DETECTION OF STUDENTS AT RISK OF ATTRITION USING DATA MINING APPROACH

BY

AMINU ONIMISI ABDULSALAMI

DEPARTMENT OF MATHEMATICS

FACULTY OF SCIENCE

AHMADU BELLO UNIVERSITY, ZARIA

NIGERIA

AUGUST, 2016

DETECTION OF STUDENTS AT RISK OF ATTRITION USING DATA MINING APPROACH

BY

Aminu Onimisi ABDULSALAMI

B.Sc. Computer Science (A.B.U, Zaria)

MSC./SCI/23347/2012-2013

A THESIS SUBMITTED TO THE SCHOOL OF POSTGRADUATE STUDIES

AHMADU BELLO UNIVERSITY (ABU), ZARIA

IN PARTIAL FULLFILMENT OF THE REQUIREMENTS FOR THE AWARD OF A

MASTER OF SCIENCE (M.Sc.) DEGREE IN COMPUTER SCIENCE

DEPARTMENT OF MATHEMATICS

AHMADU BELLO UNIVERSITY (ABU), ZARIA

AUGUST, 2016

DECLARATION

I declare that the work in this dissertation titled "DETECTION OF STUDENTS AT RISK OF ATTRITION USING A DATA MINING APPROACH" has been written by me in the Department of Mathematics under the supervision of Prof. S.B. Junaidu and Dr. A.A. Obiniyi. The information contained in the literature has been duly acknowledged in the text and a list of references provided. No part of this dissertation was previously presented for another degree or diploma at any university.

…… …… ……

Aminu Onimisi Abdulsalami Signature Date

CERTIFICATION

This dissertation titled "DETECTION OF STUDENTS AT RISK OF ATTRITION USING A DATA MINING APPROACH" by Aminu Onimisi ABDULSALAMI (M.Sc./Sci./23347/2012-2013) meets the regulations governing the award of the degree of M.Sc. Computer Science of Ahmadu Bello University (ABU), Zaria, and it is approved for its contribution to knowledge and literary presentation.

Prof. S. B. Junaidu

…….....	…….....	.....……
Chairman, Supervisory Committee	Signature	Date

Dr. A. A. Obiniyi

…….....	…….....	.....……
Member, Supervisory Committee	Signature	Date

…….....	…….....	.....……
Member, Supervisory Committee (External Examiner)	Signature	Date

Prof. A.A. Tijjani

…….....	…….....	.....……
H. O. D. Mathematics	Signature	Date

Prof. Kabiru Bala

…….....	…….....	.....……
Dean, Post Graduate School	Signature	Date

DEDICATION

This dissertation is dedicated to the memory of my father, Late D.S.P Salami Siyaka

ACKNOWLEDGEMENT

My first words of appreciation go to Allah Almighty for his unending mercies which I have enjoyed in the course of this study and in other issues of life. Praise be to Allah (SWT) alone. I am full of gratitude to several persons who have had definite impact in my life.

I will like to acknowledge the effort of my supervisor (major) Professor S.B. Junaidu for his efforts and time to see me through the completion of this thesis and my minor supervisor Dr. A.A Obiniyi for his contributions.

I also acknowledge the effort of the entire lecturers of the Department of Mathematics for their academic and intellectual support towards the success of this thesis.

A special acknowledgment goes to my Mother, Hajiya Salamatu Salami and my humble uncle Mallam Suleiman Mohammed Jamiu for their moral support throughout my entire study.

I cannot forget mentioning some of my colleagues and friends who contributed in one way or another towards the successful completion of this thesis; these people include Mal. Aliyu Kufena, Mal. Ibrahim Umar Enesi, Mal. Salawudeen Ridwan, Dr. Emmanuel Ajumuka, Miss Faiza Shuaib and my entire class members.

Finally, a special acknowledgment also goes to the University Management for their financial support (tuition fee) towards the successful completion of this study.

