include the following.

Strengths:

1. It is very robust and has generalization ability if an appropriate kernel is chosen. This is because it does not solve for local optima.
2. Resilience against over-fitting.
3. SVM models are stable. A small change to the data does not greatly affect the hyperplane.
4. It works really well with dataset that has a clear margin of separation.
5. It is effective in high-dimensional spaces; that is where the number of dimensions is greater than the number of samples.
6. It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient. (Auria & Moro, 2011; Udomsak, 2015 and Ray, 2017)

Weaknesses:

1. It takes a long time to train on large datasets.
2. It requires extensive memory to run.
3. Choosing an appropriate function kernel may be difficult.
4. It is difficult to interpret when compared to other algorithms.
5. It does not perform well when we have a large dataset because the required training time is higher and when the dataset has more noise i.e. target classes are overlapping.
6. SVM does not directly provide probability estimates.

(Auria & Moro, 2011; Udomsak, 2015 and Ray, 2017)

SAP and dropout studies that employed the Support Vector Machines include Tekin (2014) where NN, SVM, and Extreme Learning Machine (ELM) were separately applied to students' data to predict students' GPA at graduation. Bhagvatula, Shreyas, Gupta & Singh (2015) also employed three algorithms namely, SVM, NB, and J48 algorithms separately to predict students' performance in order to investigate the contribution of students' academic efforts to their final performance. A machine learning framework was developed by Lakkaraju *et al.* (2015), using SVM and four other DM algorithms, to identify students at risk of not graduating from the high schools in the two districts used as a test bed for the framework.

A multi-class support vector machine model was developed by Asogbon *et al.* (2016) to predict students' performance using the educational dataset of students from the University of Lagos in Nigeria. The multi-class model was able to predict the performance of students adequately using 7-fold cross validation. SVM was also employed with four other algorithms by Rovira *et al.* (2017) to predict academic grades and dropout to develop a data-driven prediction system that can predict the academic grades of students in different courses and the dropout. Five selected data mining

algorithms, namely: Logistic Regression, Gaussian Naïve Bayes, Support Vector Machines, Random Forest, and Adaptive Boosting were used in the study. The performances of these algorithms were evaluated and the best of them chosen for prediction.

Eashwar, Venkatesan & Ganesh (2017) applied SVM to mine the Post Graduate (PG) students dataset collected through a questionnaire with a view to predicting students who were at the edge of their performances while using k-means for clustering. The results showed that SVM had a predictive accuracy of 96.7%. Oloruntoba & Akinode (2017) investigated the relationships between students' pre-admission academic profile and final academic performance by applying SVM on students' datasets. The study having tuned the SVM parameters found out that the RBF kernel with cost = 100 gave the best predictive accuracy of 97%. SVM was found to outperform other techniques when the results were compared with that of other machine learning techniques such as k-NN, decision trees and linear regression.

The study of Burman & Som (2019) used a multi-classifier SVM with linear and Radial Basis Function (RBF) kernels to classify students' performances. The two variants of SVM were applied to students' records containing psychological parameters collected through questionnaires. Their results showed that RBF-kernel SVM with higher predictive accuracy of 90.97% outperformed its linear counterpart.

Abu Zohair (2019) applied SVM together with four other algorithms, namely: MLP, LDA, NB, and kNN to a set of 50 graduate students' data in a Masters degree course as clusters (dataset 1 and dataset 2) to predict the students' project or dissertation grade. The dataset contains more than seven features and the study used the Leave One Out Cross Validation (LOOC) as a validation technique. The reported results showed that SVM has the best values of 76.3% and 49.3% for both

predictive accuracy and Cohen's Kappa accuracy values in dataset 2 and still holds the best values of 69,7% and 41.7% for both metrics respectively in dataset 1.

Naicker, Adeliyi, & Wing (2020) carried out a study to compare the performance of linear SVM (LSVM) with 10 state-of-the-art classical machine learning algorithms to determine the preferred algorithm for predicting students' academic performance. The study used a public dataset available on the Kaggle website containing 1000 high school students' records having 8 selected features in Mathematics, Reading, and Writing courses. LSVM was applied to the dataset using 5 cross validation and the results showed a significantly better performance when benchmark with the 10 other algorithms which include the Logistic Regression, Gaussian Naïve Bayes, and so on. Among all other metrics employed, LSVM outperformed all the algorithms in terms of accuracy with a 90.1% accuracy value. The study also discovered that among the five predictors that were considered in the study to predict students' performance, the parental level of education does not influence students' performance while others like race, gender, and lunch do.

The study of Bonifro, Gabbrielli, Lisanti, & Zingaro (2020) employed the Linear Discriminant Analysis (LDA), SVM, and Random Forest to predict student dropout using 15,000 first-year students dataset. The study modeled dropout as a logic variable having the values of 0 and 1, and the results of the study showed that when the Course Credit (CC) feature is considered all the models had a very high performance, with slightly higher results for SVM. However, this feature is not available at application time.

The final grade of students in two online networking courses was forecasted using the Support Vector Machine algorithm by Damuluri, Islam, Ahmadi, & Qureshi (2020). The students' patterns of course materials were investigated against their final grades to determine whether students are at risk of failing a course or not. The dataset used has 300 instances with data representing the

navigational behavior of students and the results revealed an accuracy of 70.21% in the prediction of students' final grades.

2.5.4.5 Neural Network (NN)

A neural network is a set of connected weighted input/output units. It was modeled after the biological neurons of human brains but formed from large numbers of simulated neurons, connected to each other like brain neurons. The strength of neuron connections may change or be changed by a learning algorithm in response to a certain stimulus or an obtained output, which enables the network to learn, in a manner just like the human brain. During the learning phase, the network learns by adjusting weights to predict the correct class labels of the input tuples (Gorunescu, 2011; Akinola, Akinkunmi & Alo, 2012).

An artificial neuron can be seen as a single device that has a certain number 'k' of real inputs x_i , weighted by respective weights w_i , summed, and then passed on to an activation function $\emptyset$ to produce a certain output, depending on a predetermined 'threshold' T (Gorunescu, 2011). Multilayer Perceptron is a form of neural network that has one or more hidden layers whose elements are called hidden neurons, that are meant to serve as intermediaries between the input layer and the output layer. Information from the environment is passed to the network through the input layer and then processed by the second layer, which is the first hidden layer, and then becomes an input to the next hidden layer. This continues till the whole hidden layers are served and the outcome is then presented to the output layer.

Studies that employed NN include Akinola *et al.* (2012) where ANN was used to predict the computer programming proficiency of undergraduate students. The study made use of a Multi-Layer Perceptron feed-forward back-propagation neural network applied to the enrollment data of 200 level Computer Science students with their results in both 100 level Mathematics and Physics

and a programming course at 200 level, which was analyzed with the aid of an open-source tool called Neuroshell. The results showed that a priori knowledge of Physics and Mathematics is an essential knowledge for a student to excel in computer programming. Simeunović & Preradović (2014) also employed NN, DT, and LR in separate to predict students' success in their studies.

Osofisan *et al.* (2014) used MLP, a variant of NN together with the J48 algorithm to study the behaviour of students' performance data. The two algorithms were applied to the M.Sc students' data from the Computer Science department of the University of Ibadan, Nigeria as a way of investigating their performances in mining education data. The comparative performance analysis of the two models ran in WEKA environment showed that MLP with 98.3% accuracy outperformed the J48 algorithm which has 85.4% accuracy although the J48 model converges faster than MLP on the training dataset. The same trend repeated itself for the test dataset where MLP has 60.2% accuracy and took 5.93secs to build but the J48 model has 52.8% accuracy and took 0.04secs to build. It was concluded that Artificial Neural Network gives the best classification results as well as prediction capability in mining education data.

In the study of Ruby & David (2015), Multi-Layer Perceptron (MLP) algorithm was used to model students' performance using data collected from the PG Computer Application course. WEKA tool was used to generate the MLP models for the two datasets of high influential factors with seven attributes and the whole dataset of twelve attributes. Models accuracy was observed and their results showed that the models of the highly influential factors perform better than that of the whole twelve attributes. Asif, Merceron *et al.* (2017) also include NN in the set of algorithms it employed, combined with a pragmatic policy to analyze the performance of undergraduate students.

Barros et al. (2019) applied MLP, DT, and Balanced bagging with and without downsample, SMOTE, and ADASYN techniques on 7,718 students' records to predict students' dropout while observing the effect of the data balancing techniques on the outcome of predictions. It was discovered that the values of the predictive accuracy, recall, and F1 were affected by the 'Accuracy Paradox' while the values of G-mean and Unweighted Average Recall (UAR) were not. The study, therefore, recommends the use of G-mean and UAR metrics for measuring reliable predictive models because they were able to accommodate minority error.

In Cuba, the MLP algorithm and J48 tree algorithm were applied by Alvarez, Callejas, & Griol (2020) to 456 students' records of Computer Engineering departments in all provinces of Cuba in order to predict at-risk students. The data employed contains both the pre-enrollment data and the students' performance data. The results of the study showed that when the binary classification was done with 'promoted' and 'not promoted' as classes, the students' at-risk of dropping out could be determined with 96.71% accuracy.

2.5.4.6 k-Nearest Neighbour (kNN)

This is a method in which a new object is classified or labeled based on its closest (k) neighbouring objects. According to Gorunescu (2011), given a training dataset and new object to be classified, the distance between the new object and the training objects is first computed (that is the one showing how farther in terms of similarity the two are to each other) and the neighboring objects closest to the new object are then chosen. The kNN algorithm, therefore, has three requirements, namely: a set of stored records (training dataset), a distance (metric) to compute the similarity between objects, and then the value of 'k'.

This technique was employed in studies like Asif, Merceron *et al.* (2017) which used kNN in the series of methods in their work to analyze the students' performance. Aulck et al. (2016) also

employed kNN together with other methods to predict students' dropout. Verma, Singh & Verma (2016) used kNN as one of the several algorithms employed in the study to predict SAP. They applied six algorithms, namely: J48 (C4.5), two Bayesian classifiers – Naïve Bayes and BayesNet, an instance-based learner (kNN), and two rule learners – OneR and JRip to the dataset of the students of SPSU University containing 10,330 records described by twenty parameters. Gender, birth year, place, current semester total university score were included in the attributes used. The results showed that students' university admission score and a number of failures in the first year exams were among the highly influencing factors on the final outcome of the performance of students. The results were validated using both 10 fold cross-validation and percentage split on testing dataset. It was observed that the J48 algorithm outperformed others by using the overall weighted average values of the two evaluations. Kabakchieva (2012) also employed IBk, an instance-based classifier and a variant of kNN among other algorithms to predict students' performance using the holdout method of percentage split.

2.5.4.7 Random Forest (RF)

This is an ensemble method that has several decision tree classifiers in it so that their collection becomes a forest as the name implies. The decision tree components of the forest are generated by random selection of attributes at each node to determine the split. Each tree depends on the value of a random vector sampled independently and with the same distribution for all trees in the forest. When RF is used for classification, each tree voted and the most popular class is returned (Han, Kamber & Pei, 2012).

Random Forest was employed in Gilbert (2017) to predict student outcomes. In the study, RF and genetic algorithm (GA) were applied to California State University freshmen and transfer students data from Fall 2000 through Fall 2010 of over 31,000 students. The study captured the interactions

between factors that might be otherwise missed in a linear system through the use of RF, and GA was also used to optimize the feature selection process. The results showed that there was a strong predictive capability for 1 and 2 year retention periods and graduation outcomes.

RF was also used with the J48 algorithm by Mishra *et al.* (2014) to predict students' performance using their social and academic integration features. The study explored the connection between the emotional skills of the students along with socio-economic features and previous academic performance parameters to predict academic performance. The values for emotional skills like assertion, leadership, and stress management were obtained using the standard Emotional Skill Assessment Process (ESAP). The results showed, among other things, that out of all emotional attributes; leadership and drive of students were found to affect students' performance.

The second approach used in data mining is the unsupervised learning approach. An unsupervised learning approach is the opposite of supervised learning and it is an approach where the class labels are unknown or there is no label at all. The clustering method and the Association rule mining are examples of the unsupervised learning approach that is being used in the study of SAP prediction. Eight DM algorithms that include RF, NN, NB, kNN, SVM, LR, and so on were applied to students dataset containing 772 instances in two related online course modules from multiple sources, namely: Students Information System (SIS), LMS, and video interactions (edify) by Hasan et al. (2020). They carried out series of experiments where they examined the effect of feature selection using Genetic Search, PCA, information gain and information gain ratio on the predicted outcome of the classifications. The experiment with feature selection, equal width transformation and information gain ratio as feature selection method turned out to produce RF as the best of the eight algorithms investigated with 88.3% accuracy.

2.5.4.8 Clustering

Clustering is the process of grouping a set of data objects into many groups or clusters based on their similarity so that objects within a cluster have a high degree of similarity but different from the objects in other clusters (Han *et al.*, 2012; Witten & Frank, 2005). In clustering, being an unsupervised learning approach, the label is unknown. Examples of clustering algorithms are the k-means algorithm and the k-medoid algorithm.

This technique was combined with other common DM methods in the study of Asif, Merceron *et al.* (2017) which used the X-means algorithm, a modification of the k-means clustering algorithm to cluster the students in the used dataset into cohorts.

2.5.4.9 Association Rule Mining

Association rule mining entails the act of finding strong rules as a way of correlation between different item-set in a large dataset. This is done by following two phases:

- Discovering the large item set, that is, the sets of items that have transaction support ‘s’ above a predetermined minimum threshold; and
- The large item set discovered is then used to generate the association rules for the database that has confidence ‘c’ above a predetermined minimum threshold (Kantardizc, 2011).

An example of the association rule mining algorithm is the Apriori algorithm. The study of Oladipupo, Daramola, Oyelade & Afolabi (2017) employed the Apriori algorithm to generate strong rules for the assessment of the influence of attendance on students’ performance prediction. The study investigated the influence of class attendance on the academic performance of students using association rule mining and it was found that the influence of class attendance on the academic performance of students was low. This indicates that class attendance is not the only factor that determines the academic performance of students.

A version of the Apriori algorithm called the AprioriC algorithm was compared with other DM techniques by Romero *et al.* (2008) when used in classifying students according to their performance. The classes used in the study are FAIL, PASS, GOOD and EXCELLENT, and several algorithms were applied in the KEEL framework for classification. The KEEL framework was used to develop a Moodle Data Mining tool experimented on the dataset. Results showed that the best algorithms were CART, GAP, Grammar-based Genetic Programming (GGP), and Neural Network Evolutionary Programming (NNEP) with more than 65% global percentage correctly classified (PCC) using the original data while the best algorithms when categorical data were used were CART and C4.5 with over 65% global PCC and the best algorithms when the balanced data were used were Corcoran, XCS, AprioriC and MaxLogicBoost with over 60% global PCC. It was also discovered that no algorithm exceeds 70% global PCC results which was attributed to the incomplete data used in the study. The moodle DM tool developed in the study does not give an overall generalization to include real-life students' performance dataset.

2.5.4.10 Nature-Inspired Algorithms (NIA)

Nature Inspired algorithms are computational intelligence algorithms that evolved through the wonderful behavior observed of some biological systems in a particular environment. This observed cooperation and foraging among these natural systems are then adapted to solve various optimization problems, that is finding optimal solutions whether in a local situation or globally (Du & Swamy, 2016).

There are many of these algorithms, however, some commonly used ones include Genetic Programming (GP), Genetic Algorithm (GA), Ant Colony Optimization (ACO), Cuckoo Optimization Algorithm (COA), Particle Swarm Optimization (PSO), and so on.

Studies that use these techniques for predicting students' academic performance and dropouts include Yıldız, Bal, Gulseçen, & Kentli (2012) which combined fuzzy logic and Genetic

Algorithm to predict students' academic performances. The Genetic Algorithm was specifically used to optimize the model. The model achieved a predictive accuracy of 84.52% making it to be an enhancement over the fuzzy logic model.

The standard Cuckoo Search (CS) and Cuckoo Optimization Algorithm (COA) were used by Chen, Hsieh & Do (2014) to train the feed-forward neural network for the prediction of SAP. In the study, CS and COA were used to optimize the weights between layers and biases of the neural network. Results showed that neural networks were well enhanced by the two optimization algorithms for better prediction of SAP; the ANN-COA however proved better than the ANN-CS in terms of RMSE, Mean Absolute Percentage Error (MAPE), and the Correlation Coefficient (R) values obtained.

Another study that employed the NIA is the work of Márquez-Vera *et al.* (2016), which predicted the students' dropouts at several stages of the course being learned using an Interpretable Classification Rule Mining (ICRM) together with a variant of GP, specifically the Grammar Based Genetic Programming (GBGP), as its core. The results of the different iterations showed that the new method was better in prediction when compared to other traditional DM techniques like the Sequential Minimal Optimization (SMO), Naïve Bayes, and so on.

The study of Hasheminejad & Sarvmili (2019) proposed a rule-based method named S3PSO (Students' Performance Prediction based on Particle Swarm Optimization) method to predict the students' academic performances using the Particle Swarm Optimization algorithm. The study used Association Rule Mining with discrete PSO algorithm to extract rules that were used to predict students' performances and the results showed that S3PSO improved by 31% of the fitness function when compared with other rule-based classification algorithms like CART, ID3, and

C4.5. It also improved by 9% of accuracy when compared with traditional DM techniques like SVM, KNN, Naïve Bayes, and so on.

One of the widely accepted and commonly used meta-heuristic algorithms is the Genetic Algorithm which is being employed in this study to produce the second category of the proposed hybrid models. A brief discussion is therefore given in respect of the algorithm in Section 2.6.

2.5.4.11 Deep Learning

Deep Learning is a technique in data mining, that approximates complex functions to the same accuracy using a ‘deep’ architecture (Zhou, Greenspan, & Shen, 2017).

The basic elements of most of the deep learning models are the auto-encoders as we have it in Deep Neural Network (DNN). However, the Deep Generative models like Deep Belief Network (DBN) and Deep Boltzmann Machine (DBM) use Restricted Boltzmann Machines (RBM) as their building blocks (Zhou *et al.*, 2017).

Another common form of deep learning model is the Convolutional Neural Network (CNN) as being explained in Aggarwal (2018).

Among the related deep learning studies in the literature though very scanty are the studies of Bendangnuksung (2018) which used a deep neural network to predict students performances and the work of Kim, Vizitei & Ganapathi (2018) which used a deep learning-based algorithm called Gritnet, that takes students’ learning activities across time as raw input and implicitly searches for parts of an event embedding sequence that are most discriminative, to predict students’ performance. Also included among the important studies in this approach is that of Li, Zhu, Zhu, Ji, & Tang (2020) where a deep learning framework called Sequential Prediction based on Deep Network (SPDN) was proposed to predict students’ performance in a course. The SPDN models students’ online behavioral sequences by utilizing the multi-source fusion CNN technique and

incorporates static information based on bi-directional LSTM. The results of several experiments conducted showed that the proposed SPDN model outperforms the baseline algorithms and has a significant improvement on early warning.

2.5.4.12 Hybrid Methods

These are the methods wherein two or more traditional DM techniques and at times optimization algorithms as may be necessary are combined to predict SAP and dropouts in order to achieve outstanding results when compared to the use of a single technique.

Some studies have evolved in this regard, even though scanty, and they include the work of Tran *et al.* (2017) which proposed a hybrid prediction model that combined a collaborative filtering strategy which uses a matrix factorization method with a linear regression strategy using SVM algorithm. The dataset employed was trained with the inclusion of skills-related features from 1,268 undergraduate students of the National University in Vietnam. Results showed that the hybrid outperformed the single implementation of the various component of the new method in terms of the Root Mean Square Error (RMSE) values as the chosen performance metric.

Altaher & Barukab (2018) also proposed a hybrid model that combined an Adaptive Neuro-Fuzzy Inference System (ANFIS) with a Genetic Algorithm (GA) for the prediction of SAP. SAP dataset containing 100 Computer Science students' performance records were passed into the ANFIS and the results of training and testing were later given to the GA component for optimization. The developed hybrid model named HGANFIS was compared with Neural Network (NN) and ANFIS models using the RMSE values as the performance metric. The results showed that the HGANFIS model was better than the other two algorithms having obtained the least RMSE values of 0.101 and 0.104 for both training and testing data respectively.

Another important study that produced a hybrid model is that of Chen *et al.* (2019) where the Decision Tree (DT) algorithm and the Extreme Learning Machine (ELM) were combined to predict the MOOC (Massive Open Online Course) dropout rate. The study used the DT algorithm to extract important features from MOOC students' learning behavior records. The extracted features by DT were later mapped with the ELM algorithm based on 2015 entry datasets and the results obtained were compared with eight other algorithms, namely: GA-ELM, DT, SVM, LR, BP (Backward Propagation Neural Network), EN (Entropy-Net), ELM and LSTM. It was found out that the DT-ELM model performed better than all others in terms of Accuracy, AuC, and F1-score which are 0.9642, 0.9412, and 0.9667 respectively as obtained for week 4 among the five weeks observations in their experiments.

Also noteworthy in the hybrid models research studies is that of Francis & Babu (2019) that combined four algorithms of SVM, Naïve Bayes, DT, and NN in a hybrid manner for selecting the best features from real-life datasets of students in various disciplines in higher education institutions. The selected features were later passed to the K-means clustering algorithm to predict students' academic performance using the majority vote approach. The feature selection phase of the model showed that the combination of academic, behavioural, and extra features gives the best accuracy of 0.7547 and the new model yielded the best result when compared with DT and NN in terms of precision, recall, F-score, and accuracy values of 0.6415 respectively.

Summarily, the taxonomy presented in this thesis so far addresses the common methods employed in SAP and dropout prediction studies, there are however some researches that employed uncommon methods like the studies of Mohsin, Norwawi, Hibadullah & Wahab (2010) where Rough set was used to determine the factors influencing the final programming performance of

students. A multi-model heterogeneous ensemble approach was also applied by Adejo & Connolly (2018) to predict students' academic performance. In their study, data from many sources like institution database, learning management system and survey were collected and an ensemble of three basic classifiers, namely: decision trees, artificial neural network and support vector machine were applied to predict students' academic performance in an efficient and accurate manner. Yang *et al.* (2018) also used multiple linear regressions and principal component analysis to predict students' performance. A neuro-fuzzy approach was also employed in Do & Chen (2013) to classify student's performance. The neuro-fuzzy classifier was trained using several algorithms including Kalman-filter; Levenberg Marquardt method and others, and with 100 iterations using 10-fold cross-validation to avoid overfitting. The model which was simulated in MATLAB R2011b had its predicted and actual class labels for testing dataset compared to determine the classifier efficiency. The approach was later compared with other common classifiers such as the SVM, Naïve Bayes, neural network, and decision tree and the result indicated that the neuro-fuzzy approach was better than all of them in performance. A statistically inclined evaluation of a host of regression and classification algorithms was done by Strecht *et al.* (2015) incorporating both common and uncommon techniques for the prediction of SAP.

2.6 Genetic Algorithm (GA)

One of the commonly employed algorithms in selecting features from a dataset before classification is Genetic Algorithm (GA) and was used in this study for feature selection in the second category of hybrid models proposed.

Genetic Algorithm is an advanced heuristic algorithm and an adaptive search technique used in finding the optimal or near optimal solution in a particular problem domain using evolution theory

in a natural selection form. It is mostly used to select features optimally in data mining (Kannan, 2016).

GA is a stochastic method that operates on a population of individuals to give better and improved approximation or near optimal solution in each generation. In a generation, a new population is created by the process of selecting individuals (a predictive model in this case) according to their fitness level and suitability in the problem domain which are later recombined together (crossover) using operators borrowed from natural genetics. The offspring may then undergo mutation for further refinement.

This process leads to the evolution of a population of individuals that are better suited to their environment than the individuals that they were created from (Khare & Burse, 2016 and Agarwal, 2015)

GA steps involved in the aforementioned are the initialization of population, fitness assignment, selection, crossover evaluation, mutation to produce a population in another generation. This is iterated in the required number of times and the fitness for all generations till the final selection of features is made. This was why scholars in the area like Babatunde, Armstrong, Leng. & Diepeveen (2014) asserted that the five important issues in GA operations are chromosome encoding, fitness evaluation, selection mechanisms, genetic operators, and criteria to stop GA.

The advantages of GA include: it's being usually better than traditional feature selection techniques, can manage datasets with many features, does not need specific knowledge of the problem domain and can be easily parallelized in computer clusters. Its shortcomings are that it might be computationally expensive since the evaluation of an individual in a generation requires building a predictive model and take a long time to converge due to its stochastic nature. (Artelnics. & Quesada, 2019).

GA is used in this study based on its wide acceptability among the researchers in the area and its great performance in feature selection optimization (Babatunde *et al.* (2014) and Kannan (2016)).

2.7 Distribution of Selected Studies by Year of Publication

The distribution of the selected research studies in this review is presented in Figure 2.9 as sourced from Table 2.1 and all the studies were further classified based on the methods employed and the goal of each study. It was observed that much of the SAP studies in this review featured in the year 2017 with 11 research works out of 61 selected for 14 years in different categories. They were grouped into studies that use one DM technique, those that use more than one, the ones particularly targeting dropout, and those that examined the influence of factors on the final prediction for convenience in the chart representing the distribution.

In addition, the distribution shows that there is a growing interest in the research area, most especially the dropout studies.