ABSTRACT

Educational data mining (EDM) is a field that concentrates on prediction and is known for its role in uncovering hidden information from large volume of data. EDM has seen an emergence of research leading to strategies that aim to address issues of higher education with primary focus on students' attrition. An unceasing student's attrition from the undergraduate programme of Ahmadu Bello University, Zaria (A.B.U, Zaria) has significant ramification for individuals affected by this problem and the society in general. This work examines the problem of predicting student's dropout in selected programmes of A.B.U Zaria. This work also focuses on addressing students' attrition by creating a first year at risk model to be used in early detection of students at the risk of attrition. Various factors that might influence students' attrition were collected from two (2) primary sources (student portal and exam processing software) of A.B.U Zaria information system. This work applies variants of existing classification algorithms in building predictive models; also an instance based learning algorithm (k nearest neighbor) was chosen and modified. An initial data preprocessing, feature selection and a 10-fold cross validation experiment was carried out in the model development phase of this work using WEKA (an open source tool for data mining tool). The classification algorithms employed in this work include: Multi-Layer Perceptron, Naïve Bayes, J48 Decision Tree, Sequential minimal optimization, K-Nearest Neighbor and a modified K-Nearest Neighbor. Results obtained showed J48 decision tree algorithm have performed nicely among all, based on our dataset with an average accuracy of 97.9%. Also the modified nearest neighbor algorithm performed next with an average accuracy of 97.3%. The result obtained were validated and analyzed using WEKA's experimenter. To choose the best model, we conducted a comparative analysis of the classifiers used in this dissertation work.