Table 2.1: Summary of SAP and Dropout prediction Studies from 2007 to 2019

YEAR	AUTHOR(S)	OBJECTIVE	DM TECHNIQUE	FINDINGS/ CONTRIBUTIONS	LIMITATIONS/ WEAKNESSES
2007	Ogor	The study's objective was to develop a methodology for students' performance prediction that can be deployed for student performance assessment & monitoring.	C5.0	Proposed DM system for monitoring and evaluating students' performances for deployment in an educational institution.	Lack the integration of students' attitude and so on into the predictive variables for mining student data.
2008	Romero, Ventura, Espejo & Hervás	It compares different data mining methods and techniques for classifying students based on their performance.	AprioriC, CART, GGP, GAP, and so on.	It was found out that some algorithms improve in classification when data is subjected to discretization and re-balancing.	The Moodle DM tool developed did not give a generalization for SAP prediction in all forms of learning.
2009	Dekker, Pechenizkiy & Vleeshouwer	To generate predictive models from selected data mining techniques that will predict Electrical Engineering students' dropouts.	BayesNet, C4.5, CART, Logistic model, RF & JRip.	It was found out that simple intuitive DT classifiers give better accuracies between 75% and 80%.	Students' progression was not considered in the study.
2010	Mohsin, Norwawi, Hibadullahi & AbdWahab	To investigate the factors influencing the final programming performance	Rough Set	The study found out that personality factors – the investigative and social type plus average cognitive in a person are important factors that influence programming performance.	The predictive accuracy used to evaluate the model can be misleading without considering other performance metrics like sensitivity and so on.

2011	Yadav, Bharadwaj & Pal	To develop decision tree-based models for predicting students' performance.	CART, ID3 & C4.5	The study concluded that CART was the best algorithm for the classification of the dataset used.	Accuracy was the only metric used as the evaluation criterion which is not enough to conclude.
2012	Akinola, Akinkumi & Alo	Predicting the computer programming proficiency of undergraduate students.	ANN	It was discovered that candidates with good background in Physics and Mathematics or Further Mathematics will perform efficiently well in computer programming.	The dataset employed did not incorporate any social and psychometric features which may be relevant to computer programming proficiency.
2012	Agarwal, Pandey & Tiwari	To generate classification models for students' performance prediction.	LR, MLP, SVM, RBF network, Voted Perceptron, SIMD, Winnow & Simple Logistic.	SVM with RBF kernel model gave the best accuracy and least RMSE among other algorithms when compared.	The study did not consider other performance metrics apart from the predictive accuracy.
2012	Kabakchieva	To predict students' performance using classification algorithms.	J48, k-NN, OneR, MLP,	It was found out that the NN model achieved the best accuracy among other algorithms used in the study.	The decision on the classifier could be misleading since it is based on accuracy alone.
2012	Yadav and Pal	To search for ways of improving the performance of Engineering students through DM	C4.5, CART, ID3.	C4.5 algorithm had the highest accuracy value when compared with others using the same dataset.	The dataset used did not have core academic features like course score so as to be able to represent students' performances while in the institution. It only focused on pre-university admission features.
2012	Pal	To develop a DM application to generate predictive models for managing Engineering students' dropout.	NB	It was found out that a short accurate prediction list of students with attritions can be produced by applying the NB model	Social behavioural features were not included in the set of features considered.

				to records of incoming new students.	
2013	Do & Chen	To present a Neuro-fuzzy approach for classifying SAP dataset	Neuro-Fuzzy	The outcome of the study showed that the proposed neuro-fuzzy approach achieved high accuracy and is comparable to other approaches.	The rigid fuzzy rule sets involved can make the approach rigid and influence the classification results.
2013	Aziz, Ismail & Ahmad	To develop a prototype SAP prediction system	NB	A framework for the prediction of SAP in higher learning institutions was proposed.	The study did not consider the students' social behaviour.
2014	Simeunovic and Preradovic	To predict student success in their studies using Data Mining	DT, LR, NN	That NN had the best accuracy of 76.8% among others for the dataset used.	The nature of the dataset used may affect the results of prediction as it may be collected through a survey where users' responses can be sometimes frustrating.
2014	Tekin, A.	To predict students' final GPA at graduation using DM techniques	ELM, NN, and SVM	The study revealed that the SVM technique yielded more accurate prediction (at a rate of 97.98%) better than ELM and NN.	Purely academic features were considered and no socio-demographic data was used.
2014	Mishra, Kumar & Gupta	To perform prediction from students' social integration, academic integration, and emotional skills data using data mining	J48 and RF	RF was the best classifier among others for early prediction of students' failure on the dataset used.	The data mining techniques used are the variants of DT only, so the results cannot be generalized.
2014	Osofisan, Adeyemi & Oluwasusi	To use DM techniques in studying the behaviour of	MLP and J48	The study found out that NN outperformed the other algorithm in terms	The two techniques considered in the study were not enough to pick the best from several DM techniques available.

			students' performance data		of accuracy, RMSE, MAE, and RAE.	
2014	Ogunde Ajibade	&	To develop a DM system that predicts students' graduation grades based on their entry results data using ID3.	ID3	The study delivers an ID3-based students' performance prediction system that predicts students' graduation grades giving their entry data.	Distinctive attributes like the social behaviour of students that can have a direct impact on the final prediction were not considered.
2015	Aziz, Ismail, Ahmad & Hassan		To develop a framework for predicting the performance of first-year Bachelor students in Computer Science	NB	The study showed the good performance of the extraction model at predicting using new data.	The social behaviour of students was excluded in the set of features used in the study which may improve the performance accuracy if considered.
2015	Joseph Devadas	&	To develop a weighted modified ID3-based model to predict SAP.	Weighted modified ID3	The study presented a weighted modified ID3 model to predict SAP.	Only the predictive accuracy was employed to evaluate the model which could make the final decision faulty.
2015	Lakkaraju, Aguiar, Shan, Miller, Bhanpuri, Ghani & Addison		To develop a machine learning framework to identify students at risk of dropout.	AB, DT, LR, RF, and SVM	A framework to identify students at risk of dropout was proposed.	The data collection tool used was a questionnaire only from which subjective responses may be obtained which may impact negatively on the result of the study.
2015	Kaur, Singh & Josan		To predict students who are slow learners.	J48, MLP, NB, SMO & REPTree	The study proposed a model to predict slow learners and their results showed that MLP was the best classifier in terms of accuracy with the dataset used.	Some other important factors like the social behaviour of students are not considered in the study. This may also a contributing factor to slow learning.
2015	Strecht, Cruz, Soares, Mendes-Moreira & Abreu		To evaluate some classification and regression	AB, AB.R2, CART, SVM, NB, RF, kNN, OLS	The study found out that in classification settings the algorithms are finding	The evaluation criteria used in the study were statistically inclined and not result-oriented

		algorithms for predicting SAP		useful patterns while in regression, the models obtained are not able to beat a simple baseline.	performance metrics used in data mining.
2015	Ahmad, Ismail & Aziz	To perform a comparative analysis of selected classification algorithms	NB, DT & Rule-Based (RB)	The Ruled-based model was the best among other techniques for the dataset used.	Only predictive accuracy was employed as a performance metric which may be misleading.
2015	Ruby & David	To analyze the influencing factors on students performance using SAP predictions	MLP	The study analyzed the influencing factors on students performances using the MLP algorithm	Social behaviour of students was not considered which would have improved the results..
2015	Bhagvatula, Shreyas, Gupta & Singh	To investigate the contribution of students' academic efforts on the prediction of students' performance.	SVM, NB, J48	The study conducted a comparative analysis of SVM, NB, and J48 and discovered that attributes that were highly coupled and more likely to change contribute more towards SAP.	The other metrics used apart from the predictive accuracy were not properly stressed.
2016	Amrieh, Hamtini & Aljarah	To examine the effect of the introduction of learners' behaviours on SAP predictions.	NB, DT, ANN & Ensemble methods	The study proposed a predictive model with students' behavioural features incorporated and adding bagging, boosting, and RF to obtain useful ensemble classifiers.	The study strictly for SAP solution and no link with dropout at all.
2016	Saa	To generate predictive classifiers from the use of multiple features of SAP	C4.5, ID3, CART & CHAID	The study found CART as the best among other algorithms applied to the dataset.	The dataset used was too small in volume and may affect the results of the prediction.
2016	Kaur & Singh	To find the best DM technique from the	NB, J48	The study found out that NB had higher accuracy	The two techniques considered in the study were not enough to

		two selected data mining techniques.		of 63.59% than the J48 algorithm with 61.53%.	generalize on, considering several available DM techniques.
2016	Verma, Singh & Verma	To conduct a performance evaluation of some DM algorithms	BayesNet, J48, NB, JRip, OneR, kNN	The study presented a comparative analysis of some algorithms for predicting SAP.	Non-inclusion of social behavioural features in the data used may make the results ungeneralisable.
2016	Hamsa, Indiradevi & Kizhakethottam	To develop a SAP prediction model using a decision tree and fuzzy-genetic algorithms.	DT, Fuzzy-genetic	The study proposed a SAP prediction model based on DT and Fuzzy genetic algorithm to identify students' performances in each subject.	The approach may be biased because of the rigidity of fuzzy rules.
2016	Aulck, Velagapudi, Blumenstock & West	To predict students' dropout rate.	Regularised LR, RF and kNN	The study was able to highlight several early indicators of students' attrition for accurate prediction of dropout rate.	Strictly on the dropout and no link with the general SAP problem.
2016	Al-Barrak & Al-Razgan	To predict students GPA based on grades in previous courses	J48	The study generated classification rules to predict students' GPA and also evaluate the important courses that most influence the final students' GPA.	The study focused more on grades in courses and not on other influencing factors on GPA
2016	Asogbon, Samuel, Omisore & Ojokoh	To develop a multiclass SVM classifier to predict the performance of students for admission into faculty programmes.	SVM	The study proposed a multiclass SVM that adequately predict students' performances at k = 7 cross validation.	Consideration is being given to SVM only, generalizability is not assured.
2016	Mueen, Zafar & Manzoor	To apply DM methods to predict SAP based on students' academic	NB, MLP & C4.5	The study found out that NB outperformed MLP and C4.5 on the datasets used.	The dataset used is relatively small and may not represent well the real-life encounter of

		records and forum participation.			students in undergraduate courses.
2017	Asif, Merceron, Ali & Haider	To analyze the performance of undergraduate students using EDM techniques.	DT, NN, NB, RI-IG, k-NN, X-means and pragmatic policy	The study introduced a pragmatic policy for finding typical students' progression.	Non-inclusion of social behavioral features may have an impact on students' performance.
2017	Asif, Hina & Haque	To study DM techniques for predicting students' graduation performance in their final year.	DT-ACC, DT-IG, DT-GI, NB, NN, RF-ACC, RF-IG, RF-GI.	The study proposes a model for predicting graduation performance in a four-year programme using pre-university marks and marks of first and second-year courses only.	Non-inclusion of socio-economic as well as demographic features can affect the model results.
2017	Gilbert	To predict student outcomes using RF and genetic algorithm.	Random Forest (RF)	The study proposed a model to predict students' success and identifying factors that show strong predictive capability with regards to successful graduation using GA and RF.	The dataset used did not include social behavioural features which can also impart on students success.
2017	Rovira, Puertas & Igual	To develop a data-driven system that predicts academic grades and dropout	LR, Gaussian NB, SVM, RF, and AB.	The study delivers a data-driven system that predicts grades and dropout that could be used for decision-making on dropout students.	The recommender system developed is specific to some selected courses in the institution of the case study.
2017	Oladipupo, Daramola, Oyelade & Afolabi	To assess the influence of attendance on SAP using the DM technique.	Association Rule mining	The study found out that the impact of class attendance on SAP is very low.	The study did not consider any social behavioural feature which could as well impart on SAP and hence may lack generalization.
2017	Oloruntoba Akinode &	To investigate the relationship between	SVM	It was found out that SVM with Radial Basis	Misleading classifier's performance claim is inevitable

		students' pre-admission academic profiles and final performance.		Function kernel gave the best performance at $c = 100$ outperforming other algorithms when compared.	owing to the use of only predictive accuracy as a performance metric.
2017	Eashwar, Venkatesam & Ganesch	To use k-means to group the number of students and SVM for classifying students' performance.	SVM, k-means	SVM tool accuracy was 96.7% in this study and it was also revealed that the junior students have high probability level than the seniors in the high class.	Emphasis was placed on predictive accuracy for judging classifier's performance which may be misleading.
2017	Makhtar, Nawaniug, Nor & Shamsuddin	To analyze the use of Naïve Bayes in predicting Students' Academic Performance.	NB	The study found out that the combination of subjects wherein there is Chemistry always gives better accuracy with reference to the case study.	A control experiment needs to be conducted to combine Chemistry with any of the Art or commercial subjects and observe the classifier's performance before hasty generalization of findings.
2017	Amra & Maghari	To use KNN and Naïve Bayes algorithms to predict secondary school students' performance.	NB, KNN	It was found out that NB was a better classifier than kNN on the SAP dataset of the secondary school used.	The dataset used did not include any of the core academic features like scores in different subjects and this may make the results ungeneralisable.
2017	Agrawal, Vishwakarma & Sharma	To analyze the use of four DM algorithms in predicting students' performance.	DT, RF, NB & Rule Induction	The study found out that DT is a better classifier than RF, NB, and RI on the dataset used.	Only the predictive accuracy was used as a performance metric and judgement on classifiers may be misleading.
2017	Sultana, Khan & Abbas	To predict potential dropouts among electrical engineering and computer students of two universities	NB, DT, LR & NN	The result of the study revealed that the Naïve Bayes algorithm outperformed all others by achieving a predictive accuracy of	The manner of data collection from students in the survey may impact negatively on the results of prediction, using confirmed

		in Pakistan using both cognitive and non-cognitive features.		65% in the best combination of the features.	data values from school databases is more reliable.
2018	Yang, Lu, Huang, Huang, Ogata & Lin	To enhance Multi LR predictive capability with PCA	Multiple Linear Regression, PCA	The study combined multiple LR with Principal Component Analysis (PCA) to predict SAP for improved accuracy.	The model was not flexible due to the fact that it is strictly associated with the learning activity and particular data in the study.
2018	Adejo & Connolly	To predict SAP using a multi-model heterogeneous ensemble approach	ANN, SVM, DT, Ensemble	The study found out that the approach of using multiple data sources along with the heterogeneous ensemble technique is very efficient and accurate in the prediction of students' performances and proper identification of students at risk of attrition.	There is a need for the inclusion of more variables in the approach proposed for better analysis. This will increase the models' generalizability level.
2018	Razaque et al.	To apply the NB algorithm to predict students' performance.	NB	The study presented an approach using NB classifier to predict the GPA of graduate students after using K-means to group the students into clusters.	Only the predictive accuracy was used as a performance metric and may fault judgement on classifiers.
2019	Burman & Som	To examine the performances of Support Vector Machines with two different kernels.	SVM	The study found out that SVM with Radial Basis Function kernel performed better than that with the linear kernel at $c = 2$.	The dataset used was collected through a questionnaire whose values may not be objective enough like the direct data from real students' performance.
2019	Barros et al.	To find the best evaluation metrics	MLP, DT & Balanced Bagging	The study found out that values of the predictive	The effect of the applied DM algorithms on data

		that could minimize the effect of Accuracy Paradox (AP) in an imbalanced dropout dataset.			accuracy, Recall, and F1 metrics were affected by AP while that of G-mean and UAR were not.	smoothening is not considered apart from the fact that instances that could produce better decisions may be lost in the process of downsampling.
2019	Abu Zohair	To predict the project or dissertation grade of students.	SVM, MLP, LDA, NB & kNN		The study reported SVM as having the best values of both the predictive accuracy and Cohen's Kappa accuracy.	The dataset used in the study is small and may influence the study results.
2020	Baker et al.	To predict K-12 dropouts in a school district.	LR		The study found out that dress code violations have a strong relationship with dropout among other factors used in the study.	The particular characteristic of the district population data employed may influence the study outcome and therefore generalization based on the study outcome may be difficult.
2020	Alvarez, Callejas & Griol.	To predict at-risk students among the students of the Computer Engineering department in all provinces of Cuba.	J48, MLP		The study found out that the students were at risk of being dropped out mostly after the first year with 96.71% accuracy when binary classification is considered as in the study.	The study places emphasis on the predictive accuracy obtained for the 3 levels of data used in the study when reporting results. This may make the results liable to accuracy paradox because of imbalance classes in dropout data.
2020	Mubarak, Cao & Zhang	To predict at-risk students' weekly and final status of the course.	Sequential SeqLR) IOHMM	LR &	The study found out that students' activities in the early weeks of the course are more interactive than in the middle or final weeks for all activities. And the proposed models outperformed the four baseline methods in terms of precision and accuracy.	The SeqLR and IOHMM cannot achieve accurate predictive accuracy in the first early weeks as the resulting prediction accuracy depends on the interaction of students and their assignments in previous weeks.

2020	Aye	To perform a comparative analysis of some selected DM algorithms.	J48, NB, Bayes Net, and Decision stump	J48 algorithm was the best with a predictive accuracy of 75.83% and an MAE value of 0.2155.	The number of DM algorithms investigated is too small when compared with the number of existing DM techniques and as such the study outcome can be challenged.
2020	Ayyas, Asif & Zahoor	To predict the performance of primary school students in various districts of Punjab, Pakistan.	NB, RF, SVM, and Deep Learning	Deep Learning has the highest predictive accuracy followed by NB with an accuracy of 75%.	The structure of the primary school data may be too simple for the DM techniques employed. Therefore the results of prediction in the study cannot be used to affirm any pattern on complex data like that obtainable in tertiary institutions.
2020	Naicker, Adeliyi & Wing	To compare the prediction performance of Linear SVM with 10 benchmark algorithms.	LSVM	LSVM outperformed all the 10 benchmark algorithms in terms of accuracy with a 90.1% value. Parental level of education as a factor does not influence students' performance while race, gender, and lunch do.	The public dataset used in the study does not have the commonly employed data in it and as such comparing the result of the study with what is obtainable in existing literature may be difficult.
2020	Hasan et al.	To investigate the effect of feature selection using Genetic Search, PCA, information gain, and information gain ratio.	RF, LR, kNN, SVM, CN2, NN, NB and Tree	Experiment with feature selection, equal width transformation, and information gain ratio produced RF as best among the eight algorithms investigated.	The number of features in the dataset employed and the use of only two course modules may not give a good representation in terms of classification outcome.
2020	Vamshidharreddy et al.	To predict students' academic performance using	NB, DT & SVM	The study reported that NB outperformed the other two techniques by having an accuracy of	Only the predictive accuracy was used for determining the best classifier. The best classifier in this study was DT.

		selected DM techniques.		77% with SVM (38%) and DT (71%).	
2020	Bonifro et al.	To predict student dropout using first-year students' dataset.	LDA, SVM & RF	The results of the study showed that when the Course Credit (CC) feature is considered, all the models reach very high performance, with slightly higher results for SVM.	The results obtained in each of the schools considered were significantly different because the peculiar circumstances of students in each of the schools were not considered.
2020	Damuluri et al.	To forecast the final grade of students in an online networking course.	SVM	The results of the study revealed an accuracy of 70.21% in the prediction of students' final grades based on navigational behavior of the students.	Data containing personal observations of students' behaviors are impossible just for any online programme as in this study. This makes it impossible for tutors to have true cause-and-effect values regarding students' performance.

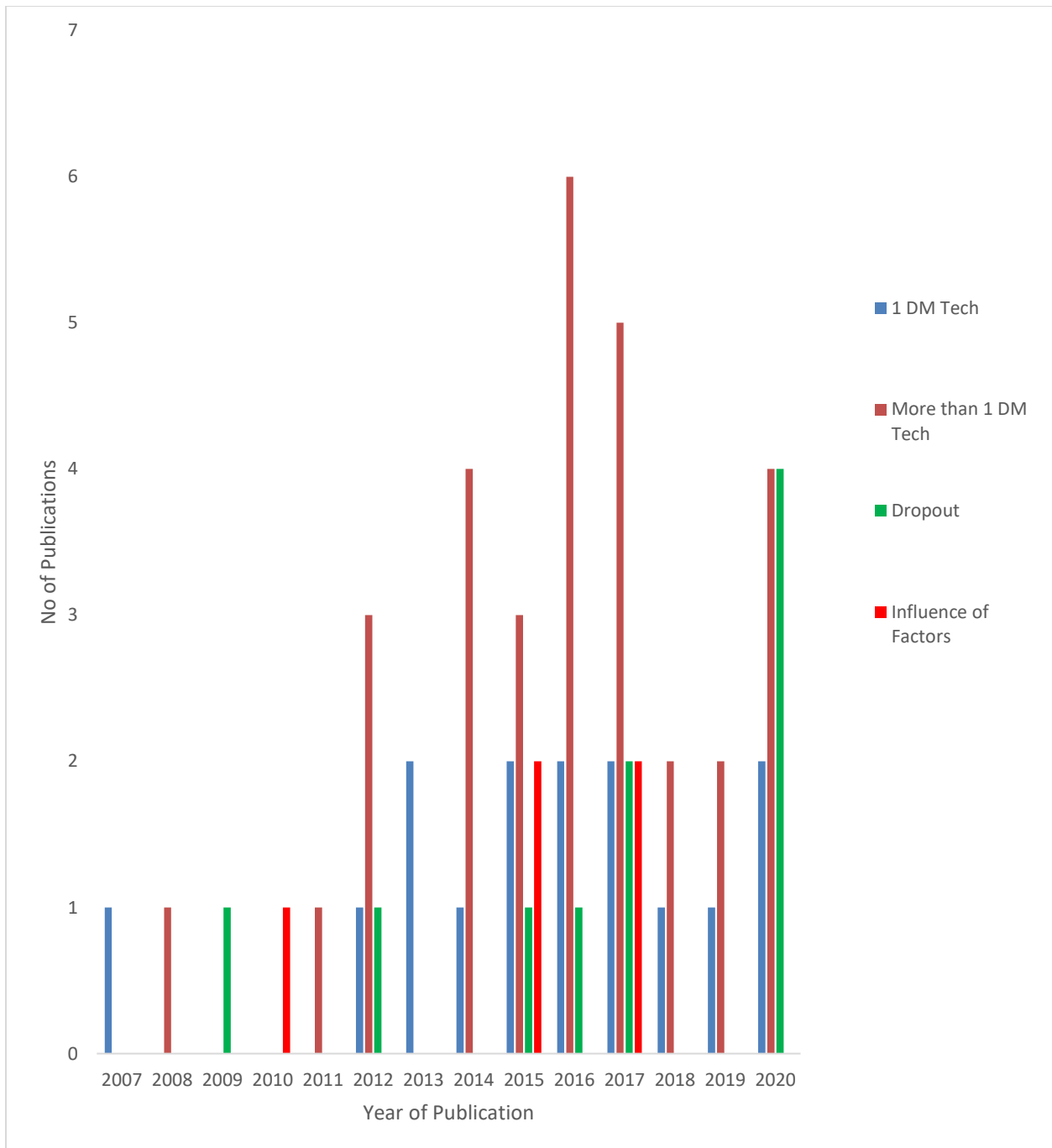


Figure 2.9: Distribution of the selected studies by year of publication

The limitations of students' academic performance prediction studies include the vastness of the attributes selected for the study, the number of performance metrics used in the evaluation of the models developed, and the non-generalizability of models. All these are shown from the summary of SAP and dropout prediction studies in Table 2.1 above.

Most of the predictions were on the general success and failure of students, no literature has dwelled on the prediction of excellent students, which if its influencing factors are known, it can be used to solve the problem of students' failure or academic dropout.

The major focus of the prediction of SAP and dropout studies is the nature of the attributes employed in the mining and prediction as well as the performance of the DM techniques used. The review also showed that hybrid models for SAP and dropout predictions are also scanty and the ones available did not use the widely accepted learning algorithms. Another inference from the review is that the SAP and dropout datasets are scanty made public because each institution of higher learning considered students' performance data as too confidential to be made public.

This study, therefore, proposes to address the problem of over-reliance on predictive accuracy metric and the quest for better classifier efficiency by developing hybrid prediction models for students' academic performance and dropout using Support Vector Machines, Naïve Bayes, and Genetic Search algorithms that will also add to the scanty hybrid studies.

2.8 Related Works

It is important that this study examine other studies that have combined the two algorithms being considered in this study to see their relationship with this study aside from the various SAP and dropout prediction studies mentioned in Table 2.1.

Among these studies is the study of Piątkowska & Martyna (2009) where a hybrid classifier based on the SVM method for cancer classification was presented. The study hybridized SVM with fuzzy Naïve Bayes (FNB) by using the FNB to estimate the probabilities for classes while SVM was used to classify samples using the original training dataset of gene expression profiles. The hybrid classifier was evaluated with published cancer data and found to have an accuracy of 80% compared with the accuracies of the SVM method and the FNB that were 72% and 68% respectively.

Sagale & Kale (2014) combined the Naïve Bayes algorithm with SVM for intrusion detection system by making the training data of network intrusion detection to pass through the whole stages of Naïve Bayes classifier and then provide the misclassified data to SVM for targeted classification. The hybrid model presented was evaluated using detection rate and false positive rate values in comparison with the same metric values for a single implementation of SVM and NB and it was found that the hybrid model provided a higher detection speed and accuracy.