TABLE OF CONTENTS

DECLARATION ..... **i**

CERTIFICATION ..... **ii**

DEDICATION ..... **iii**

ACKNOWLEDGEMENT ..... **iv**

ABSTRACT ..... **v**

TABLE OF CONTENTS ..... **vi**

LIST OF TABLES ..... **xi**

LIST OF FIGURES ..... **xii**

CHAPTER ONE: GENERAL INTRODUCTION **1**

1.1 Background to the study **1**

1.2 Motivation of the study **3**

1.3 Statement of the problem **4**

1.4 Aim of the study **6**

1.5 Objectives of the study **6**

1.6 Methodology **7**

1.7 Scope of the study **8**

1.8 Contributions to Knowledge **8**

1.9 Thesis Organization **9**

CHAPTER TWO: LITERATURE REVIEW **10**

2.1 Introduction to Data Mining 10

2.1.1 What is Data Mining? 11

2.1.2 Data Mining and Knowledge Discovery 11

2.2 Motivating Challenges for Data Mining 12

2.2.1 Scalability 12

2.2.2 High Dimensionality 13

2.2.3 Heterogeneous and Complex Data 13

2.2.4 Data Ownership and Distribution 14

2.2.5 Non-traditional Analysis 14

2.3 The Origins of Data Mining 15

2.4 Data Mining Tasks 16

2.5 Machine Learning 17

2.5.1 Tom M. Mitchell 18

2.5.2 Supervised machine learning: 20

2.5.3 Unsupervised machine learning: 20

2.6 Classification Machine Learning Algorithms 22

2.6.1 k – Nearest Neighbor 22

2.6.2 Bayesian Classification 24

2.6.3 Bayes' Theorem 24

2.6.4 Rule-Based Classification 25

2.6.5 Decision Tree 26

2.7 Review of related work 30

CHAPTER THREE: PREDICTION MODELS DEVELOPMENT 34

3.1 Data Collection 34

3.1.1 PHPMyAdmin 35

3.1.2 Microsoft Excel 35

3.2 Data Cleaning 36

3.2.1 Python Programming Language 37

3.2.2 IPython 37

3.2.3 Pandas 38

3.3 Data Exploration 39

3.4 Data Integration. 40

3.5 Data Transformation 41

3.5.1 Numerical Coding of nominal attributes 43

3.5.2 Algorithm Pseudo code for converting Nominal to numeric values using rank frequency distribution 45

3.6 Data Splitting 45

3.7 Model validation 46

3.7.1 Cross validation 47

3.7.2 Holdout method 47

3.7.3 K-fold cross validation 47

..... 48

3.7.4 Leave-one-out cross validation 48

3.8 Attribute Weighting 49

3.9 Feature Selection 50

3.9.1 Embedded approaches 51

3.9.2 Filter approaches 51

3.9.3 Wrapper approaches 51

3.9.4 Information Gain Attribute Selection 51

3.10 Proposed Algorithm for k Nearest Neighbor using Information as Weight 53

3.11 Tools and Techniques used 54

3.11.1 WEKA 54

3.11.2 Scikit learn Python Library 55

3.12 Model Evaluation 56

3.13 Evaluation measures 56

3.13.1 Measures for classification evaluation 57

3.14 System Architecture 59

CHAPTER FOUR: RESULTS AND DISCUSSIONS 61

4.1 Data Set characteristics 61

4.2 Experiments 61

4.2.1 Experiment 1 (Department of Mathematics) 62

4.2.2 Experiment 2 (Department of Building) 63

4.2.3 Experiment 3 (Faculty of Agricultural Science) 64

4.2.4 Experiment 4 (Department of Mass Communication) 65

4.2.5 Experiment 5 (Faculty of Pharmacy) 66

4.2.6 Experiment 6 (Faculty of Law) 67

CHAPTER FIVE: SUMMARY AND FUTURE WORK 70

5.1 Summary 70

5.2 Conclusion 70

5.3 Future Work 71

References 72

LIST OF TABLES

TABLE 3.1: ATTRIBUTE SELECTED FROM DATASET 40

TABLE 3.2: EXAMPLE OF NUMERIC & NOMINAL ATTRIBUTE IN DATASET..... 43

TABLE 3.3: RANK OF NOMINAL DATA 44

TABLE 3.3: ATTRIBUTE SELECTION USING IG & RANKER SEARCH METHOD..... 53

TABLE 4.1: EXPERIMENT MAPPING TO FACULTIES AND DEPARTMENTS..... 61

TABLE 4.2: STATISTICAL ANALYSIS OF CLASSIFIERS WITH CROSS VALIDATION (MATHEMATICS) 622

TABLE 4.3 STATISTICAL ANALYSIS OF CLASSIFIERS WITH CROSS VALIDATION (BUILDING) 633

TABLE 4.4: STATISTICAL ANALYSIS OF CLASSIFIERS WITH CROSS VALIDATION (*AGRIC SCIENCE*) 644

TABLE 4.5: STATISTICAL ANALYSIS OF CLASSIFIERS WITH CROSS VALIDATION (*MASS COMM*) .. 655

TABLE 4.6: STATISTICAL ANALYSIS OF CLASSIFIERS WITH CROSS VALIDATION (*PHARMACY*)..... 666

TABLE 4.7: STATISTICAL ANALYSIS OF CLASSIFIERS WITH CROSS VALIDATION (*LAW*) 677

TABLE 4.8: SUMMARY OF ACCURACY MEASURE FOR ALL CLASSIFIERS USED..... 688

LIST OF FIGURES

FIGURE 1.1: STUDENT ATTRITION IN DEPARTMENT OF MATHEMATICS, A.B.U ZARIA 5

FIGURE 2.1:AN EXAMPLE OF DECISION TREE 27

FIGURE 3.1: DATA PARTITION (TRAINING AND TEST SET). 46

FIGURE 3.2: 10 FOLD CROSS VALIDATION 48

FIGURE 3.3: CONFUSION MATRIX 577

FIGURE 3.4: SYSTEM ARCHITECTURE 57

FIGURE 4.1: WEKA KNOWLEDGE FLOW MODEL FOR ALL EXPERIMENTS 688

FIGURE 4.2: ACCURACY COMPARISON CHART 69

CHAPTER ONE: GENERAL INTRODUCTION

This chapter discusses the introductory part of the dissertation which includes background of the study, statement of the problem, research motivation, aim and objectives, scope of the research, the methodology used to carry out the research and finally the dissertation contribution to knowledge.

1.1 Background to the study

The knowledge discovery process is as old as Homo sapiens (Gorunescu, 2011). Knowledge discovery in databases (KDD), is an overall process of converting raw data into useful information. Data mining also called KDD is the field of discovering new and potentially useful information in large databases (Parneer *et al.*, 2015). Fortunately, in recent decades major challenges associated with KDD have been tackled as a result of the development of data mining technologies. By means of digging intelligently in different large databases, data mining aims to extract implicit, previously unknown and potentially useful information from data, since "knowledge is power" (Gorunescu, 2011).

Human capabilities of both generating and collecting data have been increasing rapidly (Jiawei & Micheline , 2006); contributing factors include the computerization of business, scientific, and government transactions; the widespread use of digital cameras, publication tools, and bar codes for most commercial products; and advances in data collection tools ranging from scanned text and image platforms to satellite remote sensing systems. In addition, popular use of the World Wide Web as a global information system has flooded us with a tremendous amount of data and information (Yong *et al*, 2014). This explosive growth in stored or transient data has generated an urgent need for new techniques and automated tools that can intelligently assist us in transforming the vast amounts of data into useful information and knowledge.