Abikoye *et al.* (2017) combined Naïve Bayes and SVM classifiers in an ensemble manner using stacking to build a binary text classification model. The performance evaluation of the ensemble classifier was carried out using two datasets – the sentiment labeled dataset and the SMS spam dataset in WEKA environment and the results showed that the ensemble classifier had more correctly classified instances than either the Multinomial Naïve Bayes (MNB) and the SVM in stand-alone, less number of incorrectly classified instances and higher Kappa statistic values although the time taken to build the model in the ensemble was higher.

Shakya & Sigdel (2017) presented an approach to develop a hybrid algorithm based on SVM and NB for detecting anomalies in information. They used a weighted hybrid model where classifiers with high accuracy are given higher weights and the one with lower precision is given lower

weights. The prediction of the hybrid model was computed using the weighted sum of predictions of member algorithms (that is SVM and NB) with the mean square error of forecast used as the error metric. The hybrid model, when evaluated using some common performance metrics like ROC AuC, accuracy, recall, and so on, was found to outperform the SVM and NB in their single implementations.

2.9 Research Gaps

The research gaps identified and being taken care of by this study are the over-reliance on predictive accuracy to the detriment of other metrics which can lead to the wrong conclusion on classifiers or models developed, the absence of studies that combine the two tasks of SAP and dropout predictions in EDM and non-existent of the hybridization of the widely used classification algorithms like SVM and NB in the area.

Other identified gaps which are yet to be addressed include the integration of social behavior of students in the dataset for modeling SAP and dropout models, scanty application of deep learning techniques to the prediction of SAP and dropout and the use of optimization techniques in the prediction of SAP and dropout rates is not common. The combination of various other methods or machine learning algorithms for the classification of education data to form hybrid models is also not yet explored to its full capacity.

CHAPTER THREE

METHODOLOGY

This chapter describes the various steps taken to achieve the aim of this study as demanded by the specified objectives. It explains the whole procedure in a logical sequence towards the realization of the various objectives of the study. These procedures range from the preliminary stages of data collection and preparation, and auto-classifier detection to the feature selection layer of the hybrid prediction models design. The classification layer followed the feature selection layer in the design and the chapter is concluded with the models' evaluation layer to assess the developed models' performances.

3.1 Preliminary Stages of the Hybrid Prediction Models Design

Two stages were required before the hybrid models were produced, namely: the data collection and preparation, and the auto-classifier detection stages as shown in the two models' design structures of Figure 3.3.

The essence of the two preliminary stages was to refine the datasets and make them ready for use and also to obtain the best classifier for the datasets between SVM and Naïve Bayes algorithms to be used for classification in the design of the hybrid models.

3.1.1 Data Collection and Preparation

Students' Academic Performance data incorporating the dropout data for three graduating years - 2017, 2018 & 2019 for polytechnic datasets and 2015, 2017 & 2018 for University datasets, were collected from the Federal Polytechnic, Bida, and the University of Ilorin. These institutions were chosen because they are both old tertiary institutions and have the expected large population of students that the study required.

These datasets collected were in their raw forms comprising of the detailed performances of students semester by semester in the case of the polytechnic and session by session as for that of the university. They were prepared and analyzed in an MS Excel package to make them minable by the different data mining algorithms to be used. The two datasets were transformed into four – that is two SAP datasets and two dropout datasets. The four datasets were labelled as UniDataSAP_num, PolyDataSAP_num, PolyDataDropout_num and UniDataDropout_num. The Dropout datasets were obtained from the SAP datasets as the set of data in the first semester or the first session of the students’ programme in both the polytechnic and university datasets respectively. The description of the various attributes used in the datasets after some simple transformations is as shown in Table 3.1.

Table 3.1. Datasets attributes and their description

S/N	Attributes	Description	Domain Values
1	Sex	The student’s gender	M for Male and F – Female
2	Age	The age of the student as at the time of graduation.	23, 32
3	Year of Graduation	The year the student graduated from the Polytechnic or the University.	2015, 2018
4	Scores	Scores in different courses offered in each semester. There will always be more than one course in a semester.	60, 65, 80
5	CLASS (Response Variable)	This is the adopted class label and it is the predicted value.	EXCELLENT, VERY GOOD, GOOD, PASS, LIKELY TO DROP

The class values adopted in this study were of two types, the SAP datasets are multiclass while that of the Dropout datasets is binary. The SAP datasets have EXCELLENT, VERY GOOD, GOOD, PASS as their classes. For students of polytechnic whose final remark is ‘Distinction’ it is taken as EXCELLENT class, ‘Upper Credit’ taken as VERY GOOD, ‘Lower Credit’ as GOOD

and 'Pass' as PASS. For the students of the university whose final remark is 'First Class', it is taken as EXCELLENT, 'Second Class Upper' taken as VERY GOOD, 'Second Class Lower' as GOOD, and 'Third Class' as PASS in the SAP datasets. In the Dropout datasets, there are two classes, Pass or Likely to Drop. For the polytechnic dropout dataset, a student is classed as PASS if the CGPA is greater than 2.0 otherwise it is classed as LIKELY TO DROP. For the University dropout dataset, any student with less than 2.5 as GPA is classed LIKELY TO DROP while all others are regarded as PASS.

The datasets were refined by removing attributes that were not too relevant simply because they are not common to all the students. Any instances with missing data from the four datasets were also removed during the preprocessing stage in the programming environment.

After the preprocessing stage, the pre-processed data for both SAP and dropout were passed to each of the two classifiers – SVM and NB to determine the best classifier among the classifiers in the auto-classifier detection stage.

3.1.2 Auto-Classifier Detection Stage

This is the second of the two preliminary stages in the design of the hybrid prediction models as depicted in Figure 3.3. It is the stage where the investigative study of the performances of Naïve Bayes and SVM on each dataset was carried out.

In this stage, the preprocessed data (SAP or dropout dataset) were passed to the Naïve Bayes algorithm represented by Figure 3.1 and SVM algorithm as represented by Figure 3.2 in separate and the classification performances of the two classifiers observed. This is depicted in detail in Figure 3.3.

Any of the algorithms that give the best performances on each of the datasets were noted for adaptation in the classification layer of the hybrid prediction models design. This is done irrespective of the feature selection strategy involved.

Algorithm 1: Naïve Bayes algorithm

Input: $D = \{x_1, x_2, x_3, \dots, x_n\}$ Training dataset containing attributes $x_1 \dots x_n$ with associated class labels

Output: T – Classification Output.

Method:

1. **for** $i = 1$ to j **do**
2. **for** each class $C_i \in D$, **do**
2. Find the prior probability, $P(C_i)$
3. **end for**
4. **for** each attribute value A_i **do**
5. Find the class conditional probability, $P(A_i/C_i)$
6. **end for**
7. **for** each training instances, $x_i \in D$ **do**
8. Find the posterior probability, $P(C_i/x_i)$
9. **while** $P(C_i/x_i)$ is maximum **output** T
10. **end for**
11. **end for**

Figure 3.1: Algorithm for Naïve Bayes (NB) adapted from Farid, Zhang, Rahman, Hossain, & Strachan (2014)

3.2 Design of the Proposed Hybrid Prediction Models

The proposed hybrid prediction models in this study are of two types, namely: the Recursive Feature Elimination (RFE) based hybrid models and the Genetic Algorithm (GA) based hybrid models. Each of the two types of models is also in two forms based on the kind of dataset (SAP or dropout) employed. This resulted in two hybrid prediction models for each type as follows: the RFE-based SAP hybrid prediction model and the RFE-based Dropout hybrid prediction model, both shown in Figure 3.4. The GA-based SAP hybrid prediction model and the GA-based Dropout hybrid prediction model are both depicted in Figure 3.6.

Algorithm 2: Support Vector Machine algorithm

Input: $D_T = \{x_1y_1, x_2y_2, x_3y_3, \dots, x_ny_n\}$ Training dataset containing attributes $x_1 \dots x_n$ with associated class labels y_i ,

Output: w, b

Method:

1. **for** $i = 1$ to j **do**
 2. **if** linearly separated **do**
 solve
$$(w_0, b_0) = \underset{(w,b)}{\operatorname{argmin}} \|w\|^2 \text{ s. t. } \forall i, \quad y_i((w \cdot x_i) + b) \geq 1$$
 3. write $\hat{w} = \frac{w_0}{\|w_0\|}, \hat{b} = \frac{b_0}{\|w_0\|}$
 4. **else**
 solve $\min_{w,b,\varepsilon} \left(\lambda \|w\|^2 + \frac{1}{m} \sum_{i=1}^m \varepsilon_i \right) \quad \text{s.t. } \forall i, y_i((w \cdot x_i) + b) \geq 1 - \varepsilon_i \text{ and } \varepsilon_i \geq 0$
 5. **end**
 6. **end for**
-

Figure 3.2: Algorithm for SVM adapted from Shalev-Shwartz & Ben-David (2014)

The models were designed leveraging on the outcome of the auto-classifier detection stage as shown in the two conceptual frameworks in Figure 3.4 and Figure 3.6.

There are three layers in each model design structure and they are the feature selection layer, the classification layer, and the model evaluation layer.

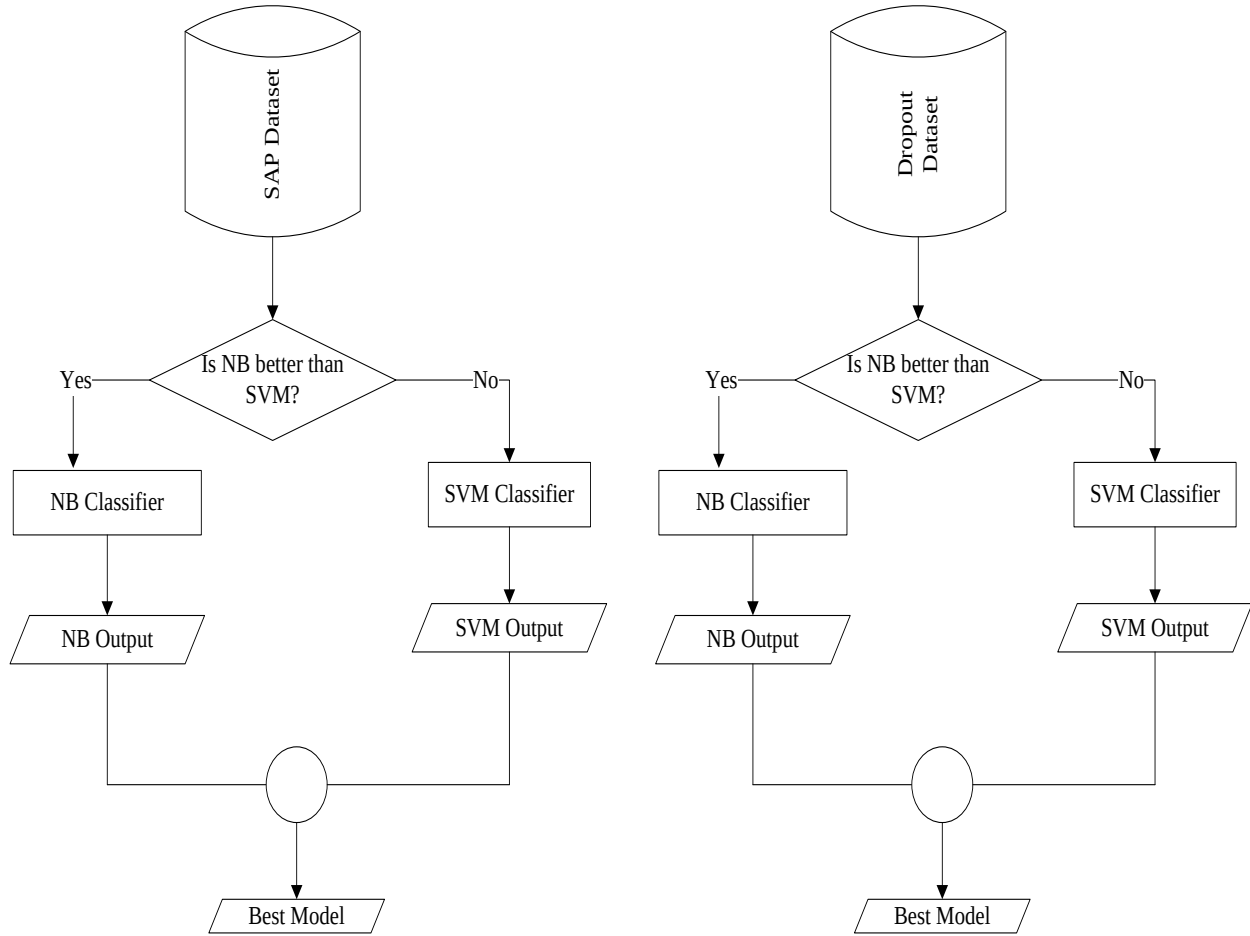


Figure 3.3: Procedures involved in the Auto-Classifer Detection stage

3.2.1 Feature Selection Layer

Two categories of feature selection strategies were applied to each of the problems (SAP and dropout) in this study. These strategies are the Recursive Feature Elimination (RFE) strategy as shown in Figure 3.5 and the GA feature selection strategy as shown in Figure 3.7.

3.2.1.1 The Recursive Feature Elimination (RFE) Strategy

The wrapper-based RFE strategy was employed using the other algorithm apart from the best classifier already gotten from the auto-classifier detection stage. Naïve Bayes algorithm is used with RFE to select the best features from any of the datasets (that is the RFE_NB) if the SVM

comes out to be the best classifier in the auto-classifier stage while SVM is used with the RFE strategy to select the best features from any of the datasets (that is RFE_SVM) if Naïve Bayes classifier comes out to be the best classifier in the auto-classifier detection stage.

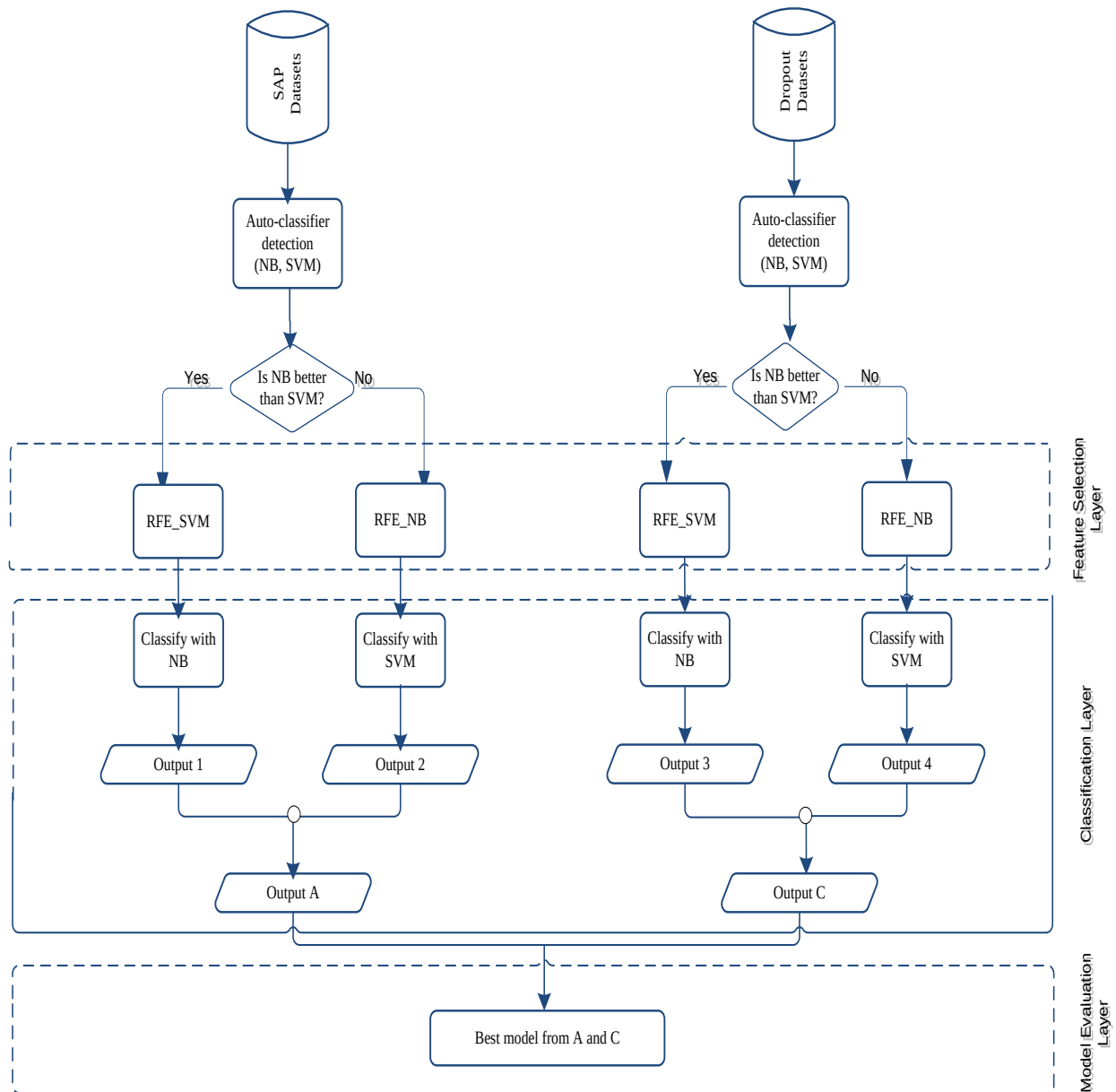


Figure 3.4: Conceptual Framework of the proposed RFE-based Hybrid Prediction Models

The SVM used here is the one with Radial Basis Function (RBF) kernel, arrived at from the auto-classifier detection stage as the best of the four variants of SVM implemented.

The RFE algorithm shown in Figure 3.5 was implemented in the R programming environment to give the best feature subsets from each of the dataset types.

Algorithm 3: Recursive Feature Elimination

Input: $A = \{x_1, x_2, x_3, \dots x_n\}$ Set of all predictors

Output: S – set of preferred predictors for classification

Method:

1. $S = \{ \}$ Initializing storage for best feature subsets
 2. Tune/train the model on the training set using all predictors
 3. Calculate model performance
 4. Calculate variable importance or rankings
 5. **for** each subset size S_j ; $j = 1 \dots s$ **do**
 6. keep the S_j most important variables
 7. pre-process the data {optional}
 8. train the model on the training set using S_j predictors
 9. calculate model performance
 10. re-calculate the rankings for each predictor [optional]
 11. **end for**
 12. Calculate the performance profile over the S_j
 13. Determine the appropriate number of predictors, S
 14. Use the model corresponding to the optimal S_j , $S = S_j$
-

Figure 3.5: Recursive Feature Elimination algorithm adapted from Goscik & Łukaszuk (2013)

The detailed working principle of the RFE algorithm had already been explained in Section 2.3.2.1.

3.2.1.2 The Genetic Search Algorithm Strategy

The Genetic Algorithm (GA) strategy was employed in searching for the optimal feature subsets from both types of datasets as in Figure 3.6 as it was done for the RFE strategy. In the GA search, the fitness function used was inbuilt in the programming environment library and it's being used generally for feature subset selection with the necessary GA operators like the selection, crossover,

and mutation operators applied in various iterations (or generations) during the feature optimization process.

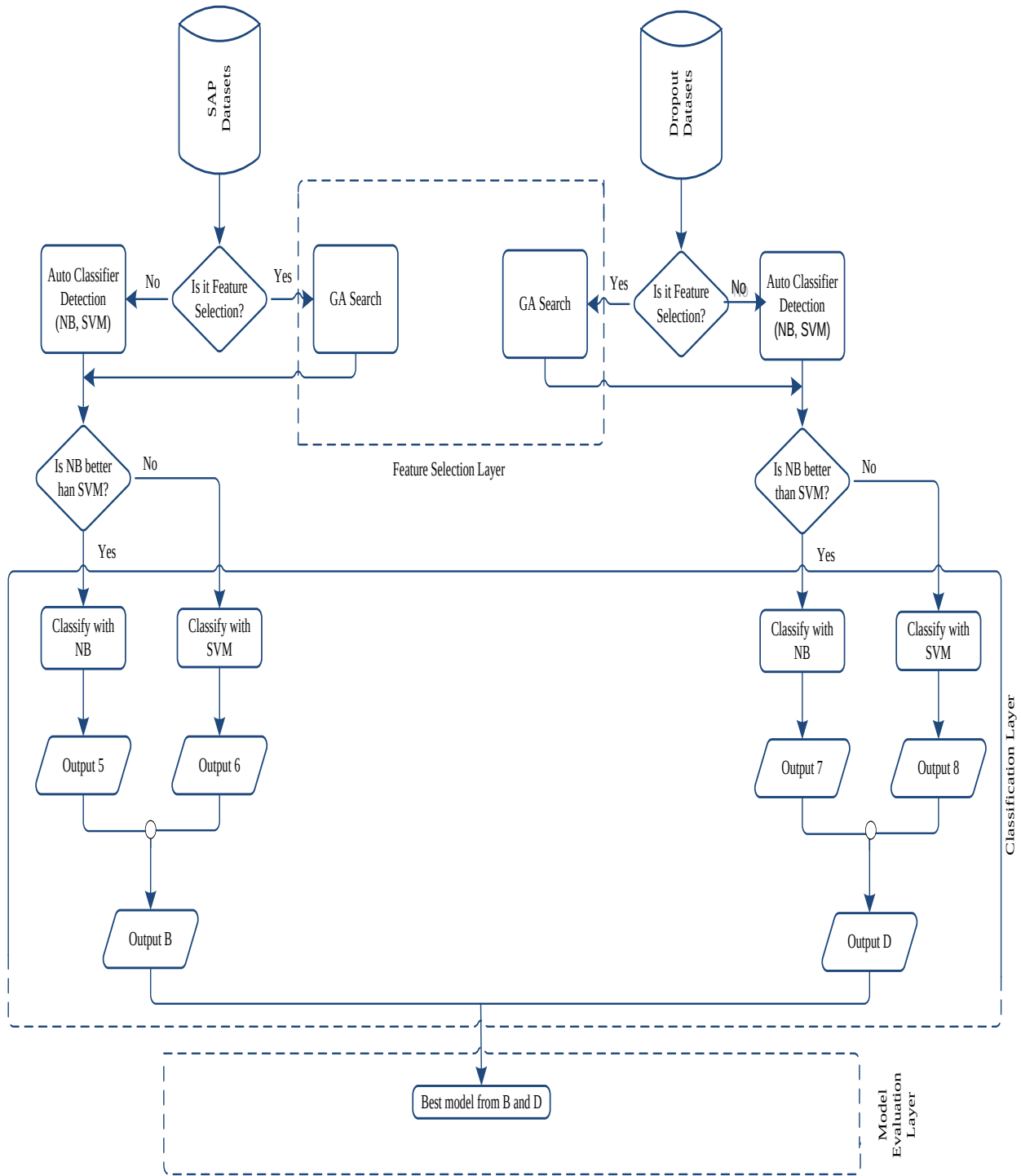


Figure 3.6: Conceptual Framework of the proposed GA-based Hybrid Prediction Models

The GA set parameters used are as stated in Table 3.2 while the GA search procedures are represented by the algorithm in Figure 3.7. Three population sizes of 20, 50, and 100 experimented for the GA strategy to observe the one that gives the best model performance in the hybrid models evaluation.

Table 3.2 Parameters used for the GA feature selection

Parameter	Value
Population Size	100
Number of Generation	20
Probability of Crossover	0.6
Probability of Mutation	0.033
Report Frequency	20

Algorithm 4: Genetic Algorithm Procedure

Input: $A = \{x_1, x_2, x_3, \dots, x_n\}$ Set of all predictors

Output: S – set of preferred predictors for classification

Method:

1. $S = \{ \}$ Initializing storage for best feature subsets
 2. **for** $i = 1$ to n *{for all predictors $K_j \in A$ }* **do**
 3. initialize the population
 4. Assign fitness function, $f(x)$
 5. while $t \neq \text{TerminationCriteria}$
 6. **do** crossover and mutation
 7. evaluate the fitness of the population
 8. Go to 4
 9. **end for**
-

Figure 3.7: Algorithm for GA feature selection process

3.2.2 Classification Layer

The next layer of the hybrid prediction model design after the feature selection is the classification layer. In this layer, the best classifier from the auto-classifier detection stage was employed at the front-end part of the hybrid model for the final classification of each of the improved datasets (SAP or dropout) having undergone feature subset selection in Section 3.2.1. That is to say that the Naïve Bayes algorithm was used for classification if it comes out to be the best classifier in the auto-classifier stage while SVM was used if it comes out to be the best classifier in the auto-classifier detection stage (Figure 3.3).

At this junction, there is a need to link the results of the auto-classifier detection stage on the best classifier to the design of a comprehensive algorithm for each of the hybrid prediction models in this study.

Therefore, with the adoption of SVM as the best classifier for the dropout datasets and NB as the best classifier for the SAP datasets, the sketch of the models' architecture of the four hybrid prediction models proposed in this study are in Figure 3.8, Figure 3.10, Figure 3.11 and Figure 3.14.