The world is overwhelmed by data - scientific data, medical data, demographic data, educational data, financial data, and marketing data; people have no time to look at this data. Human attention has become the precious resource. So, people must find ways to automatically analyze, classify, summarize, discover data and characterize trends in it, and to automatically flag anomalies. This is one of the most active and exciting areas of the database research community. Researchers in areas such as statistics, visualization, artificial intelligence, and machine learning are contributing to this field. The breadth of the field makes it difficult to grasp the extraordinary progress over the last few decades (Jiawei & Micheline , 2006).

Data mining is a multidisciplinary field, drawing work from areas including database technology, machine learning, statistics, pattern recognition, information retrieval, neural networks, knowledge-based systems, artificial intelligence, high-performance computing, and data visualization (Yong *et al*, 2014). The incessant advantages of data mining have landed its application in numerous fields including the educational research which is commonly known as Educational Data Mining (EDM).

EDM is defined as an emerging discipline, concerned with developing methods for exploring the unique types of data that come from the educational setting, and using those methods to better understand students, and the settings which they learn in (International EDM Society, 2011). It is often stressed with the improvement of student models which denote the student's current knowledge, motivation, metacognition, and attitudes.

It has been proposed that educational data mining methods are more often different from standard data mining methods, due to the need to explicitly account for (and the opportunities to exploit) the multi-level hierarchy and non-independence in educational data. EDM methods are drawn from a variety of literature, including data mining and machine learning,

psychometrics and other areas of statistics, information visualization, and computational modelling (Baker & Yacef, 2009).

Higher education systems all over the world nowadays are challenged by the new information and communication technologies (Boufardea & Garofalaskis, 2012). An issue of concern in higher education institutions across the world is the retention and success of students in their studies. With the increasing competition among higher education institutes, most are focusing on how to increase students' retention rates and numbers of completions; university performance is one means of measuring its quality and reputation (Jusoff *et al*, 2008); thus higher education institution are becoming more interested in predicting the paths of students, and identifying which students will require assistance in order to graduate (Luan, 2016). Higher education institutions encounter many problems which keep them away from achieving this objective. Some of these problems stem from knowledge gap which is the lack of significant knowledge at the educational main processes such as counselling, planning, registration, evaluation (Kuyoro et al., 2013). For instance, many learning institutions do not have access to the necessary information to counsel students, thus they are unable to give suitable recommendation or advisory services to the students. However, recent studies have considered data mining as a suitable technology in giving additional insight into educational entities (student, lecturer, staff, and alumni) and main processes (Naeimeh et al., 2008).

1.2 Motivation of the study

Since the 1960s, database and information technology have been evolving systematically from primitive file processing systems to sophisticated database systems. The idea of gaining knowledge through specialized analysis of mass data is as old as the evolution of databases and steadily increased both in the amount of data processed and the sophistication of question people try to answer. The abundance of data, coupled with the need for powerful data analysis

tools, has been described as a data rich but information poor situation (Jiawei & Micheline , 2006).

As A.B.U Zaria takes a lead in in-house software development among Nigerian universities to become e-complaint (Lawal, 2015), the university developed a host of software applications (such as student and staff portals, examination records processing and CBT applications, etc.) generating huge amount of data on students and staff.

As these data grow with time, since 2008, it is highly desirable that the university takes steps to create information value out of this growing body of data. For instance, it will be highly beneficial to a number of stakeholders to gain insight into the attrition pattern of students in various programs of the university.

A driving motivation for this dissertation is to exploit students' data, as obtainable from ABU in-house software applications, with a view to creating information value from these data with respect to students' attrition.

1.3 Statement of the problem

Students' attrition can be seen as a departure from all forms of higher education prior to completion of a degree or other credential. It tends to be a systematic concern for many higher education programmes such as sciences, engineering and medicine etc.

Student attrition is the reduction in numbers of students attending courses as time goes by. Concern about the student has led to much research on college student attrition and retention (Ramist, 1981).