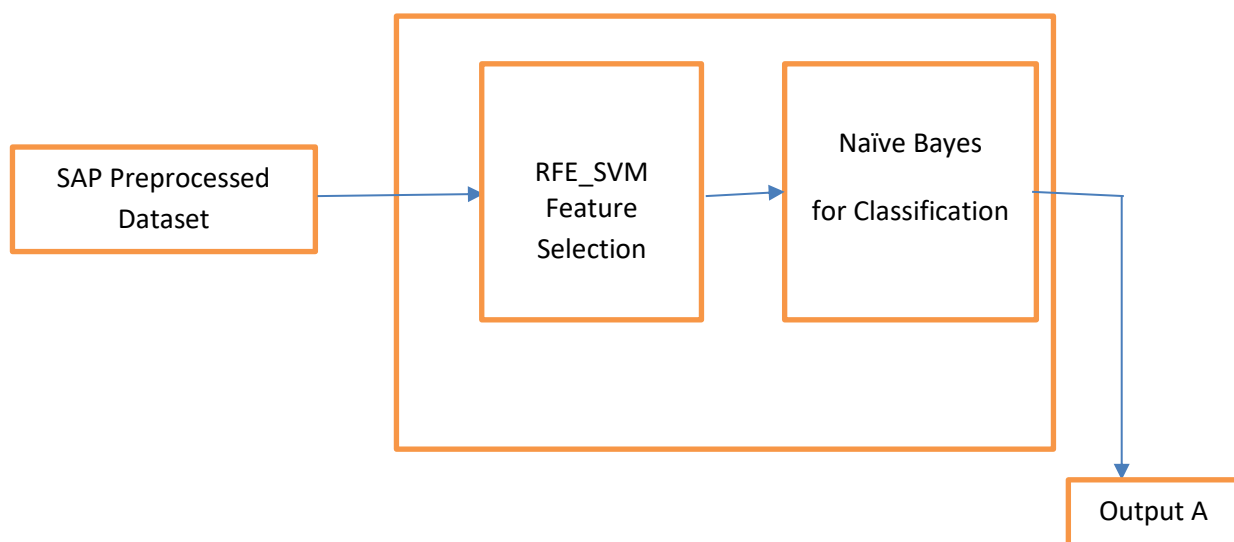


Figure 3.8: Model Architecture of the RFE_SVM-NB

The model architecture in Figure 3.8 has the RFE_SVM feature selection as its top layer while the bottom layer is the classification using the Naïve Bayes algorithm. This is the proposed SAP hybrid prediction model based on RFE and its algorithm is as shown in Figure 3.9.

Algorithm 5: SAP Hybrid Prediction Model (RFE_SVM-NB)

Input: $D = \{x_1, x_2, x_3, \dots, x_n\}$ Training dataset containing attributes $x_1 \dots x_n$ with associated class labels

Output: T – Classification Output.

Method:

1. $G = \{ \}$ Initializing storage for best feature subsets.
 2. **for** $i = 1$ to j *{for all attributes, $K_j \in D$ }* **do**
 3. compute rank, r (based on RFE using SVM)
 4. **if** $r > t$ ($t =$ threshold of SVM)
 5. Add K_j to G (K_j is the j^{th} attribute)
 6. **else**
 7. Remove K_j
 8. **end if**
 9. **end for**
 10. **for** each class $C_i \in D$, **do**
 11. Find the prior probability, $P(C_i)$
 12. **end for**
 13. **for** each attribute value $A_i \in G$ **do**
 14. Find the class conditional probability, $P(A_i/C_i)$
 15. **end for**
 16. **for** each training instances, $x_i \in G$ **do**
 17. Find the posterior probability, $P(C_i/x_i)$
 18. **if** $P(C_i/x_i)$ is maximum
 19. write T
 20. **else**
 21. go to 16
 22. **end if**
 23. **end for**
-

Figure 3.9: Algorithm for RFE_SVM-NB model

The SAP hybrid prediction model based on the GA feature selection strategy in Figure 3.10 has the GA feature selection strategy as the top layer and the same Naïve Bayes algorithm as its bottom layer for final classification.

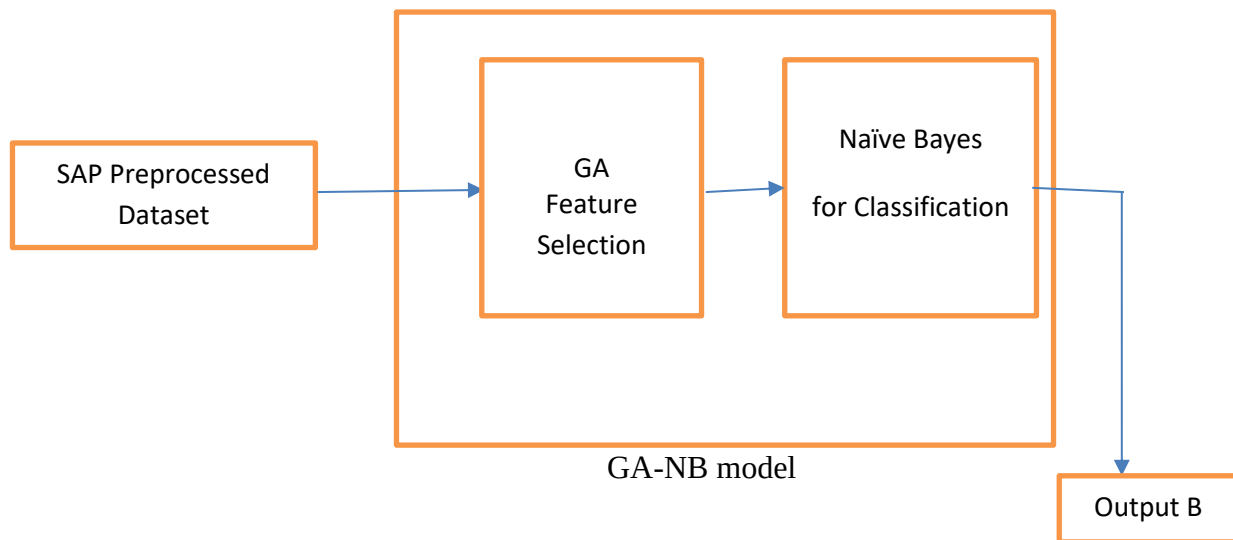


Figure 3.10: Model Architecture of the GA-NB

The algorithm for this model is as shown in Figure 3.12 which detailed the fusion of GA feature selection strategy with Naïve Bayes algorithm for SAP datasets.

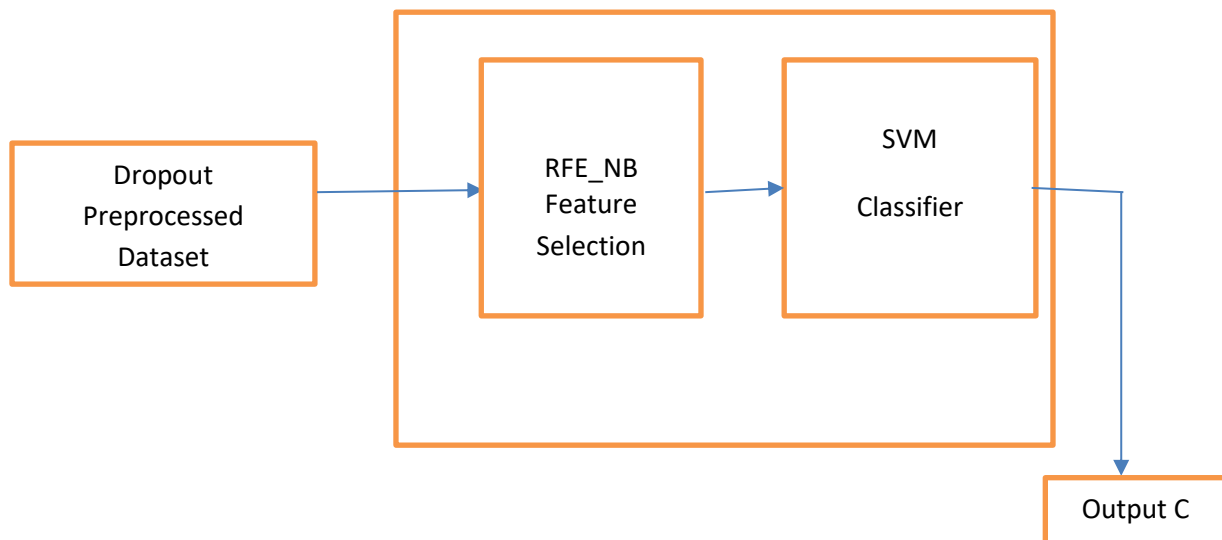


Figure 3.11: Model Architecture of the RFE_NB-SVM model

The hybrid prediction model for dropout datasets based on the RFE strategy in Figure 3.11 has the RFE feature selection strategy as the top layer and the SVM classification algorithm as the bottom layer. The algorithm for the model is represented by Figure 3.13.

Algorithm 6: SAP Hybrid Prediction Model (GA-NB)

Input: $D = \{x_1, x_2, x_3, \dots, x_n\}$ Training dataset containing attributes $x_1 \dots x_n$ with associated class labels

Output: T – Classification Output.

Method:

1. $G = \{ \}$ Initializing storage for best feature subsets.
 2. **for** $i = 1$ to j *{for all attributes, $K_j \in D$ }* **do**
 3. initialize population
 4. Assign fitness function, $f(x)$
 5. while $t \neq \text{TerminationCriteria}$
 6. **do** crossover and mutation
 7. evaluate fitness of population
 8. Go to 4
 9. **end for**
 10. **for** each class $C_i \in D$, **do**
 11. Find the prior probability, $P(C_i)$
 12. **end for**
 13. **for** each attribute value $A_i \in G$ **do**
 14. Find the class conditional probability, $P(A_i/C_i)$
 15. **end for**
 16. **for** each training instances, $x_i \in G$ **do**
 17. Find the posterior probability, $P(C_i/x_i)$
 18. **if** $P(C_i/x_i)$ is maximum
 19. write T
 20. **else**
 21. go to 16
 22. **end if**
 23. **end for**
-

Figure 3.12: Algorithm for GA-NB model

Algorithm 7: Dropout Hybrid Prediction Model (RFE_NB-SVM)

Input: $D = \{x_1, x_2, x_3, \dots, x_n\}$ Training dataset containing attributes $x_1 \dots x_n$ with associated class labels

Output: T – Classification Output.

Method:

1. $G = \{ \}$ Initializing storage for best feature subsets.
 2. **for** $i = 1$ to j {for all attributes, $K_j \in D$ } **do**
 3. compute rank, r (based on RFE Naïve Bayes)
 4. **if** $r > t$ (t = threshold for FS in Naïve Bayes)
 5. Add K_j to G (K_j is the j^{th} attribute)
 6. **else**
 7. Remove K_j
 8. **end if**
 9. **end for**
 10. **for** each class $C_i \in G$, **do**
 11. classify using SVM with RBF
 12. write T (classification output)
 13. **end for**
-

Figure 3.13: Algorithm for RFE_NB-SVM model

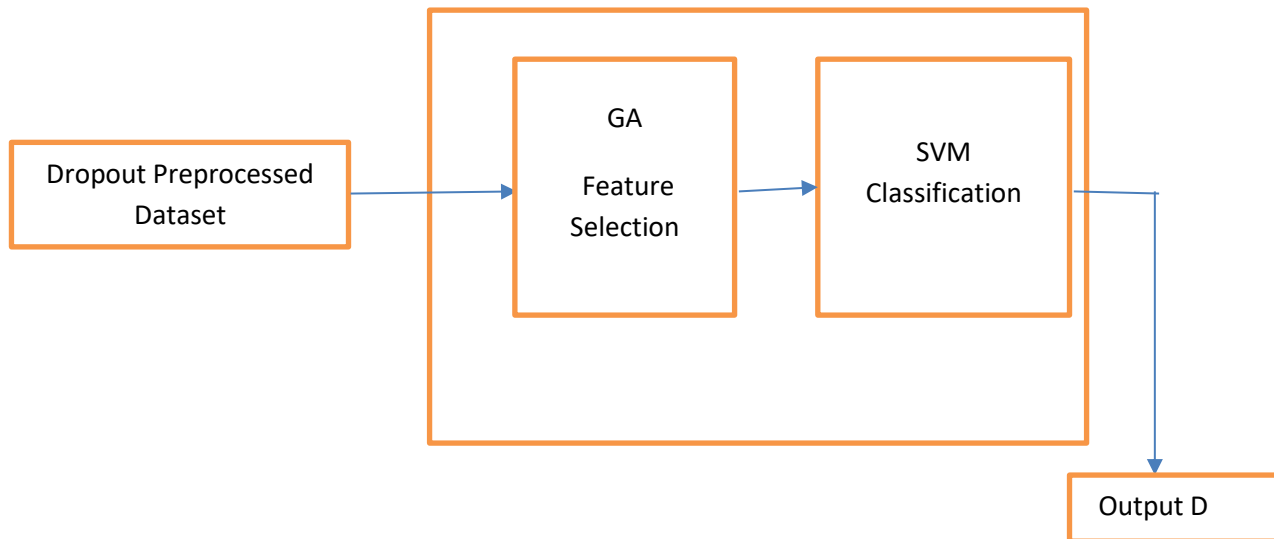


Figure 3.14: Model Architecture of the GA-SVM

In the proposed dropout hybrid models based on GA feature selection strategy as shown in Figure 3.14, the GA feature selection strategy is placed as the top layer while the SVM classifier is used as the bottom layer for final classification. Its algorithm is represented by Figure 3.15.

Algorithm 8: Dropout Hybrid Prediction Model (GA-SVM)

Input: $D = \{x_1, x_2, x_3, \dots, x_n\}$ Training dataset containing attributes $x_1 \dots x_n$ with associated class labels

Output: T – Classification Output.

Method:

1. $G = \{ \}$ Initializing storage for best feature subsets.
2. **for** $i = 1$ to j *{for all attributes, $K_j \in D$ }* **do**
3. initialize pop
4. Assign fitness function, $f(x)$
5. while $t \neq \text{TerminationCriteria}$
6. do crossover and mutation
7. evaluate fitness of pop
8. Go to 4
9. **end for**
10. **for** each class $C_i \in G$, **do**
11. classify using SVM with RBF
12. write T (classification output)
13. **end for**

Figure 3.15: Algorithm for GA-SVM model

The expected outcome of this layer is the set of classification outputs A and B for SAP datasets and C and D for Dropout datasets.

3.2.3 Model Evaluation Layer

The last layer in the design of the hybrid prediction models is the model evaluation layer. In this stage, the various models obtained were evaluated based on their Accuracy, Sensitivity (or recall), Specificity, Precision, F-measure, Geometric Mean, and Youden's Index values obtained from the various experiments conducted.

3.3 Evaluation Metrics

The performance metrics used in the model evaluation stage and their explanation are as follows.

1. **Predictive Accuracy:-** This is a metric that gives a measure for the overall accuracy of the classifier. It is mostly expressed in simple terms as the ratio of the number of correctly classified cases to the total number of cases. When computed from the confusion matrix; it is expressed as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \dots\dots\dots 3.1$$

where TP = True Positive, TN = True Negative,

FP = False Positive and FN = False Negative

2. **Precision:** This measures the positive predictive value of a classifier. The precision of class X is always expressed as the ratio of the number of correctly classified cases of class X to the number of cases classified as belonging to class X. When using values from the confusion matrix, precision is expressed as follows.

$$Precision = \frac{TP}{TP+FP} \times 100\% \dots\dots\dots 3.2$$

3. **Recall (or Sensitivity):-**It is a measure of the sensitivity of the classifier which measures the ability of a test to be positive when the condition is actually present. It is often recognized as the True Positive Rate. It is expressed as

$$Recall = \frac{TP}{TP+FN} \times 100\% \dots\dots\dots 3.3$$

The higher the value of the sensitivity of a classifier, the better the performance of the classifier.

4. **Specificity:-** This is the True Negative Rate (TNR) of a classifier which is the ratio of the correctly classified negative cases to the total number of negative cases. It is derived from the formula

$$TNR = \frac{TN}{FP+TN} \times 100\% \dots\dots\dots 3.4$$

A high value of specificity for a classifier signifies its high performance.

5. **F-measure (F1 Score)**:- This is the harmonic mean of precision and recall. It is expressed as:

$$F = \frac{2Precision \times Recall}{Precision + Recall} \dots\dots\dots 3.5$$

This measure performs well when the data is fairly balanced.

6. **Geometric Mean (GM)**:- This metric measures the balance between classification performances on both the majority and minority classes. It is computed using the formula

$$GM = \sqrt{TPR \times TNR} = \sqrt{Sensitivity \times Specificity} \dots\dots\dots 3.6$$

Low G-Mean for a classifier indicates that the classifier is poor in performance in the classification of positive cases while a high value of GM is an indication of high performance

7. **Youden's Index**:- It is a metric that evaluates the ability of a classifier to avoid misclassification. It is computed as

$$Youden's Index (YI) = sensitivity - (1 - specificity) \dots\dots\dots 3.7$$

The higher the value of YI the better the performance of the classifier.

CHAPTER FOUR

RESULTS AND DISCUSSION

This chapter contains the results of all the experiments carried out in the previous chapter towards the achievement of the goal of this study. The exploratory data analysis of the collected datasets, the results of the auto-classifier detection stage, the results of the feature selection layer, and the results of the classification layer that gives the proposed hybrid prediction models are all presented here. The chapter is concluded with the performance analysis of the models and a comparison with existing models.

4.1 Exploratory Data Analysis of the Preprocessed Data

In this section, the four datasets used are explored after being subjected to the preprocessing stage.

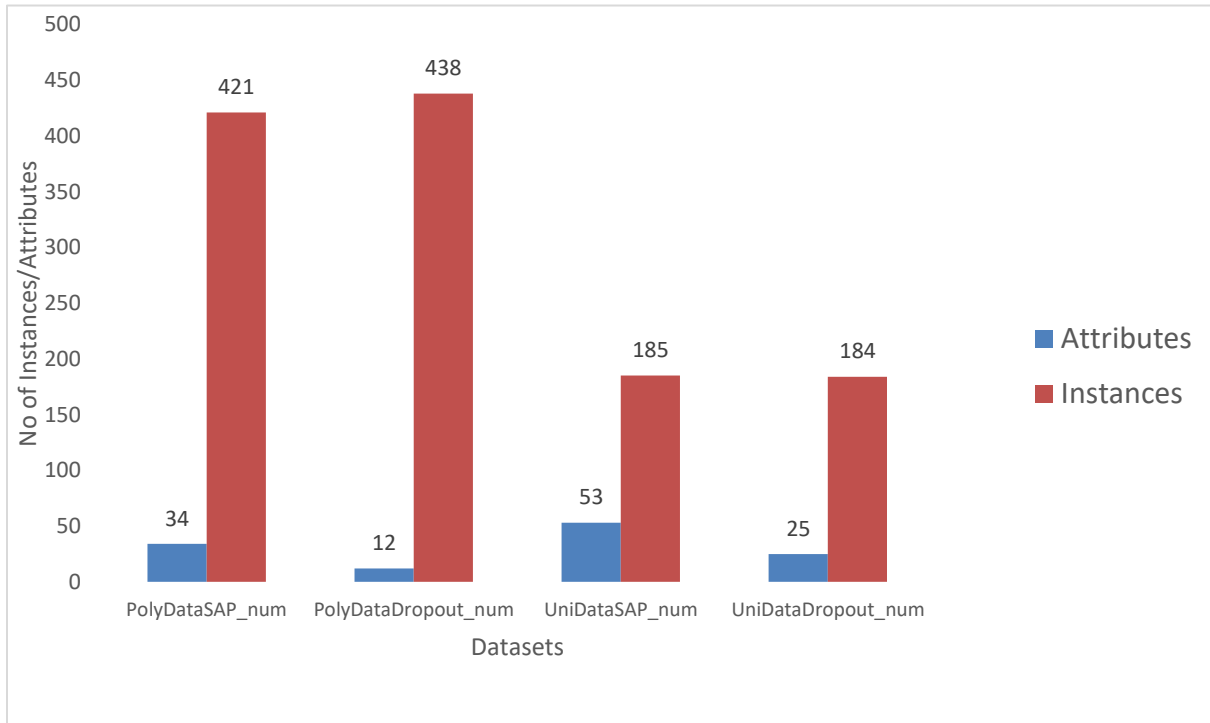


Figure 4.1: Instances-Attributes across the four datasets

Figure 4.1 shows the volume of each of the datasets used before they were being pre-processed. The PolyDataSAP_num dataset originally has 34 attributes and 421 instances, the UniDataSAP_num dataset has 53 attributes and 185 instances, the PolyDataDropout_num dataset has 12 attributes and 438 instances while the UniDataDropout_num dataset has 25 attributes and 184 instances.

Let us now proceed to visualize each of the four datasets one after the other. The essence of the visualization of the data used is to examine the degree of imbalance or otherwise of the dataset so as to link properly with the final results of classification of the models proposed in the study.

4.1.1 Exploratory Data Analysis of the UniDataSAP_num dataset

The number of instances across classes in UniDataSAP_num is shown in Figure 4.2 and the percentage distribution of classes in the dataset is depicted in Figure 4.3.



Figure 4.2: Instances versus Classes in UniDataSAP_num

The two figures revealed that there are 5 instances (corresponding to 3% of the whole instances in the dataset) for class: “Excellent”, 87 instances (47%) for class: “Very Good”, class: “Good” has 26 instances (14%) while there are 67 instances (36%) for class: “Pass”.

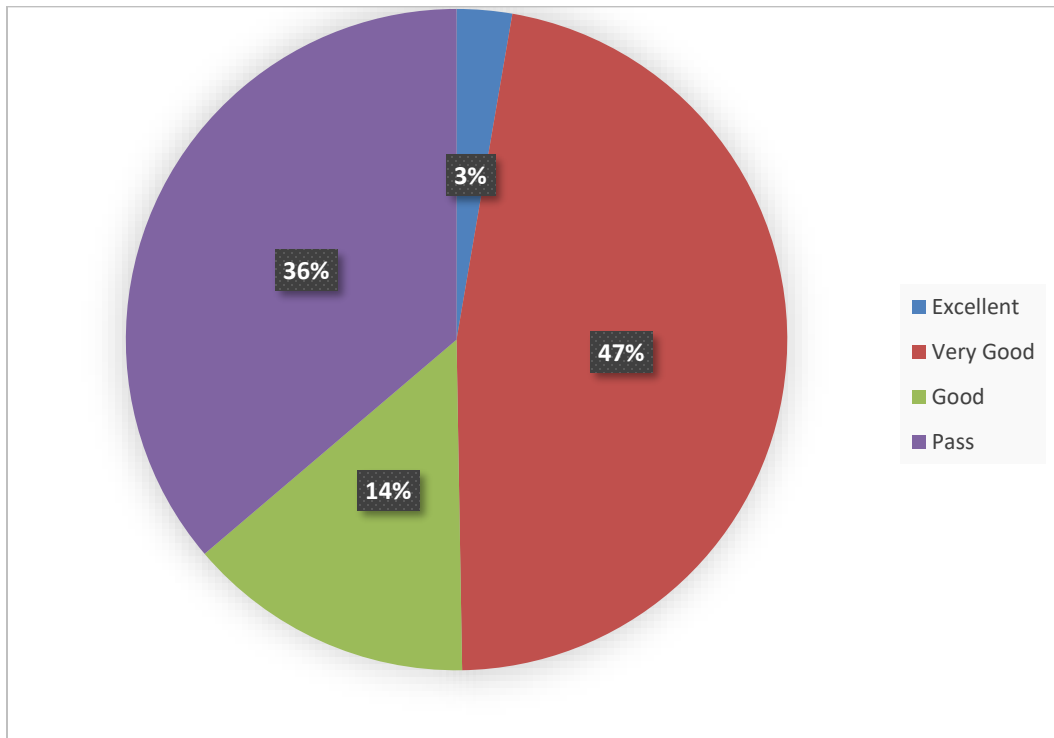


Figure 4.3: Percentage Distribution of Classes in UniDataSAP_num

4.1.2 Exploratory Data Analysis of the PolyDataSAP_num dataset

The number of instances across classes in PolyDataSAP_num is shown in Figure 4.4 and the percentage distribution of classes in the dataset is depicted in Figure 4.5.

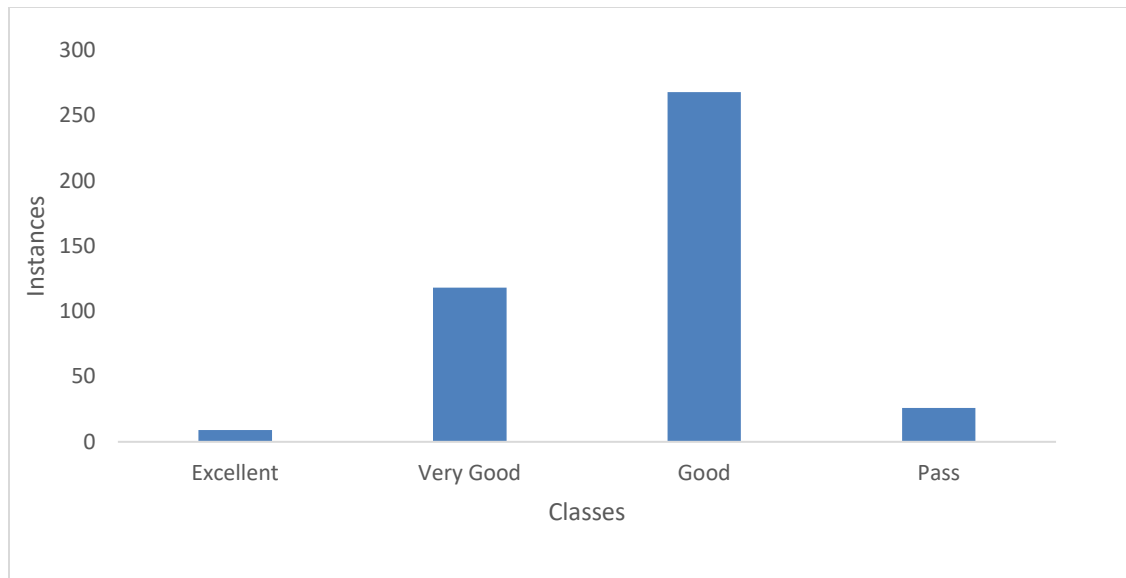


Figure 4.4: Instances versus Classes in PolyDataSAP_num

The two figures revealed that there are 9 instances (corresponding to 2% of the whole instances in the dataset) for class: “Excellent”, there are 118 instances (28%) for class: “Very Good”, class: “Good” has 268 instances (64%) while there are 26 instances (6%) for class: “Pass”.