College students attrition has been a problem studied for decades (Seidman, 2005). Student's attrition is important as a result of the negative impact it has on the economy. From the

institutional perspective, attrition is considered a waste of university resource, especially in environment of limited financial and general resources (Stillman, 2016)

The problem of unceasing students' attrition from undergraduate programmes in A.B.U., Zaria as exemplified in **Figure 1.1** cannot be overemphasized.

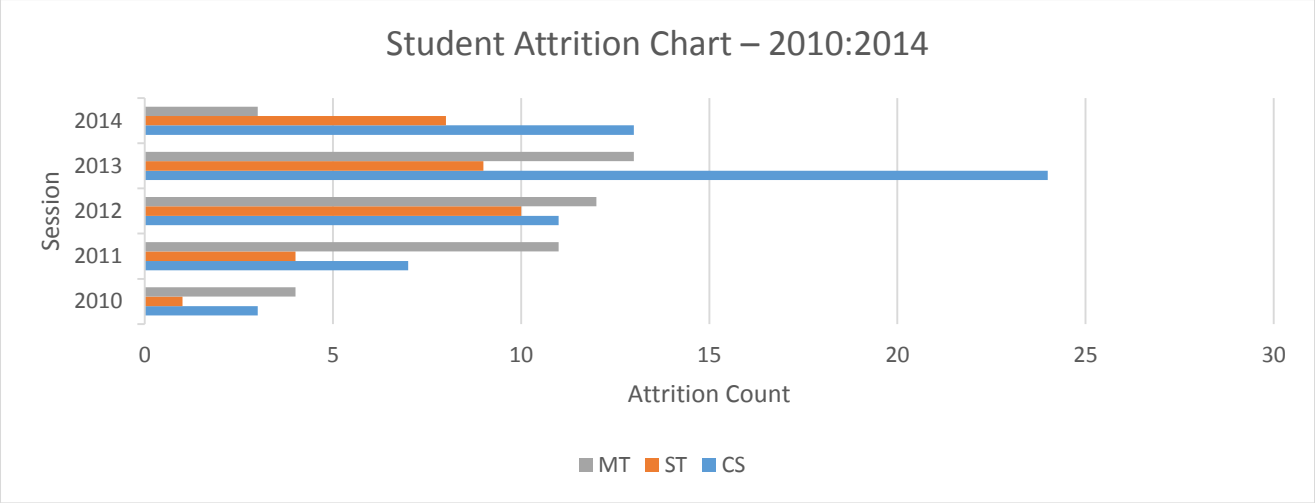


Figure 1.1: Student attrition in Department of Mathematics, A.B.U Zaria

Figure 1.1 expresses the unceasing rate of students' attrition by showing the number of students drop outs in Mathematics, Computer Science and Statistics programme of A.B.U Zaria from 2010 to 2014. As expressed by Figure 1.1, Computer science section recorded the highest number of students drop out in 2013 session while the Statistics section recorded the lowest in 2010 session.

The complexity of this problem is multi-faceted as student experiences and expectations are not same for all institutions.

Attrition is important because it damages the reputation of a university and creates long term implications for attracting new students (Ozga & Sukhnadan, 1998); attrition represents a loss of income for universities and, all things being equal, a failure to accomplish their educational

mission (Bean, 1990). Low attrition rates also impact upon the image of a university from the perspective of stakeholders, parents and students.

However failure to address the ramifications of university student's attrition exacerbates a negative influence on the society through an unskilled workforce. Tinto, (2004) compared unemployment rate among high school graduates and university graduates who were unemployed stating that those without a university degree were unemployed at twice the rate of those with a university degree.

1.4 Aim of the study

The overall aim of this study is to build a first year at risk model and to apply the model to datasets of selected departments in A.B.U Zaria for early detection of students at the risk of attrition.

1.5 Objectives of the study

The objectives of this study are to:

1. Identify variables that have potentially high effect on students' attrition and generate a data source of predictive variables.
2. Design and implement a modified k-Nearest Neighbor classifier in Python programming language
3. Build, validate and evaluate predictive models on available data using six classifiers namely:
    a. K-Nearest Neighbors
    b. Modified K-Nearest Neighbors
    c. Multilayer perception
    d. Naïve Bayes
    e. J48

f. SMO

4. Compare selected classifiers to determine the best for predicting students' dropout based on selected departments in A.B.U Zaria.

1.6 Methodology

The following are the proposed steps needed for the realization of this research work.