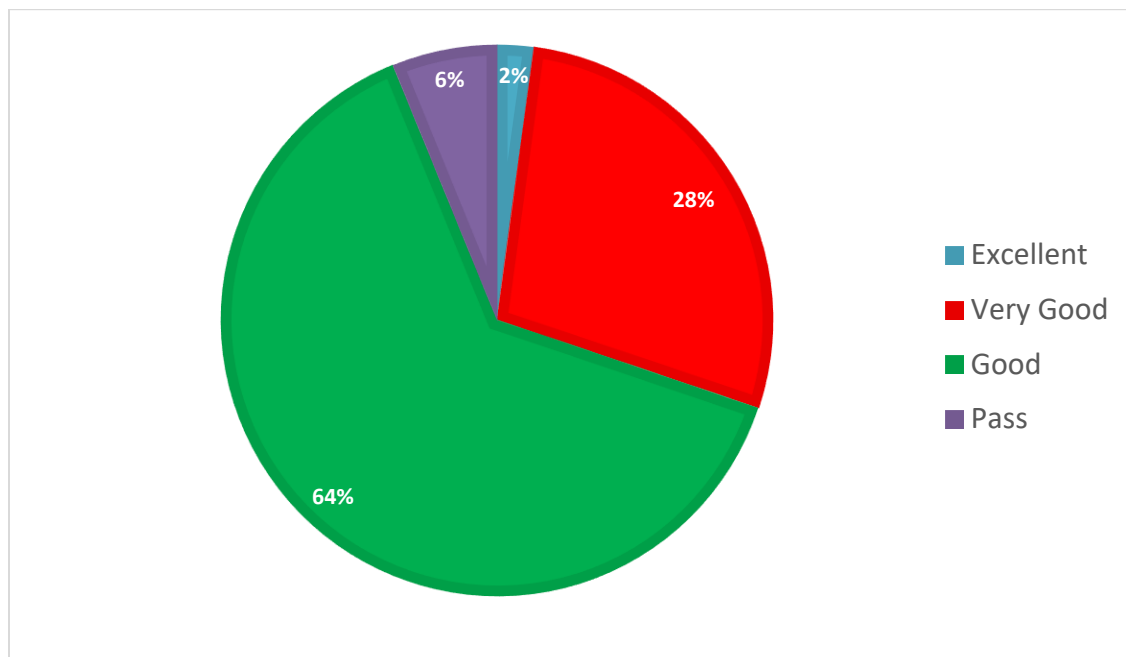


Figure 4.5: Percentage Distribution of Classes in PolyDataSAP_num

4.1.3 Exploratory Data Analysis of the UniDataDropout_num dataset

The number of instances across classes in the UniDataDropout_num dataset is shown in Figure 4.6 and the percentage distribution of classes in the dataset is depicted in Figure 4.7.

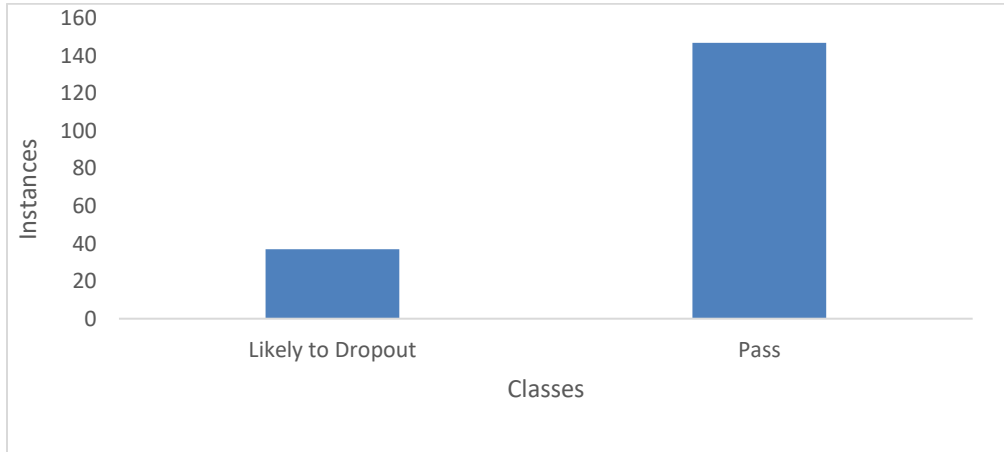


Figure 4.6: Instances versus Classes in UniDataDropout_num dataset

The two figures revealed that there are 37 instances (corresponding to 20% of the whole instances in the dataset) for class: “Likely to Dropout” and 147 instances (80%) for class: “Pass”.

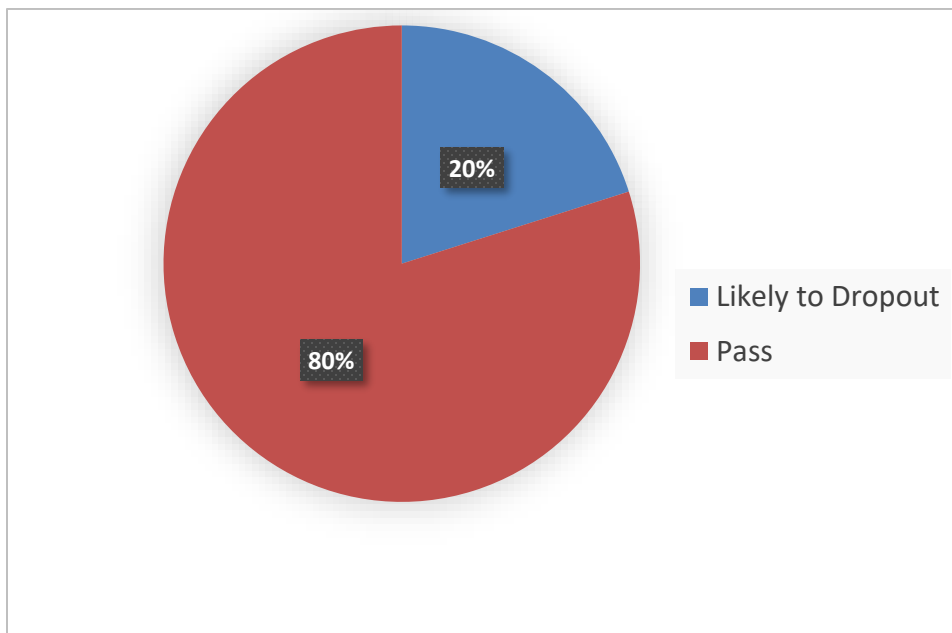


Figure 4.7: Percentage Distribution of Classes in UniDataDropout_num

4.1.4 Exploratory Data Analysis of the PolyDataDropout_num dataset

The number of instances across classes in the PolyDataDropout_num dataset is shown in Figure 4.8 and the percentage distribution of classes in the dataset is depicted in Figure 4.9.

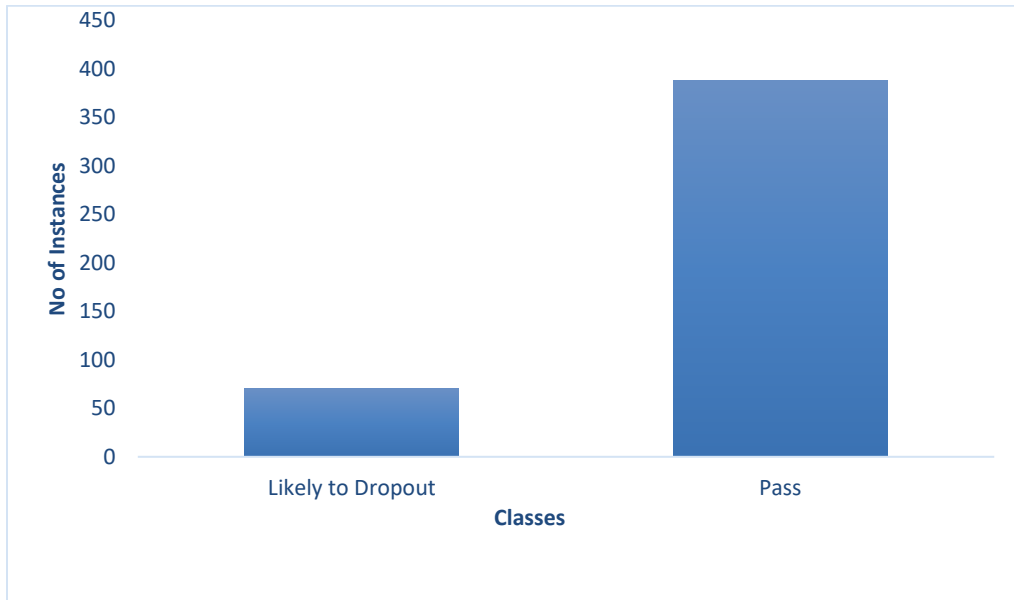


Figure 4.8: Instances versus Classes in PolyDataDropout_num

The two figures revealed that there are 70 instances (corresponding to 15% of the whole instances in the dataset) for class: “Likely to Dropout” and 388 instances (85%) for class: “Pass”.

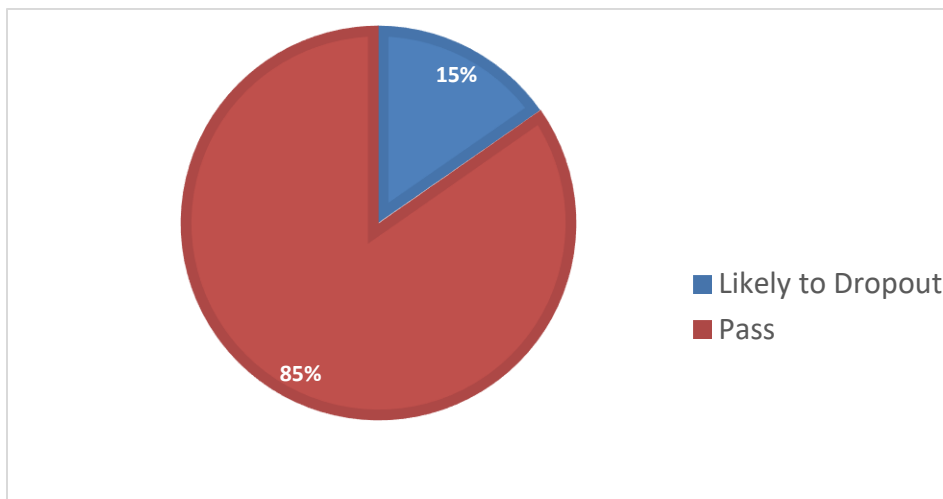


Figure 4.9: Percentage Distribution of Classes in PolyDataDropout_num

It can be seen that the distribution of classes in the four datasets has shown highly imbalanced data distribution most especially the dropout datasets where the classes are widely separated.

4.2 Results of the Auto-Classifer Detection Stage

The results of the experiments conducted on SAP and dropout datasets to determine the best classifier for the design of the hybrid prediction models are stated in this section.

4.2.1 Analysis of Results from Experiment I

The results of the experiment carried out using UniDataSAP_num dataset as summarized in Table 4.1 shows that the Naïve Bayes algorithm has the same value of predictive accuracy with SVM having radial basis function kernel though outperform all other classifiers in terms of its precision value of 0.6878. SVM with radial basis function has the best sensitivity and specificity values of 0.6933 and 0.9512 respectively while the SVM with sigmoid function kernel was the fastest to be executed (0.02 second).

Table 4.1. Results of Experiment I (with UniDataSAP_num dataset)

Performance Metrics	SVM_Linear	SVM-Polynomial	SVM_Radial	SVM_Sigmoid	Naïve Bayes
Accuracy	0.7963	0.8704	0.8889	0.4815	0.8889
Sensitivity	0.6423	0.6837	0.6933	0.2500	0.6904
Specificity	0.9211	0.9438	0.9512	0.7500	0.9496
Precision	0.6023	0.6735	0.6847	0.1204	0.6878
Exec. Time (s)	0.42	3.07	0.77	0.02	1.91

SVM with radial basis function kernel outperforms all other models of SVM on this dataset.

4.2.2 Analysis of Results from Experiment II

The results of the experiment carried out with PolyDataSAP_num dataset as summarized in Table 4.2 revealed that SVM with radial basis function kernel has the best predictive accuracy value of 83.06%, Naïve Bayes has the best sensitivity, specificity, and precision values of 0.8174, 0.9208, and 0.7852 respectively. SVM with sigmoid function kernel is still the poorest in all other metrics apart from the fact that it converges quickly (0.01 second) when compared to others.

Table 4.2. Results of Experiment II (with PolyDataSAP_num dataset)

Performance Metrics	SVM_Linear	SVM-Polynomial	SVM_Radial	SVM_Sigmoid	Naïve Bayes
Accuracy	0.7984	0.8145	0.8306	0.6452	0.8226
Sensitivity	0.6415	0.6398	0.4424	0.2500	0.8174
Specificity	0.9029	0.9043	0.9066	0.7500	0.9208
Precision	0.7398	0.7607	0.4070	0.1613	0.7852
Exec. Time (s)	0.48	4.14	0.97	0.01	1.83

SVM with radial basis function kernel outperforms all other variants of SVM on this dataset like in the first experiment having the best values in three metrics out of five among them.

4.2.3 Analysis of Results from Experiment III

Table 4.3 summarizes the results of the experiment with the UniDataDropout_num dataset. In this experiment SVM with radial basis function kernel outperforms all other models by having the highest predictive accuracy of 92.73% though shared the best precision values of 1 with SVM having polynomial function kernel. The Naïve Bayes model has the best sensitivity value of 0.8182. Three of the four SVM variants; namely: SVM with polynomial function kernel, SVM

with Radial function kernel, and SVM with sigmoid function kernel have the same specificity value of 1 each.

Table 4.3. Results of Experiment III (with UniDataDropout_num dataset)

Performance Metrics	SVM_Linear	SVM-Polynomial	SVM_Radial	SVM_Sigmoid	Naïve Bayes
Accuracy	0.9091	0.9091	0.9273	0.8000	0.8909
Sensitivity	0.6364	0.5455	0.6364	0.0000	0.8182
Specificity	0.9773	1.0000	1.0000	1.0000	0.9091
Precision	0.8750	1.0000	1.0000	NaN	0.6923
Exec. Time (s)	0.80	3.89	0.97	0.00	1.56

SVM with radial basis function kernel is, therefore, the best choice of SVM on this dataset.

4.2.4 Analysis of Results from Experiment IV

The results in Table 4.4 are obtained from the experiment using the PolyDataDropout_num dataset. In this experiment, Naïve Bayes has the best sensitivity value of 0.8571. SVM with polynomial function kernel outperformed all others in predictive accuracy and precision values of 93.43% and 0.8000 respectively.

Table 4.4. Results of Experiment IV (with PolyDataDropout_num dataset)

Performance Metrics	SVM_Linear	SVM-Polynomial	SVM_Radial	SVM_Sigmoid	Naïve Bayes
Accuracy	0.9197	0.9343	0.9124	0.8467	0.9197
Sensitivity	0.7143	0.7619	0.5714	0.0000	0.8571
Specificity	0.9569	0.9655	0.9741	1.0000	0.9310
Precision	0.7500	0.8000	0.8000	NaN	0.6923
Exec. Time (s)	0.42	1.91	1.17	0.000	0.76

SVM with sigmoid function kernel remains consistently the poorest in performance although it was the fastest to be executed (0.00 second) though had the best specificity value of 1.

4.2.5 Summary of Findings from the Auto-Classifer Detection Stage

The following are the findings from the results obtained from the Auto-Classifer detection stage.

1) The findings from experiment I are as follows.

- (i) Naïve Bayes algorithm did well with the same predictive accuracy (88.89%) as the best of the SVM and it also had the best precision value of 0.6878. However, its specificity value of 0.9496 is in close range with that of SVM having radial basis function kernel with the best specificity value of 0.9512.
- (ii) SVM with radial basis function kernel was best in sensitivity and specificity values but have the same best predictive accuracy of 0.8889 with Naïve Bayes.
- (iii) The metric values of SVM with polynomial function kernel are very close to that of SVM having radial basis function kernel. That is, it is the next best kernel among the SVM variants after the SVM with radial basis function kernel.
- (iv) SVM with the sigmoid function kernel produces a poor model based on this dataset.

The Naïve Bayes model is concluded the best on this dataset based on the above findings.

2) Experiment II revealed that

- (i) Naïve Bayes model had the best sensitivity, specificity, and precision values of 0.8174, 0.9208, and 0.7852 respectively.
- (ii) SVM with sigmoid function kernel is still not good for the prediction of SAP based on its performance in this experiment.

Naïve Bayes model is still concluded the best on this dataset based on the above findings.

3) The findings from experiment III are as follows.

- (i) SVM with radial basis function kernel had the best accuracy (0.9273), best precision, and specificity values of 1 each. It also converged faster (0.97 seconds) than NB (1.56 seconds).
- (ii) Naïve Bayes model was only better than the SVM with sigmoid function kernel judging from the values of the models' accuracies but has the best sensitivity value of 0.8182.
- (iii) SVM with sigmoid function kernel is still the poorest performed model in this experiment.

SVM with radial basis function kernel is concluded the best classifier on this dataset based on the findings.

4) Experiment IV revealed the following.

- (i) SVM with polynomial function kernel came best in terms of predictive accuracy (0.9343) and precision value of 0.8 on this dataset.
- (ii) SVM with radial basis function kernel has a predictive accuracy of 0.9124 which is close to the best value and also share the best precision value of 0.8 with SVM having polynomial function kernel.
- (iii) Naïve Bayes had the best sensitivity value of 0.8571.
- (iv) SVM with sigmoid function kernel though still the poorest performed model but with the best specificity value of 1.

SVM with polynomial function kernel is concluded the best classifier on this dataset based on the findings.

The results of the four experiments conducted showed that Naïve Bayes models of SAP have good performances and the Naïve Bayes algorithm is therefore adopted as the best classifier for SAP dataset because of the glaring performances across the metrics while SVM with radial basis function kernel performs well on dropout datasets among all the variants of SVM. SVM with radial basis function kernel was chosen because of its best performances in three out of the metrics in experiment III which makes it better than SVM with polynomial function kernel that led in two metrics in experiment IV. The choice of SVM with Radial Basis Function kernel from the findings in this study is in agreement with the studies of Oloruntoba & Akinode (2017) and Burman & Som (2019)

4.3 Results of the Proposed Hybrid Models

There are two kinds of models proposed in this study as said earlier, the RFE-based (that is the RFE_SVM-NB and the RFE_NB-SVM) and the GA-based (that is the GA-NB and the GA-SVM) models. Seven performance metrics excluding the execution time were employed in the evaluation of the performances of the proposed hybrid models as earlier explained in Section 3.3.

The results obtained from the application of the proposed hybrid models on the four datasets are presented in the following sections.

4.3.1 SAP Hybrid Prediction Models (SAPHPM)

4.3.1.1 RFE_SVM-NB models

The Recursive Feature Elimination technique using SVM as the baseline algorithm applied to the ‘UniDataSAP_num’ dataset selected 18 features out of 50 with an accuracy of 0.8098. The features are as in Figure 4.10.

```
[1] "STA221" "CSC212" "CSC314" "MAT114" "STA203" "CSC321" "CSC317"  
    "CSC220" "TCS102" "ICS101" "CSC325" "CSC213" "STA124" "MAT213"  
    "CSC112" "MAT112" "STA131" "CSC323"
```

Figure 4.10 Features selected from UniDataSAP_num using RFE_SVM strategy

The same RFE using SVM returned the following 18 features out of 32 selected when applied to the ‘PolyDataSAP_num’ dataset with an accuracy of 0.8279 as in Figure 4.11.

```
[1] "COM314" "COM322" "STA411" "COM312" "COM413" "COM415" "COM412"  
    "COM329" "COM321" "COM326" "MTH321" "COM323" "COM311" "GNS411"  
    "EED413" "COM324" "COM423" "COM414"
```

Figure 4.11 Features selected from PolyDataSAP_num using RFE_SVM strategy

The selected features were used to prepare two input files: ‘UniDataSAP_num_FS’ and ‘PolyDataSAP_num_FS’ that were later passed separately to the Naïve Bayes classifier at the bottom layer of the RFE_SVM-NB hybrid model. The performance evaluation results obtained from the SAP hybrid models based on RFE are as presented in Table 4.5.

Table 4.5 Results of RFE_SVM-NB model on SAP datasets

Metrics	UniDataSAP_num	PolyDataSAP_num
Accuracy	88.89%	90.55%
Sensitivity	0.6962	0.8808
Specificity	0.9548	0.9544
Precision	0.6630	0.9100
F-measure	0.6791	0.8952
Geometric Mean	0.8153	0.9169
Youden's Index	0.651	0.8352
Exec. Time (second)	1.77	1.99

In Table 4.5, when the performance of the RFE_SVM-NB model on the two SAP datasets was compared it could be seen that the model performed well on the PolyDataSAP_num dataset than the UniDataSAP_num dataset having higher values in six out of the seven evaluation metrics considered.

4.3.1.2 GA_NB models

The second category of the hybrid models proposed in this study for SAP prediction is that based on GA feature selection and that is the GA-NB model. The GA feature selection experiments were conducted on the 'UniDataSAP_num' and the 'PolyDataSAP_num' datasets using three different population sizes of 20, 50, and 100, and the performances of the GA-NB model were observed as shown in Table 4.6, Table 4.7, and Table 4.8 respectively.

Table 4.6: Results of GA-NB model on the two SAP datasets at population = 20

Metrics	UniDataSAP_num	PolyDataSAP_num
Accuracy	92.59%	87.9%
Sensitivity	0.6893	0.8148
Specificity	0.9659	0.9477
Precision	0.7110	0.7170
F-measure	0.7000	0.7628
Geometric Mean	0.8160	0.8787
Youden's Index	0.6552	0.7625
Exec. Time (second)	3.15	2.86

When the performances of the GA-NB model on the two SAP datasets were compared it could be seen that the model performed well on the PolyDataSAP_num dataset than the UniDataSAP_num dataset having higher values in five out of the seven evaluation metrics considered as in Table 4.6. The performances of the GA-NB model on the two SAP datasets at a population size of 50 as shown in Table 4.7 revealed that the model performed well on the PolyDataSAP_num dataset than it performed on the UniDataSAP_num dataset having higher values in four out of the seven evaluation metrics considered.

Table 4.7: Results of GA-NB model on the two SAP datasets at population = 50

Metrics	UniDataSAP_num	PolyDataSAP_num
Accuracy	92.73%	85.6%
Sensitivity	0.6966	0.8344
Specificity	0.9685	0.9425
Precision	0.7077	0.6863
F-measure	0.7021	0.7531
Geometric Mean	0.8214	0.8868
Youden's Index	0.6651	0.7769
Exec. Time (second)	3.31	2.88

When the population size was changed to 100 as in Table 4.8, there was a little improvement in the values of the metrics compared to what was obtained in Table 4.6 and Table 4.7.

It was observed that the GA-NB model was best in the classification results obtained when the population size is 100. Therefore, the experiment where the population size is 100 was considered for this model.

The GA feature selection strategy selected 36 features out of 50 features from the UniDataSAP_num dataset as in Figure 4.12 and 22 features out of 32 features from the PolyDataSAP_num dataset as in Figure 4.13.

"CSC111" "CSC112" "GNS112" "ICS101" "MAT111" "MAT112" "MAT113" "MAT114"
 "PHY115" "PHY152" "PHY192" "PLB101" "STA121" "STA124" "STA131" "TCS102"
 "CSC212" "CSC213" "CSC216" "CSC220" "CSC 224" "GNS212" "MAT211" "MAT213"
 "PHY252" "STA203" "STA221" "CSC314" "CSC315" "CSC316" "CSC317" "CSC321"
 "CSC323" "CSC325" "GNS311" "CSC431"

Figure 4.12 Features selected from UniDataSAP_num using GA strategy

“COSM311” “COM312” “COM314” “STA311” “STA314” “COM321” “COM322”
“COM323” “COM324” “COM326” “COM329” “MTH321” “STA321” “COM412”
“COM413” “COM415” “EED413” “GNS411” “STA411” “COM422” “COM423”
“COM424”

Figure 4.13 Features selected from PolyDataSAP_num using GA strategy

These features were the ones passed into the classification layer of the GA-NB model that led to the results in Table 4.8.

Table 4.8: Results of GA-NB model on the two SAP datasets at population = 100

Metrics	UniDataSAP_num	PolyDataSAP_num
Accuracy	94.55%	85.6%
Sensitivity	0.7063	0.8344
Specificity	0.9756	0.9425
Precision	0.7197	0.6863
F-measure	0.7129	0.7531
Geometric Mean	0.8301	0.8868
Youden’s Index	0.6819	0.7769
Exec. Time (second)	3.02	2.91

The model still performed better on the PolyDataSAP_num dataset by having higher values in four out of the seven evaluation metrics considered.

4.3.2 Dropout Hybrid Prediction Models

4.3.2.1 RFE_NB-SVM Model

The Recursive Feature Elimination using NB carried out on the ‘UniDataDropout_num’ dataset selected 12 features out of 22 with an accuracy of 0.8847 and the features are as shown in Figure 4.14.

```
[1] "STA121" "MAT113" "MAT112" "MAT111" "MAT114" "STA124"  
    "PLB101" "STA131" "PHY115" "ICS101" "CSC112" "TCS101"
```

Figure 4.14 Features selected from UniDataDropout_num using RFE_NB strategy

The same RFE using NB returned 7 features out of 9 selected when applied to the ‘PolyDataDropout_num’ dataset with an accuracy of 0.9227. Figure 4.15 shows the list of features selected by the RFE_NB strategy.

```
[1] "COM314" "COM312" "COM311" "GNS311" "STA314" "COM313"  
    "STA311"
```

Figure 4.15 Features selected from PolyDataDropout_num using RFE_NB strategy

The selected features were used to prepare two input files: ‘UniDataDropout_num_FS’ and ‘PolyDataDropout_num_FS’ that were later passed separately to the SVM with radial basis function kernel classifier at the bottom layer of the RFE_NB-SVM hybrid model. The performance evaluation results obtained for the hybrid RFE_NB-SVM model on the two dropout datasets are presented in Table 4.9.

Table 4.9 Results of RFE_NB-SVM model on the Dropout datasets

Metrics	UniDataDropout_num	PolyDataDropout_num
Accuracy	92.73%	93.43%
Sensitivity	0.9091	0.8095
Specificity	0.9318	0.9569
Precision	0.7692	0.7727
F-measure	0.8333	0.7907
Geometric Mean	0.9204	0.8801
Youden's Index	0.8409	0.7664
Exec. Time (second)	2.31	1.08

In Table 4.9, there is a reverse in what had been occurring in the cases of SAP datasets in that the performance of the RFE_NB-SVM model on the two dropout datasets when compared showed that the model performed well on UniDataDropout_num dataset than the PolyDataDropout_num dataset having higher values in four out of the seven evaluation metrics considered.

4.3.2.2 GA-SVM Models

The second category of the hybrid models proposed for dropout prediction in this study is that based on GA feature selection and in this case it is the GA-SVM model. The GA feature selection experiments were conducted on the UniDataDropout_num and the PolyDataDropout_num datasets using different population sizes of 20, 50, and 100, and the performance of the GA-SVM models was observed as in Table 4.10, Table 4.11, and Table 4.12 respectively.

Table 4.10: Results of GA-SVM model on the Dropout datasets at population = 20

Metrics	UniDataDropout_num	PolyDataDropout_num
Accuracy	92.73%	92.7%
Sensitivity	0.6364	0.7143
Specificity	1	0.9655
Precision	1	0.7895
F-measure	0.7778	0.7500
Geometric Mean	0.7978	0.8305
Youden's Index	0.6364	0.6798
Exec. Time (second)	1.10	1.17

When the performance of the GA-SVM model on the two dropout datasets was compared it could be seen that the model performed well on the UniDataDropout_num dataset than the PolyDataDropout_num dataset having higher values in four out of the seven evaluation metrics considered as in Table 4.10.

The performance of the GA-SVM model on the two dropout datasets at a population size of 50 as shown in Table 4.11 revealed that the model performed well on the PolyDataDropout_num dataset than it performed on the UniDataDropout_num dataset having higher values in five out of the seven evaluation metrics considered.

When the population size was changed to 100 as in Table 4.12, there was a little improvement in the values of the metrics compared to what was obtained in Table 4.10 and Table 4.11 most significantly on the UniDataDropout_num dataset.

Table 4.11: Results of GA-SVM model on the Dropout datasets at population = 50

Metrics	UniDataDropout_num	PolyDataDropout_num
Accuracy	87.27%	92.7%
Sensitivity	0.4546	0.7143
Specificity	0.9773	0.9655
Precision	0.8333	0.7895
F-measure	0.5883	0.7500
Geometric Mean	0.6665	0.8305
Youden's Index	0.4319	0.6798
Exec. Time (second)	1.14	1.13

It was observed that the GA-SVM model was best in the classification results obtained when the population size is 100. Therefore the experiment where the population size is 100 was considered for this model.

The GA feature selection strategy selected 14 features out of 22 features from the UniDataDropout_num dataset as in Figure 4.16 and 6 features out of 9 features from the PolyDataDropout_num dataset as shown in Figure 4.17.

"CSC111" "CSC112" "MAT111" "MAT112" "MAT113" "PHY115" "PHY152"
"PHY192" "PLB101" "STA121" "STA124" "STA131" "TCS101" "TCS102"

Figure 4.16 Features selected from UniDataDropout_num using GA strategy

"SEX" "COM311" "COM312" "COM314" "GNS311" "STA314"

Figure 4.17 Features selected from PolyDataDropout_num using GA strategy

These were the features passed into the classification layer of the GA-NB model that led to the results in Table 4.12.

Table 4.12: Results of GA-SVM model on the Dropout datasets at population = 100

Metrics	UniDataDropout_num	PolyDataDropout_num
Accuracy	89.09%	92.7%
Sensitivity	0.4546	0.7143
Specificity	1	0.9655
Precision	1	0.7895
F-measure	0.6251	0.7500
Geometric Mean	0.6742	0.8305
Youden's Index	0.4546	0.6798
Exec. Time (second)	0.97	1.09

However, when the performances in Table 4.12 are observed, the model still performed better on the PolyDataDropout_num dataset by having higher values in five out of the seven evaluation metrics considered.

4.4 Performance Analysis of the Proposed Hybrid Models

The performance analysis of the proposed hybrid models is explained in this section with respect to the comparison of the two models against each other and their comparison with the single classifiers of NB and SVM in the Auto-Classifer Detection stage of the models' design.

4.4.1 Analysis of SAPHPM

4.4.1.1 Comparison of the Two Categories of the Proposed Hybrid SAP models

The two categories of hybrid SAP models proposed in this study are the RFE_SVM-NB models and the GA-NB models. Table 4.13 and Table 4.14 give the results for the comparison.

Table 4.13 Results of the Comparison between the two hybrid models on UniDataSAP_num

Metrics	RFE_SVM-NB	GA-NB
Accuracy	88.89%	94.55%
Sensitivity	0.6962	0.7063
Specificity	0.9548	0.9756
Precision	0.6630	0.7197
F-measure	0.6791	0.7129
Geometric Mean	0.8153	0.8301
Youden's Index	0.6510	0.6819
Exec Time (second)	1.77	3.02

The higher values of the GA-NB model in all the seven performance metrics employed on the UniDataSAP_num dataset as in Table 4.13 showed that the GA-NB model glaringly outperformed the RFE_SVM-NB model on the UniDataSAP_num dataset but the RFE-SVM-NB model converged faster than the GA_NB model.

The results in Table 4.14 however, showed a better performance of the RFE_SVM-NB model over the GA-NB model on the PolyDataSAP_num dataset as shown by its higher metric values.

Table 4.14 Results of the Comparison between the two hybrid models on PolyDataSAP_num

Metrics	RFE_SVM-NB	GA-NB
Accuracy	90.55%	85.6%
Sensitivity	0.8808	0.8344
Specificity	0.9544	0.9425
Precision	0.9100	0.6863
F-measure	0.8952	0.7531
Geometric Mean	0.9169	0.8868
Youden's Index	0.8352	0.7769
Exec Time (second)	1.99	2.91

4.4.1.2 Comparison of the Two Categories of the Proposed Hybrid SAP models with Single Classifiers

When the results of the two hybrid SAP models in this study were compared with the results of the single classifiers from the Auto-Classifier Detection stage of the model design, the results were as presented in Table 4.15 and Table 4.16.

It was observed that the GA-NB model outperformed all the single classifiers of NB and SVM models of the Auto-Classifier Detection stage on the UniDataSAP_num dataset by having the best values in all the seven metrics considered as could be seen in Table 4.15. The only exception is that the SVM model converged faster than the two hybrid models.

Table 4.15 Results of the Comparison between the two hybrid models and the single classifiers on UniDataSAP_num

Metrics	NB	SVM	RFE_SVM-NB	GA-NB
Accuracy	88.89%	88.89%	88.89%	94.55%
Sensitivity	0.6904	0.6933	0.6962	0.7063
Specificity	0.9496	0.9512	0.9548	0.9756
Precision	0.6878	0.6847	0.6630	0.7197
F-measure	0.6891	0.6890	0.6791	0.7129
Geometric Mean	0.8097	0.8121	0.8153	0.8301
Youden's Index	0.6400	0.6445	0.6510	0.6819
Exec Time (second)	1.91	0.77	1.77	3.02

In the case of the PolyDataSAP_num dataset, the RFE_SVM-NB model glaringly outperformed all the single classifiers in all the seven metrics considered as in Table 4.16.

Table 4.16 Results of the Comparison between the two hybrid models and the single classifiers on PolyDataSAP_num

Metrics	NB	SVM	RFE_SVM-NB	GA-NB
Accuracy	82.26%	83.06%	90.55%	85.6%
Sensitivity	0.8174	0.4424	0.8808	0.8344
Specificity	0.9208	0.9066	0.9544	0.9425
Precision	0.7852	0.4070	0.9100	0.6863
F-measure	0.8010	0.4240	0.8952	0.7531
Geometric Mean	0.8676	0.6333	0.9169	0.8868
Youden's Index	0.7382	0.3490	0.8352	0.7769
Exec Time (second)	1.83	0.97	1.99	2.91

The results obtained in this section revealed that the GA-NB model performed better on the UniDataSAP_num dataset while the PolyDataSAP_num dataset is favoured by the RFE_SVM-NB model for both the comparative analysis between the SAP hybrid models and with the single classifiers.

4.4.2 Analysis of DHPM

There are four proposed hybrid models for Dropout predictions in this study, namely: the two RFE_NB-SVM models and the two GA-SVM models owing to the two sources of datasets.

4.4.2.1 Comparison of the Two Categories of the Proposed Hybrid Dropout models

In Table 4.17, the RFE_NB_SVM model outperformed the GA_SVM model on the UniDataDropout_num dataset in five out of seven metrics considered in this study.

Table 4.17 Results of the Comparison between the two hybrid models on UniDataDropout_num

Metrics	RFE_NB-SVM	GA-SVM
Accuracy	92.73%	89.09%
Sensitivity	0.9091	0.4546
Specificity	0.9318	1
Precision	0.7692	1
F-measure	0.8333	0.6251
Geometric Mean	0.9204	0.6742
Youden's Index	0.8409	0.4546
Exec Time (second)	2.31	0.97

The same scenario that happened with the UniDataDropout_num dataset (Table 4.17) repeats itself on the PolyDataDropout_num dataset with the RFE_NB-SVM model outperforming the GA-SVM model in five out of the seven metrics considered in the study as in Table 4.18.

Table 4.18 Results of the Comparison between the two hybrid models on PolyDataDropout_num

Metrics	RFE_NB-SVM	GA-SVM
Accuracy	93.43%	92.7%
Sensitivity	0.8095	0.7143
Specificity	0.9569	0.9655
Precision	0.7727	0.7895
F-measure	0.7907	0.7500
Geometric Mean	0.8801	0.8305
Youden's Index	0.7664	0.6798
Exec Time (second)	1.08	1.09

4.4.2.2 Comparison of the Two Categories of the Proposed Hybrid Dropout models with Single Classifiers

Table 4.19 and Table 4.20 show the results of comparing the two hybrid models with the single classifiers from the Auto-Classifer Detection stage on both UniDataDropout_num and PolyDataDropout_num datasets.

In Table 4.19, the RFE_NB-SVM model outperformed all the single classifiers by taking the lead in four metrics out of seven metrics considered in the study. The SVM model only outperformed the two hybrid models in terms of specificity and precision apart from the fact that it was the quickest to converge.

Table 4.19 Results of the Comparison between the two hybrid models and the single classifiers on UniDataDropout_num

Metrics	NB	SVM	RFE_NB-SVM	GA-SVM
Accuracy	89.09%	92.73%	92.73%	89.09%
Sensitivity	0.8182	0.6364	0.9091	0.4546
Specificity	0.9091	1	0.9318	1
Precision	0.6923	1	0.7692	1
F-measure	0.7500	0.7778	0.8333	0.6251
Geometric Mean	0.8625	0.7978	0.9204	0.6742
Youden's Index	0.7273	0.6364	0.8409	0.4546
Exec Time (second)	1.56	0.97	2.31	0.97

A different pattern of performances revealed itself when comparing the two hybrid models with the single classifiers in the Auto-Classifier Detection stage on the PolyDataDropout_num dataset as shown in Table 4.20.

The RFE_NB-SVM model outperformed the single classifiers in two metrics out of the seven metrics like the SVM model which also led in two metrics. NB classifier also performed better than the two proposed hybrid models in three metrics out of the seven metrics considered and it even converged faster (0.76 seconds).

The performance of the proposed hybrid models are therefore well compared to that of the single classifiers of NB and SVM in the Auto-Classifier Detection stage of the study, however, the hybrid models are superior in the fact that the model complexities were highly reduced with the feature selection strategies embedded.

Table 4.20 Results of the Comparison between the two hybrid models and the single classifiers on PolyDataDropout_num

Metrics	NB	SVM	RFE_NB-SVM	GA-SVM
Accuracy	91.97%	91.24%	93.43%	92.7%
Sensitivity	0.8571	0.5714	0.8095	0.7143
Specificity	0.9310	0.9741	0.9569	0.9655
Precision	0.6923	0.8000	0.7727	0.7895
F-measure	0.7659	0.6667	0.7907	0.7500
Geometric Mean	0.8933	0.7461	0.8801	0.8305
Youden's Index	0.7881	0.5455	0.7664	0.6798
Exec Time (second)	0.76	1.17	1.08	1.09

The RFE_NB-SVM model outperformed all other classifiers in both the comparative analysis between the dropout hybrid models and with the single classifiers except in the case of the PolyDataDropout_num dataset where the two hybrid models performed comparably well with the single classifiers of NB and SVM.

However, in the overall assessment from the four datasets, the RFE-based models showed their better performances over the GA-based models. However, the four proposed hybrid models for SAP predictions, namely: the two RFE_SVM-NB models and the two GA-NB models performed better than the single applications of each of the NB and SVM classifiers in the Auto-Classifer Detection stage judging from their performance metrics values.

4.5 Comparative Analysis of the Proposed Models with Existing Models

The results obtained for the hybrid models were compared with the existing models in the literature, two of which were considered in this study. They are Burman & Som (2019) and Amra & Maghari (2017). These two existing models in the literature used university students' datasets and so the results of the proposed models of the UniDataSAP_num dataset in this study are used for the comparison. The two existing models chosen for comparison are single classifiers and this is because, there is no hybrid model of the two DM algorithms (SVM and NB) considered in this study in existence, to the best of our knowledge.

The compared results are as shown in Table 4.21 and Table 4.22.

Table 4.21: Results of Comparison of Hybrid models with Burman & Som (2019)

Metrics	Burman & Som (2019)	RFE_SVM-NB	GA-NB
Accuracy	90.74%	88.89%	94.55%
Sensitivity	0.6797	0.6962	0.7063
Specificity	0.9585	0.9548	0.9756
Precision	0.6982	0.6630	0.7197
F-measure	0.6888	0.6791	0.7129
Geometric Mean	0.8072	0.8153	0.8301
Youden's Index	0.6382	0.651	0.6819
Exec Time (seconds)	1.32	1.77	3.02

The proposed hybrid GA-NB model overwhelmingly outperformed Burman & Som (2019) model by having the highest values in all the seven metrics considered. The proposed hybrid RFE_SVM-NB model also outperformed Burman & Som (2019) in terms of the values of the sensitivity (0.6962), Geometric Mean (0.7967), and Youden's Index (0.651) while Burman & Som (2019)

outperformed it in terms of the predictive accuracy, specificity, precision and F-measure values which were within very close range. This makes the RFE_SVM-NB model to be comparable to Burman & Som (2019).

This is still sustained even though the Burman & Som (2019) model converged faster (1.32 seconds) than the two proposed hybrid models. This could be due to the fact that Burman & Som (2019) model is a single classifier model.

The proposed hybrid GA-NB model maintains its overwhelming performances when compared with Amra & Maghari (2017) by having the highest values in all the seven metrics considered as shown in Table 4.22. RFE_SVM-NB model also has a higher value of sensitivity (0.6962) than Amra & Maghari (2017) with the values of other metrics keeping a close margin with the existing model.

The two proposed models in this study also converged faster than Amra & Maghari (2017) model and the reason is that it uses two classifiers (NB and KNN) and also does not incorporate feature optimization like in the case of the proposed models.

Table 4.22: Results of Comparison of Hybrid models with Amra & Maghari (2017)

Metrics	Amra & Maghari (2017)	RFE_SVM_NB	GA-NB
Accuracy	92.59%	88.89%	94.55%
Sensitivity	0.6922	0.6962	0.7063
Specificity	0.9675	0.9548	0.9756
Precision	0.7077	0.6630	0.7197
F-measure	0.7000	0.6791	0.7129
Geometric Mean	0.8184	0.8153	0.8301
Youden's Index	0.6597	0.651	0.6819
Exec. Time (second)	4.06	1.77	3.02

In the overall assessment, the proposed hybrid GA-NB model is more efficient than the two existing models considered for comparison in this study while the RFE_SVM-NB model performances were comparable with the existing models. The most significant achievement of the proposed hybrid prediction models in this study is that they were able to reduce the model complexity through the feature selection strategy incorporated.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATIONS

This chapter gives the summary of all the findings from the study, the conclusion arrived at, and list suggestions for future work with respect to the outcome of this study.

5.1 Summary

Students' Academic Performance (SAP) and Dropout predictions are two important measures of the academic progress of a student in an institution and were examined in this study. The degree of success of a student becomes higher as the possibility of getting accurate predictions of his academic status becomes higher, most especially in measuring the dropout rate to allow necessary strategies to be put in place by all stakeholders so that the uncalled for (that is dropping out) can be averted.

The search for improved classification tool to accurately classify students' performances will continue to be one of the major preoccupations of the EDM researchers simply because there is the dare need for the avoidance of wrong classification of the performances of students to avert possible devastating effects it can have on students, tutors/level advisers, parents and other stakeholders in the educational institutions and examination bodies.

The framework of using data mining algorithms or general machine learning techniques to predict students' academic performances and dropout has been presented in this study with two categories of hybrid models developed for both SAP and dropout predictions.

The study found out that SVM with radial basis function kernel is a good tool for predicting SAP and therefore its performance on SAP datasets was excellent when an investigative study and analysis of both the Support Vector Machines and Naïve Bayes algorithms were carried out. Another most important deduction from the outcome of the initial investigation of SVM and NB

in our auto-classifier detection stage is that the study was able to assign one to the top layer of the proposed hybrid model design and the other to the bottom layer based on the scientific facts from the findings.

These scientific facts from the investigative study (that is the Auto-Classifier Detection stage) include the facts that the Naïve Bayes algorithm performed very well on all the datasets used but having noticed the high performance of SVM on the dropout dataset, the study adopted SVM for classifying dropout datasets and Naïve Bayes for classifying SAP datasets.

The study then proposed Recursive Feature Elimination (RFE) based hybrid models and Genetic Algorithm based hybrid models, the two of which performed well in most cases when compared with the single classifier examined in the study.

The RFE-based models performed better in the overall assessment than the GA-based models judging from the accumulation of the performances from the four datasets used in the study and the reason for this is not far-fetched other than for the fact that the RFE-based hybrid models contain two learning algorithms of NB and SVM as a pure hybrid of NB and SVM using recursive feature elimination technique, unlike the GA-based models which have either NB or SVM classifier and not the two.

The GA-based models have the best performances in feature selection layers of the proposed model design with all the selected features well distributed across the four semesters for better representation of the academic performance of a student in a session. The reason for this is because GA is a population-based feature selection and an adaptive learning technique.

Ultimately, the GA-based models exhibit better performance when compared with the existing models and the RFE-based models are well comparable with the existing models in the literature.

5.2 Conclusion

This study had successfully presented two categories of hybrid models developed for each of the four datasets employed. The feature selection component of the hybrid models is always at the top layer while one of the SVM and NB that best classify the type of datasets (SAP or Dropout) is added at the bottom layer of the models' design.

There were basically lot of enhancement by the hybrid models in most of the cases when all the metrics were considered and in the few cases where enhancement is not too glaring the performances were kept at close range thereby indicating that the use of the proposed hybrid models in such cases is still very promising. This is because the model complexity is reduced for all the cases of the hybrid models since the data dimensionality reduction is ensured leading to a reduced computational cost and time. The proposed models were still able to maintain their high accuracies despite the achieved reduction in the computation cost and time of the models.

The results of this study imply that there is a great possibility of higher efficiency in the hybrid prediction models when compared to the existing single classifier models although the class imbalance in the dataset must be properly taken care of. The possible reason for the closeness of the results of the hybrid prediction models where the performance is not overwhelming with that of the existing model may be due to the high degree of class imbalances that existed in the datasets used.

The validity of the results of this study could be threatened by the introduction of noise to the datasets, the use of any function kernel for SVM implementation other than the Radial Basis Function kernel that had been chosen for the study, and careless extraction of features after feature selection processes. In order to safeguard these, some measures were taken on the two empirical components of this study - dataset and the machine learning algorithms (SVM, NB, and GA) used

in the development of the hybrid models. These measures include the refinement of the data used through data cleaning during preprocessing stage so as to maintain the originality of all data values used as well as keeping constant the set parameters of all the algorithms used in the study in all the experiments conducted. Feature extraction from the original datasets after the feature selection process was also carefully done before final classification.

5.3 Recommendations

This study recommends that the datasets to be used in the testing or application of the various hybrid models proposed should be properly cleaned and well pre-processed.

The study made use of only the course scores principally as the dominating academic features for the development of the models, it is recommended that other influential features that can contribute to SAP and dropout predictions like economic features and socio-behavioural features should be incorporated into the datasets in future experiments.

Stakeholders in the educational sectors are also enjoined to take the SAP and dropout predictions as an important component of every guidance and counseling centre in our various educational institutions.

Lastly, the framework in this study is recommended for prototyping and further deployment in our educational institutions for effective classification of students' academic performances. In essence, the course tutors and/or level advisers should see the data mining tool of this nature as an important component of their offices.

5.4 Contributions to Knowledge

This study has contributed the following to the body of knowledge in the field of Educational Data Mining (EDM), most importantly in the area of SAP and dropout predictions.

- (i) This study is among the few researches in the area that have employed more than one performance metric (predictive accuracy) in the decision on classifiers' performances. It employed other metrics like sensitivity (or recall), specificity, precision, F-measure, Geometric Mean, and Youden's Index. The majority of the studies in the area based their judgement on classifiers' performances on the predictive accuracy alone.
- (ii) This study combined the two areas of SAP and Dropout in the EDM research circle using their strong relationship to aid model design, and this has made it to be novel in the field.
- (iii) This study developed a hybrid RFE_SVM-NB prediction model for Students' Academic Performance (SAP) using the Recursive Feature Elimination technique.
- (iv) This study also developed a hybrid RFE_NB-SVM prediction model for dropout using the Recursive Feature Elimination technique.
- (v) This study also has to its credit; the development of a hybrid GA-NB model for SAP prediction using the Genetic Algorithm feature selection optimization technique.
- (vi) This study also contributed a hybrid GA-SVM model for dropout prediction using the Genetic Algorithm feature selection optimization technique.
- (vii) Lastly, the study had another novel contribution in terms of the datasets used which were the creation of the researcher.

5.5 Suggestions for Further Studies

This study can be enhanced in different ways, among which are the following.

- (i) The effect of the inclusion of both socio-economic and behavioural features on the performances of SAP and dropout models is a prospective area to dwell on.
- (ii) The GA-based hybrid model can further be optimized by the use of modified GA since this study employed the simple GA feature selection technique.
- (iii) The wrapper technique-oriented RFE-based hybrid can be optimized to reduce the accompanied high consumption time of wrapper procedures.
- (iv) Better data refinement and composition can be used to obtain more efficient models that can be translated into prototypes for deployment in various educational institutions.
- (v) The two proposed models in this study can also be extended to other problem domains.

REFERENCES

- Abikoye, O. C., Omokanye, S. O., & Aro, T. O. (2017). Binary Text Classification Using an Ensemble of Naïve Bayes and Support Vector Machines. *GESJ: Computer Science and Telecommunications*, 2(52), 37–45.
- Abu Zohair, M. L. (2019). Prediction of Student's Performance by Modelling Small Dataset Size. *International Journal of Educational Technology in Higher Education*, 16(27), 1–18.
- Adejo, O. W., & Connolly, T. (2018). Predicting Student Academic Performance Using Multi-model Heterogeneous Ensemble Approach. *Journal of Applied Research in Higher Education*, 10(1), 61–75. <https://doi.org/10.1108/JARHE-09-2017-0113>
- Aden, A. A., Yahye, Z. A., & Dahir, A. M. (2013). The Effect of Student's Attendance on Academic Performance: A Case Study At Simad University Mogadishu. *Academic Research International*, 4(6), 409–417.
- Agarwal, R. (2015). Genetic Algorithm in Data Mining. *International Journal of Advanced Research in Computer Science and Software Engineering*, 5(9), 631–634.
- Agarwal, S., Pandey, G. N., & Tiwari, M. D. (2012). Data Mining in Education : Data Classification and Decision Tree Approach. *International Journal of E-Education, e-Business, e-Management and e-Learning*, 2(2), 140–145. <https://doi.org/10.1080/13604819608713449>
- Agbedo, O. (2019, May 25). Real Reasons Nigerian Youths are Committing Suicide. *The Guardian Saturday Magazine*. Retrieved from <https://guardian.ng/saturday-magazine/cover/real-reasons-nigerian-youths-are-committing-suicide/>
- Aggarwal, C. C. (2014). An Introduction to Data Classification. In *Data Classification: Algorithms and Applications* (pp. 1–36). <https://doi.org/10.1201/b17320>
- Aggarwal, C. C. (2018). Neural Networks and Deep Learning. In *Artificial Intelligence*. <https://doi.org/10.1201/b22400-15>
- Agrawal, S., Vishwakarma, S. K., & Sharma, A. K. (2017). Using Data Mining Classifier for Predicting Student's Performance in UG Level. *International Journal of Computer Applications*, 172(8), 39–44. <https://doi.org/10.5120/ijca2017915201>
- Ahmad, F., Ismail, N. H., & Aziz, A. A. (2015). The Prediction of Students ' Academic Performance Using Classification Data Mining Techniques. *Applied Mathematical Sciences*, 9(129), 6415–6426.

- Akinola, O. S., Akinkunmi, B. O., & Alo, T. S. (2012). A Data Mining Model for Predicting Computer Programming Proficiency of Computer Science Undergraduate Students. *Afr J Comp & ICT*, 5(1), 43–52.
- Akosa, J. S. (2017). Predictive Accuracy : A Misleading Performance Measure for Highly Imbalanced Data. *SAS Global Forum*, 942, 1–12.
- Al-barrak, M. A., & Al-razgan, M. (2016). Predicting Students Final GPA Using Decision Trees : A Case Study. *International Journal of Information and Education Technology*, 6(7), 528–533. <https://doi.org/10.7763/IJIET.2016.V6.745>
- Ali, S., Haider, Z., Munir, F., Khan, H., & Ahmed, A. (2013). Factors Contributing to the Students Academic Performance: A Case Study of Islamia University Sub-Campus. *American Journal of Educational Research*, 1(8), 283–289. <https://doi.org/10.12691/education-1-8-3>
- Altaher, A., & Barukab, O. M. (2018). An Intelligent Hybrid Approach for Predicting the Academic Performance of Students using Genetic Algorithms and Neuro Fuzzy System. *International Journal of Computer Science and Network Security*, 18(10), 64–70.
- Alvarez, N. L., Callejas, Z., & Griol, D. (2020). Predicting Computer Engineering Students' Dropout in Cuban Higher Education with Pre-Enrollment and Early Performance Data. *Journal of Technology and Science Education*, 10(2), 241–258.
- Alyahyan, E., & Düşteğör, D. (2020). Predicting Academic Success in Higher Education: Literature Review and Best Practices. *International Journal of Educational Technology in Higher Education*, 17(1). <https://doi.org/10.1186/s41239-020-0177-7>
- Ameen, A. O., Alarape, M. A., & Adewole, K. S. (2019). Students' Academic Performance and Dropout Predictions: A review. *Malaysian Journal of Computing*, 4(2), 278–303.
- Ameri, S., Fard, M. J., Chinnam, R. B., & Reddy, C. K. (2016). Survival Analysis based Framework for Early Prediction of Student Dropouts. *International Conference on Computational Intelligence and Knowledge Management (CIKM)*, 903–912. <https://doi.org/http://dx.doi.org/10.1145/2983323.2983351>
- Amra, I. A. A., & Maghari, A. Y. A. (2017). Students Performance Prediction Using KNN and Naïve Bayesian. *2017 8th International Conference on Information Technology (ICIT) Students*, 909–913. IEEE.
- Amrieh, E. A., Hamtini, T., & Aljarah, I. (2016). Mining Educational Data to Predict Student's

- Academic Performance using Ensemble Methods. *International Journal of Database Theory and Application*, 9(8), 119–136. <https://doi.org/10.14257/ijdta.2016.9.8.13>
- Andonie, R., & Kovalerchuk, B. (2004). Neural Networks for Data Mining: Constrains and Open Problems. *ESANN'2004 Proceedings - European Symposium on Artificial Neural Networks*, (April), 449–458. Retrieved from <http://www.cwu.edu/~andonie/MyPapers/ESANN2004Tutorial.pdf>
- Andrew, Y. N., and Michael, I. J. (2001). On Discriminative Vs. Generative Classifiers: A Comparison of Logistic Regression and Naïve Bayes. <https://Ai.Stanford.Edu/~ang/Papers/Nips01-Discriminativegenerative.Pdf>. Retrieved from <https://ai.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf>.
- Artelnics., F. G., & Quesada, A. (2019). Genetic Algorithms for Feature Selection. Retrieved October 24, 2019, from https://www.neuraldesigner.com/blog/genetic_algorithms_for_feature_selection
- Asif, R., Hina, S., & Haque, S. I. (2017). Predicting Student Academic Performance using Data Mining Methods. *IJCSNS International Journal of Computer Science and Network Security*, 17(5), 187–191.
- Asif, R., Merceron, A., Ali, S. A., & Haider, N. G. (2017). Analyzing Undergraduate Students' Performance Using Educational Data Mining. *Computers and Education*, 113, 177–194. <https://doi.org/10.1016/j.compedu.2017.05.007>
- Asogbon, M. G., Samuel, O. W., Omisore, M. O., & Ojokoh, B. A. (2016). A Multi-class Support Vector Machine Approach for Students Academic Performance Prediction. *International Journal of Multidisciplinary and Current Research Research*, 4(March), 210–215.
- Aulck, L., Velagapudi, N., Blumenstock, J., & West, J. (2016). Predicting Student Dropout in Higher Education. *International Conference on Machine Learning (ICML)*, 16–20. <https://doi.org/10.1002/prot.24187>
- Auria, L., & Moro, R. A. (2011). Support Vector Machines (SVM) as a Technique for Solvency Analysis. *SSRN Electronic Journal*, (August). <https://doi.org/10.2139/ssrn.1424949>
- Aye, N. (2020). A Comparative Analysis of Classification Algorithms on Students' Performance. *TRANSACTIONS ON NETWORKS AND COMMUNICATIONS*, 8(2), 20–34. <https://doi.org/10.14738/tnc.82.8267>

- Ayyaz, F., Asif, A., & Zahoor, A. (2020). Predicting the Performance of Primary School Students. *American Journal of Computer Science and Survey*, 8(1), 1–8.
<https://doi.org/10.36648/computer-science-engineering-survey.08.01.01>
- Aziz, A. A., Ismail, N. H., & Ahmad, F. (2013). Mining Students' Academic Performance. *Journal of Theoretical and Applied Information Technology*, 53(3), 485–495.
- Aziz, A. A., Ismail, N. H., Ahmad, F., & Hassan, H. (2015). A Framework for Students' Academic Performance Analysis Using Naïve Bayes Classifier. *Jurnal Teknologi*, 3, 13–19.
- Babatunde, O., Armstrong, L., Leng, J., & Diepeveen, D. (2014). A Genetic Algorithm-Based Feature Selection. *International Journal of Electronics Communication and Computer Engineering*, 5(4), 899–905.
- Baker, R. S., & Inventado, P. S. (2017). Educational Data Mining and Learning Analytics. In Richard E. West (Ed.), *Foundations of Learning and Instructional Design Technology* (pp. 544–561). Retrieved from <https://liddfoundations.pressbooks.com/chapter/sociocultural-learning/>
- Baker, R. S. J. (2011). *Data Mining for Education*. <https://doi.org/10.4018/978-1-59140-557-3>
- Baker, R. S. J. D., & Yacef, K. (2009). The State of Educational Data Mining in 2009 : A Review and Future Visions. *Journal of Educational Data Mining*, 1(1), 3–16.
<https://doi.org/http://doi.ieeecomputersociety.org/10.1109/ASE.2003.1240314>
- Baker, Ryan S., Berning, A. W., Gowda, S. M., Zhang, S., & Hawn, A. (2020). Predicting K-12 Dropout. *Journal of Education for Students Placed at Risk*, 25(1), 28–54.
<https://doi.org/10.1080/10824669.2019.1670065>
- Barros, T. M., Neto, P. A. S., Silva, I., & Guedes, L. A. (2019). Predictive Models for Imbalanced Data : A School Dropout Perspective. *Educational Sciences*, 9(275), 1–13.
<https://doi.org/10.3390/educsci9040275>
- Bendangnuksung. (2018). Students' Performance Prediction Using Deep Neural Network. *International Journal of Applied Engineering Research*, 13(02), 1171–1176.
- Bhagvatula, S., Shreyas, K. J., Gupta, S., & Singh, M. (2015). Contribution of Students' Academic Efforts on Prediction of Students' Performance Using WEKA. *International Journal of Current Engineering and Scientific Research (IJCESR)*, 2(11), 83–89. Retrieved from troindia.in/journal/ijcesr/vol2iss11/83-89.pdf
- Bonifro, F. Del, Gabbrielli, M., Lisanti, G., & Zingaro, S. P. (2020). Student Dropout Prediction.

- In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-030-52237-7_11
- Burman, I., & Som, S. (2019). Predicting Students Academic Performance Using Support Vector Machine. *Proceedings - 2019 Amity International Conference on Artificial Intelligence, AICAI 2019*, 756–759. <https://doi.org/10.1109/AICAI.2019.8701260>
- Chao, W. (2011). Machine Learning Tutorial. In *Aec-Apc* (First Edit). Retrieved from <http://disp.ee.ntu.edu.tw/~pujols/Machine Learning Tutorial.pdf>
- Chen, J. F., Hsieh, H. N., & Do, Q. H. (2014). Predicting Student Academic Performance: A Comparison of Two Meta-heuristic Algorithms Inspired by Cuckoo Birds for Training Neural Networks. *Algorithms*, 7(4), 538–553. <https://doi.org/10.3390/a7040538>
- Chen, J., Feng, J., Sun, X., Wu, N., Yang, Z., & Chen, S. (2019). MOOC Dropout Prediction Using a Hybrid Algorithm Based on Decision Tree and Extreme Learning Machine. *Mathematical Problems in Engineering*, 2019. <https://doi.org/10.1155/2019/8404653>
- Chopra, N., & Lal, M. (2012). Research Methodology for Educational Data Mining in India. *INDIACom 2012 Proceedings*. Retrieved from [www.bvicam.ac.in › news › INDIACom 2012 Proceedings › papers](http://www.bvicam.ac.in/news/INDIACom2012Proceedings/papers)
- Damuluri, S., Islam, K., Ahmadi, P., & Qureshi, N. (2020). Analyzing Navigational Data and Predicting Student Grades Using Support Vector Machine. *Emerging Science Journal*, 4(4), 243–252. <https://doi.org/10.28991/esj-2020-01227>
- Daud, A., Aljohani, N. R., Abbasi, R. A., Lytras, M. D., Abbas, F., & Alowibdi, J. S. (2017). Predicting Student Performance using Advanced Learning Analytics. *International World Wide Web Conference Committee (IW3C2)*, 415–421. Retrieved from <https://dl.acm.org/citation.cfm?id=3054164>
- Dekker, G. W. ., Pechenizkiy, M. ., & Vleeshouwers, J. M. . (2009). Predicting Students Drop Out: A Case Study. *International Conference On Educational Data Mining*, 41–50. <https://doi.org/doi:10.1037/0893-3200.21.3.344>
- Do, Q. H., & Chen, J. F. (2013). A neuro-fuzzy approach in the classification of students' academic performance. *Computational Intelligence and Neuroscience*, 2013, 1–8. <https://doi.org/10.1155/2013/179097>
- Du, K.-L., & Swamy, M. N. S. (2016). Search and Optimization by Metaheuristics: Techniques

- and Algorithms Inspired by Nature. In *Search and Optimization by Metaheuristics*.
<https://doi.org/10.1007/978-3-319-41192-7>
- Dugong, F. (2011). Strengths and Weaknesses of this algorithm are as. Retrieved October 24, 2019, from <https://www.coursehero.com/file/p6omit2/Strengths-and-weaknesses-of-this-algorithm-are-as-follows-Strength-1-Simple/>
- Eashwar, K. B., Venkatesan, R., & Ganesh, D. (2017). Student Performance Prediction Using SVM. *International Journal of Mechanical Engineering and Technology (IJMET)*, 8(11), 649–662.
- Enu, J., Agyman, O. K., & Nkum, D. (2015). Factors Influencing Students' Mathematics Performance in Some Selected Colleges of Education in Ghana. *International Journal of Education Learning and Development*, 3(3), 68–74. <https://doi.org/10.1007/978-1-4471-5520-1>
- Fareo, D. O. (2013). Truancy and Academic Performance of Secondary School Students in Southwestern Nigeria : Implications for Counselling. *International Journal for Cross-Disciplinary Subjects in Education (IJCDSE), Special Issue*, 3(2), 1424–1428.
- Farid, D. M., Zhang, L., Rahman, C. M., Hossain, M. A., & Strachan, R. (2014). Hybrid Decision Tree and Naïve Bayes Classifiers for Multi-class Classification Tasks. *Expert Systems with Applications*, 41(4 PART 2), 1937–1946.
<https://doi.org/10.1016/j.eswa.2013.08.089>
- Francis, B. K., & Babu, S. S. (2019). Predicting Academic Performance of Students Using a Hybrid Data Mining Approach. *Journal of Medical Systems*, 43(6).
<https://doi.org/10.1007/s10916-019-1295-4>
- Gilbert, N. (2017). Predicting Success : An Application of Data Mining Techniques to Student Outcomes. *International Journal of Data Mining & Knowledge Management Process*, 7(2), 01–20. <https://doi.org/10.5121/ijdkp.2017.7201>
- Gorunescu, F. (2011). *Data Mining: Concepts, Models and Techniques* (Intelligence). Verlag Berlin Heidelberg: Springer.
- Goscik, J., & Łukaszuk, T. (2013). Application of the Recursive Feature Elimination and the Relaxed Linear Separability Feature Selection Algorithms To Gene Expression Data Analysis. *Advances In Computer Science Research Application*, 10, 39–52. Retrieved from <file:///Users/Littleyi/Downloads/Goscik.pdf>

- Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3, 1157–1182. Retrieved from <http://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf>
- Hamsa, H., Indiradevi, S., & Kizhakkethottam, J. J. (2016). Student Academic Performance Prediction Model Using Decision Tree and Fuzzy Genetic Algorithm. *Procedia Technology*, 25, 326–332. <https://doi.org/10.1016/j.protcy.2016.08.114>
- Han, J., Kamber, M., & Pei, J. (2012). *Data Mining Concepts and Techniques* (Third). USA: Morgan Kaufmann Publishers.
- Han, J., & Kamber, M. (2006). Data Mining Concepts and Techniques. In *Data Mining: Concepts and Techniques* (Second). <https://doi.org/10.1007/978-3-642-19721-5>
- Hasan, R., Palaniappan, S., Mahmood, S., Abbas, A., Sarker, K. U., & Sattar, M. U. (2020). Predicting Student Performance in Higher Educational Institutions Using Video Learning Analytics and Data Mining Techniques. *Applied Sciences (Switzerland)*, 10(11), 1–20. <https://doi.org/10.3390/app10113894>
- Hasheminejad, S. M., & Sarvmili, M. (2019). S3PSO: Students' Performance Prediction Based on Particle Swarm Optimization. *Journal of AI and Data Mining*, 0(0), 77–96. <https://doi.org/10.22044/jadm.2018.5506.1662>
- Hijazi, S. T., & Raza Naqvi, S. M. M. (2006). Factors Affecting Students' Performance. *Bangladesh E-Journal of Sociology*, 3(1), 1–10.
- Homsapaya, K., & Sornil, O. (2017). Improving Floating Search Feature Selection Using Genetic Algorithm. *Journal of ICT Research and Applications*, 11(3), 299–317. <https://doi.org/10.5614/itbj.ict.res.appl.2017.11.3.6>
- Joseph, S., & Devadas, L. (2015). Student's Performance Prediction Using Weighted Modified ID3 Algorithm. *International Journal of Scientific Research Engineering & Technology (IJSRET)*, 4(5), 571–576.
- Kabakchieva, D. (2012). Student Performance Prediction by Using Data Mining Classification Algorithms. *International Journal of Computer Science and Management Research*, 1(4), 686–690.
- Kannan, V. (2016). *Feature Selection using Genetic Algorithm* (San José State University). <https://doi.org/10.1145/2908961.2931636>
- Kantardizc, M. (2011). *Concepts, Models, Methods, and Algorithms* (second; R. Abhari, J. A. G.

- W. A. F. Canavero, M. E.- Hawary, B.-M. H. M. L. D. Jacobson, & O. P. M. S. N. T. S. G. Zobrist, Eds.). New Jersey: John Wiley & Sons Inc.
- Kaur, G., & Singh, W. (2016). Prediction Of Student Performance Using Weka Tool. *International Journal of Engineering Sciences*, 17(January), 2229–6913.
- Kaur, P., Singh, M., & Josan, G. S. (2015). Classification and Prediction Based Data Mining Algorithms to Predict Slow Learners in Education Sector. *Procedia Computer Science*, 57, 500–508. <https://doi.org/10.1016/j.procs.2015.07.372>
- Kaushik, S. (2016). Introduction to Feature Selection methods with an example (or how to select the right variables?). Retrieved October 24, 2019, from www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables
- Khamis, A., Azmeeza, C., Hamat, C. A., Affendi Abdullah, A. M., Saharan, S., & Mohd Asrah, N. (2017). Survival Analysis for Undergraduate Student's Performance. *International Journal of Advanced Research in Science, Engineering and Technology*, 4(10), 4649–4654.
- Khare, P., & Burse, K. (2016). Feature Selection Using Genetic Algorithm and Classification using Weka for Ovarian Cancer. *International Journal of Computer Science and Information Technologies (IJCSIT)*, 7(1), 194–196.
- Kim, B. H., Vizitei, E., & Ganapathi, V. (2018). GritNet: Student Performance Prediction with Deep Learning. *Proceedings of the 11th International Conference on Educational Data Mining, EDM 2018*. Retrieved from educationaldatamining.org/files/conferences/EDM2018/.../EDM2018_paper_257.pdf
- LAK. (2011). Calls for Papers. *1st International Conference on Learning Analytics and Knowledge*. Banff, Alberta.
- Lakkaraju, H., Aguiar, E., Shan, C., Miller, D., Bhanpuri, N., Ghani, R., & Addison, K. L. (2015). A Machine Learning Framework to Identify Students at Risk of Adverse Academic Outcomes. *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15*, 1909–1918. <https://doi.org/10.1145/2783258.2788620>
- Li, X., Zhu, X., Zhu, X., Ji, Y., & Tang, X. (2020). *Student Academic Performance Prediction Using Deep Multi-source Behavior Sequential Network Xiang* (Vol. 1). <https://doi.org/10.1007/978-3-030-47426-3>

- Lu, O., Huang, A., Huang, J., Lin, A., Ogata, H., & Yang, S. (2018). Applying Learning Analytics for the Early Prediction of Students' Academic Performance in Blended Learning. *International Forum of Educational Technology & Society*, 21(2), 220–232. Retrieved from https://www.jstor.org/stable/26388400?seq=1&cid=pdf-reference#references_tab_contents
- Makhtar, M., Nawang, H., Nor, S., & Shamsuddin, W. A. N. (2017). Analysis on Students' Performance Using Naive Bayes. *Journal of Theoretical and Applied Information Technology*, 95(16), 3993–4000.
- Márquez-Vera, C., Cano, A., Romero, C., Noaman, A. Y. M., Mousa Fardoun, H., & Ventura, S. (2016). Early Dropout Prediction using Data Mining: A Case Study with High School Students. *Expert Systems*, 33(1). <https://doi.org/10.1111/exsy.12135>
- Mishra, T., Kumar, D., & Gupta, S. (2014). Mining Students' Data for Prediction Performance. *2014 Fourth International Conference on Advanced Computing & Communication Technologies*, 255–262. <https://doi.org/10.1109/ACCT.2014.105>
- Mohsin, F., Norwawi, M. N., M., Hibadullah, C. F., & Wahab, M. H. A. (2010). Mining the Student Programming Performance Using Rough Set. *Proceedings of 2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering, ISKE 2010*, 478–483. <https://doi.org/10.1109/ISKE.2010.5680824>
- Mubarak, A. A., Cao, H., & Zhang, W. (2020). Prediction of Students' Early Dropout Based on their Interaction Logs in Online Learning Environment. *Interactive Learning Environments*, 0(0), 1–20. <https://doi.org/10.1080/10494820.2020.1727529>
- Mueen, A., Zafar, B., & Manzoor, U. (2016). Modeling and Predicting Students' Academic Performance Using Data Mining Techniques. *International Journal of Modern Education and Computer Science*, 8(11), 36–42. <https://doi.org/10.5815/ijmeecs.2016.11.05>
- Mushtaq, I., & Khan, S. N. (2012). Factors Affecting Students' Academic Performance. *Global Journal of Management and Business Research*, 12(9), 16–22.
- Naicker, N., Adeliyi, T., & Wing, J. (2020). Linear Support Vector Machines for Prediction of Student Performance in School-Based Education. *Mathematical Problems in Engineering*, 2020, 1–7. <https://doi.org/10.1155/2020/4761468>
- Nayebzadeh, S., Dehnavi, H. D., Nejad, M. S., Mir, M., & Sadrabadi, M. (2013). Factors Affecting the Academic Improvement of Accounting Students in Islamic Azad University of Yazd. *International Journal of Academic Research in Economics and Management*

- Sciences*, 2(4), 231–243. <https://doi.org/10.6007/IJAREMS/v2-i4/122>
- Nivre, J. (2003). Discriminative Models. *Stp.Lingfil.Uu.Se/~nivre/Master/Discriminative.Pdf*, 1–5. Retrieved from stp.lingfil.uu.se/~nivre/master/discriminative.pdf
- Ogor, E. N. (2007). Student Academic Performance Monitoring and Evaluation Using Data Mining Techniques. *Fourth Congress of Electronics, Robotics, and Automotive Mechanics*, 354–359. <https://doi.org/10.1109/CERMA.2007.78>
- Ogunde, A. O., & Ajibade, D. A. (2014). A Data Mining System for Predicting University Students' Graduation Grades Using ID3 Decision Tree Algorithm. *Computer Science and Information Technology*, 2(1), 21–46.
- Oladipupo, O., Daramola, O., Oyelade, J., & Afolabi, I. (2017). Assessing the Impact of Class Attendance on Student's Academic Performance Using Data Mining. *Asian Journal of Information Technology*, 16(2–5), 218–226.
- Oloruntoba, S. A., & Akinode, J. L. (2017). Student Academic Performance Prediction Using Support Vector Machine. *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY STUDENT*, 6(12), 588–598.
- Osofisan, A., Adeyemo, O., & Oluwasusi, S. (2014). Empirical Study of Decision Tree and Artificial Neural Network Algorithm for Mining Educational Database. *African Journal of Computing & ICT* ©, 7(2), 187–196. Retrieved from http://www.ajocict.net/uploads/V7N2P22-2014_AJOCICT_-_Paper_22.pdf
- Pal, S. (2012). Mining Educational Data Using Classification to Decrease Dropout Rate of Students. *International Journal of Multidisciplinary Sciences and Engineering*, 3(5), 35–39. Retrieved from www.ijmse.org
- Pandey, U. K., & Pal, P. S. (2011). *Data Mining : A Prediction of Performer or Underperformer Using Classification*. 2(2), 686–690.
- Patacsil, F. F. (2020). Survival Analysis Approach for Early Prediction of Student Dropout using Enrollment Student Data and Ensemble Models. *Universal Journal of Educational Research*, 8(9), 4036–4047. <https://doi.org/10.13189/ujer.2020.080929>
- Piątkowska, W., & Martyna, J. (2009). A Hybrid Classifier Based on SVM Method for Cancer Classification. *Studia Informatica*, 30(2), 229–307.
- Rajeshinigo, D., & Jebamalar, J. P. A. (2017). Educational Mining: A Comparative Study of Classification Algorithms Using WEKA. *International Journal of Innovative Research in*

- Computer and Communication Engineering*, 5(3), 5583–5589.
<https://doi.org/10.15680/IJIRCCE.2017>.
- Ray, S. (2017). Understanding Support Vector Machine Algorithm from examples (along with code). Retrieved September 22, 2018, from
<https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>
- Razaque, F., Soomro, N., Shaikh, S. A., Soomro, S., Samo, J. A., Kumar, N., & Dharejo, H. (2018). Using Naïve Bayes Algorithm to Students' Bachelor Academic Performances Analysis. *4th IEEE International Conference on Engineering Technologies and Applied Sciences, ICETAS 2017, 2018-Janua*(November), 1–5.
<https://doi.org/10.1109/ICETAS.2017.8277884>
- Respondek, L., Seufert, T., Stupnisky, R., & Nett, U. E. (2017). Perceived Academic Control and Academic Emotions Predict Undergraduate University Student Success: Examining Effects on Dropout Intention and Achievement. *Frontiers in Psychology*, 8(MAR), 1–18.
<https://doi.org/10.3389/fpsyg.2017.00243>
- Romero, C., & Ventura, S. (2007). Educational Data Mining: A Survey from 1995 to 2005. *Expert Systems with Applications*, 33(1), 135–146.
<https://doi.org/10.1016/j.eswa.2006.04.005>
- Romero, C., & Ventura, S. (2010). Educational data mining: A Review of the State of the Art. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 40(6), 601–618. <https://doi.org/10.1109/TSMCC.2010.2053532>
- Romero, Cristóbal, Ventura, S., Espejo, P. G., & Hervás, C. (2008). Data Mining Algorithms to Classify Students. *1st International Conference on Educational Data Mining*, 8–17.
Retrieved from
<https://pdfs.semanticscholar.org/c6e2/9d4a3d70cd09c1397e736a38afd25def7bd3.pdf>
- Rovira, S., Puertas, E., & Igual, L. (2017). Data-driven System to Predict Academic Grades and Dropout. *PLoS ONE*, 12(2), 1–21. <https://doi.org/10.1371/journal.pone.0171207>
- Ruby, J., & David, K. (2015). Analysis of Influencing Factors in Predicting Students Performance Using MLP -A Comparative Study. *International Journal of Innovative Research in Computer and Communication Engineering*, 3297(2), 1085–1092.
<https://doi.org/10.15680/ijircce.2015.0302070>

- Saa, A. A. (2016). Educational Data Mining & Students' Performance Prediction. *International Journal of Advanced Computer Science and Applications*, 7(5), 212–220.
<https://doi.org/10.14569/IJACSA.2016.070531>
- Sagale, A. D., & Kale, S. G. (2014). Hybrid Model For Intrusion Detection Using Naive Bayesian and Support Vector Machine. *International Journal of Computing and Technology*, 1(3), 56–59.
- Scikit-learn. (2019). sklearn.feature_selection.RFE. Retrieved October 24, 2019, from https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html
- Seifert, K., & Sutton, R. (2009). *Educational Psychology Second Edition* (Second; J. S. Kelvin Seifert, Marisa Drexel, Ed.). [https://doi.org/10.1016/S0738-3991\(02\)00243-4](https://doi.org/10.1016/S0738-3991(02)00243-4)
- Shakya, S., & Sigdel, S. (2017). An Approach to Develop a Hybrid Algorithm Based on Support Vector Machine and Naive Bayes for Anomaly Detection. *Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2017, 2017-Janua*(1), 323–327. <https://doi.org/10.1109/CCAA.2017.8229836>
- Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding Machine Learning*. <https://doi.org/10.1017/CBO9781107298019>
- Simeunović, V., & Preradović, L. (2014). Using Data Mining to Predict Success in Studying. *Croatian Journal of Education*, 16(December 2012), 491–523.
- Solomon, D., Patil, P. seem., & Agrawal, P. proov. (2018). Predicting Performance and Potential Difficulties of University Student using Classification : Survey Paper. *International Journal of Pure and Applied Mathematics*, 118(18), 2703–2707.
- Strecht, P., Cruz, L., Soares, C., Mendes-Moreira, J., & Abreu, R. (2015). A Comparative Study of Classification and Regression Algorithms for Modelling Students' Academic Performance. *Proceedings of the 8th International Conference on Educational Data Mining*, 392–395. <https://doi.org/10.1038/417671a>
- Suhirman, X. X., Zain, J. M., & Herawan, T. (2014). Data Mining for Education Decision Support: A Review. *International Journal of Emerging Technologies in Learning*, 9(6), 4–19. <https://doi.org/10.3991/ijet.v9i6.3950>
- Sultana, S., Khan, S., & Abbas, M. A. (2017). Predicting Performance of Electrical Engineering Students Using Cognitive and Non-Cognitive Features for Identification of Potential Dropouts. *International Journal of Electrical Engineering Education*, 54(2), 105–118.

<https://doi.org/10.1177/0020720916688484>

- Tang, J., Alelyani, S., & Liu, H. (2014). Feature Selection for Classification: A Review. In C. C. Aggarwal (Ed.), *Data Classification: Algorithms and Applications* (pp. 37–64). Retrieved from <https://www.pdfdrive.com/data-classification-algorithms-and-applications-d157702665.html>
- Tekin, A. (2014). Early Prediction of Students' Grade Point Averages at Graduation: A Data Mining Approach. *Eurasian Journal of Educational Research*, 14(54), 207–226.
<https://doi.org/10.14689/ejer.2014.54.12>
- Tinto, V. (1975). *Dropout from Higher Education : A Theoretical Synthesis of Recent Research*. (1).
- Tinto, V. (2006). Research and Practice of Student Retention: What Next? *Journal of College Student Retention: Research, Theory and Practice*, 8(1), 1–19.
<https://doi.org/10.2190/C0C4-EFT9-EG7W-PWP4>
- Tran, T.-O., Dang, H.-T., Dinh, V.-T., Truong, T.-M.-N., Vuong, T.-P.-T., & Phan, X.-H. (2017). Performance Prediction for Students: A Multi-Strategy Approach. *Cybernetics and Information Technologies*, 17(2). <https://doi.org/10.1515/cait-2017-0024>
- Udomsak, N. (2015). *How do the naive Bayes classifier and the Support Vector Machine compare in their ability to forecast the Stock Exchange of Thailand?* Retrieved from <http://arxiv.org/abs/1511.08987>
- Upadhyay, N. & Katiyar, V. (2014). A Survey on the Classification Techniques In Educational Data Mining. *International Journal of Computer Applications Technology and Research*, 3(11), 725–728.
- Vamshidharreddy, V. S., Saketh, A. S. & Gnanajeyaraman, R. (2020). Student's Academic Performance Prediction Using Machine Learning Approach. *International Journal of Advanced Science and Technology*, 29(9), 6731–6737.
- Verma, K., Singh, A., & Verma, P. (2016). A Review on Predicting Student Performance Using Data Mining Method. *International Journal of Current Engineering and Scientific Research (IJCESR)*, 3(1), 127–132.
- Witten, I. H. (2017). *Data Mining with WEKA - Online Course*. Retrieved from <https://www.futurelearn.com/courses/data-mining-with-weka>
- Witten, I. H., & Frank, E. (2005). *Data Mining Practical Machine Learning Tools and*

- Techniques* (Second). <https://doi.org/0120884070>, 9780120884070
- Yadav, S. K., Bharadwaj, B., & Pal, S. (2011). Data Mining Applications: A Comparative Study for Predicting Student's performance. *International Journal of Innovative Technology & Creative Engineering*, 1(12), 13–19.
- Yadav, S. K., & Pal, S. (2012). Data Mining : A Prediction for Performance Improvement of Engineering Students using Classification. *World of Computer Science and Information Technology Journal (WCSIT)*, 2(2), 51–56.
- Yang, S. J. H., Lu, O. H., Huang, A. Y. Q., Huang, J. C. H., Ogata, H., & Lin, A. J. Q. (2018). Predicting Students ' Academic Performance Using Multiple Linear Regression and Principal. *Journal of Information Processing*, 26, 170–176.
<https://doi.org/10.2197/ipsjjip.26.170>
- Yassein, N. A. M., Helali, R. G., & Mohomad, S. B. (2017). Predicting Student Academic Performance in KSA using Data Mining Techniques. *Journal of Information Technology & Software Engineering*, 07(05), 370. <https://doi.org/10.4172/2165-7866.1000213>
- Yıldız, O., Bal, A., Gulseçen, S., & Kentli, F. D. (2012). A Genetic-Fuzzy Based Mathematical Model to Evaluate The Distance Education Students' Academic Performance. *Procedia - Social and Behavioral Sciences*, 55, 409–418. <https://doi.org/10.1016/j.sbspro.2012.09.519>
- Zhou, S. K., Greenspan, H., & Shen, D. (2017). Deep Learning for Medical Image Analysis. In *Deep Learning for Medical Image Analysis* (First Edit).
https://doi.org/10.4103/jpi.jpi_27_18

APPENDIX

Sample Codes

i) Using UniDataSAP_num dataset

```
library(caret)
library(dplyr)
library(partykit)
library(mlbench)
library(readxl)
library(e1071)
library(klaR)

set.seed(101)

preProcessData<-function(d){

  DF <- as.data.frame(unclass(d)) #convert all chr columns to factor

  return (DF)

}

#default parameters for modeling using PERCENTAGE SPLIT ####
#####

train_svm_model_linear<-function(d,t){

  svm_time<-system.time(svm_model <-svm(Class ~ ., data = d, kernel = "linear", scale =
FALSE))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}
```

```

train_svm_model_poly<-function(d,t){
  svm_time<-system.time(svm_model <-svm(Class ~ ., data = d, kernel = "polynomial", scale =
FALSE))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return
(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))
}

train_svm_model_radial<-function(d,t){
  svm_time<-system.time(svm_model <-svm(Class ~ ., data = d, kernel = "radial", scale =
FALSE))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return
(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))
}

train_svm_model_sigmoid<-function(d,t){
  svm_time<-system.time(svm_model <-svm(Class ~ ., data = d, kernel = "sigmoid", scale =
FALSE))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return

```

```

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))
}

##### END OF PERCENTAGE SPLIT #####

#default parameters for modeling using 10 FOLD-CROSS VALIDATION ####
#####

train_svm_model_linear_fold<-function(d,t){

  svm_time<-system.time(svm_model<-train(Class ~ ., data = d, method =
"svmLinear",trControl=trainControl(method='cv',number=10)))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))
}

train_svm_model_poly_fold<-function(d,t){

  svm_time<-system.time(svm_model<-train(Class ~ ., data = d, method =
"svmPoly",trControl=trainControl(method='cv',number=10)))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))
}

train_svm_model_radial_fold<-function(d,t){

```

```

svm_time<-system.time(svm_model<-train(Class ~ ., data = d, method =
"svmRadial",trControl=trainControl(method='cv',number=10)))

pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

performance<-confusionMatrix(pred_svm,t$Class)

return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}

train_svm_model_sigmoid_fold<-function(d,t){

svm_time<-system.time(svm_model<-train(Class ~ ., data = d, method =
"svmSigmoid",trControl=trainControl(method='cv',number=10)))

pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

performance<-confusionMatrix(pred_svm,t$Class)

return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}

##### END OF 10 FOLD CROSS-VALIDATION #####

#Naive Bayes mod

train_nb_model<-function(d,t){

nb_time<-system.time(nb_model<-train(d[,1:ncol(d)-
1],d[,ncol(d)],'nb',trControl=trainControl(method='cv',number=10)))

pred_nb<-predict(nb_model,t[,1:ncol(t)-1])

performance<-confusionMatrix(pred_nb,t$Class)

return (list("time"=nb_time,"model"=nb_model,"pred"=pred_nb,"performance"=performance))

```

```

}

#setwd("~/EducationDM")

data<-read_excel("C:\\Users\\Abu Anas\\Documents\\EducationDM\\UniDataSAP_num.xlsx")

#data$Mat.No=NULL

#rm(list=ls())

#data$MatricNO=NULL remove matric number

#na.omit(data_grade)

data<-preProcessData(data)

data<-na.omit(data)

index<-createDataPartition(data$Class,p=0.7,list=FALSE)

train_data<-data[index,]

test_data<-data[-index,]

p_svm_linear<-train_svm_model_linear_fold(train_data,test_data)
p_svm_poly<-train_svm_model_poly_fold(train_data,test_data)
p_svm_radial<-train_svm_model_radial_fold(train_data,test_data)
p_svm_sigmoid<-train_svm_model_sigmoid(train_data,test_data)
p_nb_model<-train_nb_model(train_data,test_data)

```

(ii) Using PolyDataSAP_num dataset

```

library(caret)
library(dplyr)
library(partykit)
library(mlbench)
library(readxl)
library(e1071)
library(klaR)

set.seed(101)

preProcessData<-function(d){

  DF <- as.data.frame(unclass(d)) #convert all chr columns to factor

```

```

    return (DF)
}

#default parameters for modeling using PERCENTAGE SPLIT ####
#####

train_svm_model_linear<-function(d,t){

  svm_time<-system.time(svm_model <-svm(Class ~ ., data = d, kernel = "linear", scale =
FALSE))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}

train_svm_model_poly<-function(d,t){

  svm_time<-system.time(svm_model <-svm(Class ~ ., data = d, kernel = "polynomial", scale =
FALSE))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}

train_svm_model_radial<-function(d,t){

```

```

svm_time<-system.time(svm_model <-svm(Class ~ ., data = d, kernel = "radial", scale =
FALSE))

pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

performance<-confusionMatrix(pred_svm,t$Class)

return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}

train_svm_model_sigmoid<-function(d,t){

svm_time<-system.time(svm_model <-svm(Class ~ ., data = d, kernel = "sigmoid", scale =
FALSE))

pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

performance<-confusionMatrix(pred_svm,t$Class)

return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}

##### END OF PERCENTAGE SPLIT #####

#default parameters for modeling using 10 FOLD-CROSS VALIDATION ####

#####

train_svm_model_linear_fold<-function(d,t){

svm_time<-system.time(svm_model<-train(Class ~ ., data = d, method =

"svmLinear",trControl=trainControl(method='cv',number=10)))

pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

```

```

performance<-confusionMatrix(pred_svm,t$Class)

return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}

train_svm_model_poly_fold<-function(d,t){

  svm_time<-system.time(svm_model<-train(Class ~ ., data = d, method =
"svmPoly",trControl=trainControl(method='cv',number=10)))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}

train_svm_model_radial_fold<-function(d,t){

  svm_time<-system.time(svm_model<-train(Class ~ ., data = d, method =
"svmRadial",trControl=trainControl(method='cv',number=10)))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}

train_svm_model_sigmoid_fold<-function(d,t){

  svm_time<-system.time(svm_model<-train(Class ~ ., data = d, method =
"svmSigmoid",trControl=trainControl(method='cv',number=10)))

```

```

pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

performance<-confusionMatrix(pred_svm,t$Class)

return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}

##### END OF 10 FOLD CROSS-VALIDATION #####

#Naive Bayes mod

train_nb_model<-function(d,t){

  nb_time<-system.time(nb_model<-train(d[,1:ncol(d)-

1],d[,ncol(d)],'nb',trControl=trainControl(method='cv',number=10)))

  pred_nb<-predict(nb_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_nb,t$Class)

  return (list("time"=nb_time,"model"=nb_model,"pred"=pred_nb,"performance"=performance))

}

#setwd("~/EducationDM")

data<-read_excel("C:\\Users\\Abu Anas\\Documents\\EducationDM\\PolyDataSAP_num.xlsx")

#data$Mat.No=NULL

#rm(list=ls())

#data$MatricNO=NULL remove matric number

#na.omit(data_grade)

data<-preProcessData(data)

data<-na.omit(data)

index<-createDataPartition(data$Class,p=0.7,list=FALSE)

```

```

train_data<-data[index,]

test_data<-data[-index,]

p_svm_linear<-train_svm_model_linear_fold(train_data,test_data)
p_svm_poly<-train_svm_model_poly_fold(train_data,test_data)
p_svm_radial<-train_svm_model_radial_fold(train_data,test_data)
p_svm_sigmoid<-train_svm_model_sigmoid(train_data,test_data)
p_nb_model<-train_nb_model(train_data,test_data)

```

(iii) Using UniDataDropout_num dataset

```

library(caret)
library(dplyr)
library(partykit)
library(mlbench)
library(readxl)
library(e1071)
library(klaR)

set.seed(101)

preProcessData<-function(d){

  DF <- as.data.frame(unclass(d)) #convert all chr columns to factor

  return (DF)

}

#default parameters for modeling using PERCENTAGE SPLIT ####
#####

train_svm_model_linear<-function(d,t){

  svm_time<-system.time(svm_model <-svm(Class ~ ., data = d, kernel = "linear", scale =
FALSE))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

```

```

return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}

train_svm_model_poly<-function(d,t){

  svm_time<-system.time(svm_model <-svm(Class ~ ., data = d, kernel = "polynomial", scale =
FALSE))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}

train_svm_model_radial<-function(d,t){

  svm_time<-system.time(svm_model <-svm(Class ~ ., data = d, kernel = "radial", scale =
FALSE))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}

train_svm_model_sigmoid<-function(d,t){

  svm_time<-system.time(svm_model <-svm(Class ~ ., data = d, kernel = "sigmoid", scale =
FALSE))

```

```

pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

performance<-confusionMatrix(pred_svm,t$Class)

return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}

##### END OF PERCENTAGE SPLIT #####

#default parameters for modeling using 10 FOLD-CROSS VALIDATION ####

#####

train_svm_model_linear_fold<-function(d,t){

  svm_time<-system.time(svm_model<-train(Class ~ ., data = d, method =

"svmLinear",trControl=trainControl(method='cv',number=10)))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}

train_svm_model_poly_fold<-function(d,t){

  svm_time<-system.time(svm_model<-train(Class ~ ., data = d, method =

"svmPoly",trControl=trainControl(method='cv',number=10)))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

```

```

    return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))
}

train_svm_model_radial_fold<-function(d,t){

  svm_time<-system.time(svm_model<-train(Class ~ ., data = d, method =

"svmRadial",trControl=trainControl(method='cv',number=10)))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))
}

train_svm_model_sigmoid_fold<-function(d,t){

  svm_time<-system.time(svm_model<-train(Class ~ ., data = d, method =

"svmSigmoid",trControl=trainControl(method='cv',number=10)))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))
}

##### END OF 10 FOLD CROSS-VALIDATION #####

#Naive Bayes mod

train_nb_model<-function(d,t){

```

```

nb_time<-system.time(nb_model<-train(d[,1:ncol(d)-
1],d[,ncol(d)],'nb',trControl=trainControl(method='cv',number=10)))

pred_nb<-predict(nb_model,t[,1:ncol(t)-1])

performance<-confusionMatrix(pred_nb,t$class)

return (list("time"=nb_time,"model"=nb_model,"pred"=pred_nb,"performance"=performance))
}

#setwd("~/EducationDM")

data<-

read_excel("C:\\Users\\AbuAnas\\Documents\\EducationDM\\UniDataDropout_num.xlsx")

#data$Mat.No=NULL

#rm(list=ls())

#data$MatricNO=NULL remove matrix number

#na.omit(data_grade)

data<-preProcessData(data)

data<-na.omit(data)

index<-createDataPartition(data$class,p=0.7,list=FALSE)

train_data<-data[index,]

test_data<-data[-index,]

p_svm_linear<-train_svm_model_linear_fold(train_data,test_data)
p_svm_poly<-train_svm_model_poly_fold(train_data,test_data)
p_svm_radial<-train_svm_model_radial_fold(train_data,test_data)
p_svm_sigmoid<-train_svm_model_sigmoid(train_data,test_data)
p_nb_model<-train_nb_model(train_data,test_data)

```

(iv) Using PolyDataDropout_num dataset

```
library(caret)
library(dplyr)
jklibrary(partykit)
library(mlbench)
library(readxl)
library(e1071)
library(klaR)

set.seed(101)

preProcessData<-function(d){

  DF <- as.data.frame(unclass(d)) #convert all chr columns to factor

  return (DF)

}

#default parameters for modeling using PERCENTAGE SPLIT #####
#####

train_svm_model_linear<-function(d,t){

  svm_time<-system.time(svm_model <-svm(Class ~ ., data = d, kernel = "linear", scale =
FALSE))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}

train_svm_model_poly<-function(d,t){

  svm_time<-system.time(svm_model <-svm(Class ~ ., data = d, kernel = "polynomial", scale =
FALSE))
```

```

pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

performance<-confusionMatrix(pred_svm,t$Class)

return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))
}

train_svm_model_radial<-function(d,t){

  svm_time<-system.time(svm_model <-svm(Class ~ ., data = d, kernel = "radial", scale =
FALSE))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))
}

train_svm_model_sigmoid<-function(d,t){

  svm_time<-system.time(svm_model <-svm(Class ~ ., data = d, kernel = "sigmoid", scale =
FALSE))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))
}

##### END OF PERCENTAGE SPLIT #####

#default parameters for modeling using 10 FOLD-CROSS VALIDATION ####

```

```
#####

train_svm_model_linear_fold<-function(d,t){

  svm_time<-system.time(svm_model<-train(Class ~ ., data = d, method =
"svmLinear",trControl=trainControl(method='cv',number=10)))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}

train_svm_model_poly_fold<-function(d,t){

  svm_time<-system.time(svm_model<-train(Class ~ ., data = d, method =
"svmPoly",trControl=trainControl(method='cv',number=10)))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}

train_svm_model_radial_fold<-function(d,t){

  svm_time<-system.time(svm_model<-train(Class ~ ., data = d, method =
"svmRadial",trControl=trainControl(method='cv',number=10)))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

}
```

```

return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}

train_svm_model_sigmoid_fold<-function(d,t){

  svm_time<-system.time(svm_model<-train(Class ~ ., data = d, method =

"svmSigmoid",trControl=trainControl(method='cv',number=10)))

  pred_svm<-predict(svm_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_svm,t$Class)

  return

(list("time"=svm_time,"model"=svm_model,"pred"=pred_svm,"performance"=performance))

}

##### END OF 10 FOLD CROSS-VALIDATION #####

#Naive Bayes mod

train_nb_model<-function(d,t){

  nb_time<-system.time(nb_model<-train(d[,1:ncol(d)-

1],d[,ncol(d)],'nb',trControl=trainControl(method='cv',number=10)))

  pred_nb<-predict(nb_model,t[,1:ncol(t)-1])

  performance<-confusionMatrix(pred_nb,t$Class)

  return (list("time"=nb_time,"model"=nb_model,"pred"=pred_nb,"performance"=performance))

}

#setwd("~/EducationDM")

data<-

read_excel("C:\\Users\\AbuAnas\\Documents\\EducationDM\\PolyDataDropout_num.xlsx")

```

```

#data$Mat.No=NULL

#rm(list=ls())

#data$MatricNO=NULL remove matric number

#na.omit(data_grade)

data<-preProcessData(data)

data<-na.omit(data)

index<-createDataPartition(data$Class,p=0.7,list=FALSE)

train_data<-data[index,]

test_data<-data[-index,]

p_svm_linear<-train_svm_model_linear_fold(train_data,test_data)
p_svm_poly<-train_svm_model_poly_fold(train_data,test_data)
p_svm_radial<-train_svm_model_radial_fold(train_data,test_data)
p_svm_sigmoid<-train_svm_model_sigmoid(train_data,test_data)
p_nb_model<-train_nb_model(train_data,test_data)


library(caret)
library(dplyr)kj
library(partykit)
library(mlbench) #for the data
library(readxl)
library(e1071)
library(klaR)
library(kernlab)
#library(pathClass)

setwd("C:\\Users\\Keystone-Dell\\Documents\\EducationDM")

set.seed(101)

preProcessData<-function(d){

  DF <- as.data.frame(unclass(d)) #convert all chr columns to factor

  return (DF)

}

```

```

#data<-read_excel("PolyDataSAP_num.xlsx")

data<-read_excel("UniDataSAP_num.xlsx")

data<-preProcessData(data)

data<-na.omit(data)

k=dim(data)

n=k[2]-1 #number of columns minus class attribute

control <- rfeControl(functions=rfFuncs, method="cv", number=10) #rfe control

# run the Recursive Feature Elimination algorithm for Wrapper method

rfe_time<-system.time(features_rfe <- rfe(Class ~ ., data = data,

                                     sizes = c(1:n),

                                     rfeControl=control, method = "svmLinear"))

#rfe_time<-system.time(features_rfe_caret <- rfe(Class ~ ., data = data,sizes = c(1:n),

#                                     rfeControl = rfeControl(functions = caretFuncs,number = 10),

#                                     ## pass options to train()

#                                     method = "svmRadial"))

# summarize the results

print(features_rfe)

#get results

features_rfe$results

# list the chosen features

predictors(features_rfe)

# plot the results

plot(features_rfe, type=c("g", "o"))

```

```

#write.csv(data_rfe_without_na,"data_rfe_without_na.csv")

#plot(features_rfe, toFile=FALSE)

library(caret)
library(dplyr)
library(partykit)
library(mlbench) #for the data
library(readxl)
library(e1071)
library(klaR)

#data(Sonar)

#data<-Sonar

setwd("C:\\Users\\Keystone-Dell\\Documents\\EducationDM")

set.seed(101)

preProcessData<-function(d){

  DF <- as.data.frame(unclass(d)) #convert all chr columns to factor

  return (DF)

}

#data<-read_excel("PolyDataDropout.xlsx")

data<-read_excel("UniDataDropout.xlsx")

data<-preProcessData(data)

data<-na.omit(data)

k=dim(data)

n=k[2]-1 #number of columns minus class attribute

index<-createDataPartition(data$Class,p=0.7,list=FALSE)

train_data<-data[index,]

test_data<-data[-index,]

```

```

rctrl1 <- rfeControl(method = "cv",
                    number = 10,
                    returnResamp = "all",
                    functions = caretFuncs,
                    saveDetails = TRUE)

#model is based on the selected features by rfe
model <- rfe(Class ~ ., data = data,
            sizes = c(1:n), #sizes = c(1, 5, 10, 15)
            method = "nb",
            trControl = trainControl(method = "cv",
                                     classProbs = TRUE),
            #tuneGrid = data.frame(k = 1:10),
            rfeControl = rctrl1)

```