1. Review of related literatures to:
    a. Understand the applicability of data mining in an educational setting.
2. Identification and collection of data from primary sources:
    a. A.B.U Zaria exam processing software (Exams-LOGIC).
    b. A.B.U Zaria student portal.
3. Execute data preprocessing tasks by using data exploration, cleaning and transformation.
4. Conduct data transformation by:
    a. Summarizing the subset of data scattered around multiple tables.
    b. Adding derived attributes where applicable.
5. Conduct relevant attribute selection by using the filter technique (in which attribute subset are selected independent of the classifiers).
6. Transforming nominal attributes to numeric and adding weights to attributes using information gain of decision trees (IG)
7. Selection and modification of an instance based learning algorithm (k Nearest Neighbor);
8. Build and validate predictive models on selected dataset with modified kNN, Multilayer perceptron, Naïve Bayes, J48, SMO and traditional kNN algorithms.

9. Perform data validation task by partitioning the dataset into training and test instances using k-fold cross validation.
10. Evaluate, interpret and present results generated by the classifiers using the accuracy measure.

1.7 Scope of the study

This study is scoped to the academic profiles of students in selected faculties and departments of A.B.U Zaria. The selection is driven by the availability and accessibility of data. Below is the list of selected faculties and departments:

1. Department of Mathematics
2. Faculty of Pharmaceutical Sciences
3. Department of Building Engineering
4. Department of Mass Communication
5. Faculty of Law
6. Faculty of Agriculture.

1.8 Contributions to Knowledge

Data mining techniques have been successfully applied in many different fields and several contributions have been made in the field of EDM towards addressing students' attrition in higher education. A recent work (Parneer *et al*, 2015) focused on identifying slow learners and displaying via a predictive data mining model using classification algorithms such as Multi-layer perceptron, Naïve Bayes, SMO, J48, Reptree and Support vector machine (SVM).

This dissertation extended the work of (Parneer *et al*, 2015) and made the following contributions:

1. Generated data source of predictive variables from ABU Zaria software applications.
2. Adopted and modified an instance based learning algorithm (k Nearest Neighbor).
3. Built and validated models for predicting attrition among A.B.U Zaria students
4. Reported comprehensively the accuracy of the predictive models developed.
5. Conducted an accuracy comparison of all classifiers used in this work.

1.9 Thesis Organization

The rest of this work is structured as follows:

Chapter 2: Gives introductory notes on related researches in the direction that this thesis concentrates on, and it also goes further to unveil relevant concepts behind supervised and unsupervised machine learning. It also moves further to explain data, data mining algorithms used in this work and predictive modelling. Also in this chapter, WEKA (Waikato Environment for Knowledge Analysis – an open source tool for data mining) is introduced.

Chapter 3: Concentrates on the prediction models development; also experiments were carried out based on selected classification algorithms. In the process, the following steps were discussed: data collection, cleaning, exploration, integration, transformation and feature selection. Each of these processes were described before discussing models evaluation issues.

Chapter 4: Gives a discussion of results of the experiments carried out in the model build up process; also shown is the experiments involving cross validating prepared dataset with selected classification algorithms. This chapter also shows statistics of results obtained after experiments using tables, charts and the knowledge flow in WEKA.

Chapter 5: Wraps up the work done in the course of the research. The classification models built have been summarized; this chapter also reports the result of classifiers comparison using the accuracy metric. Furthermore, the limitations encountered with regards to the work are also listed. Recommendation are given and, lastly, a clear direction for future work is drawn.

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction to Data Mining

Rapid advances in data collection and storage technology have enabled organizations to accumulate vast amounts of data. However, extracting useful information has proven extremely challenging. Often, traditional data analysis tools and techniques cannot be used because of the massive size of a data set. Sometimes, the non-traditional nature of the data means that traditional approaches cannot be applied even if the data set is relatively small. In other situations, the questions that need to be answered cannot be addressed using existing data analysis techniques, and thus, new methods need to be developed (Wong & Li, 2006).

Data mining is a technology that blends traditional data analysis methods with sophisticated algorithms for processing large volumes of data. It has also opened up exciting opportunities for exploring and analyzing new types of data and for analyzing old types of data in new ways. Business Point-of-sale data collection (bar code scanners, radio frequency identification (RFID), and smart card technology) have allowed retailers to collect up-to-the-minute data about customer purchases at the checkout counters of their stores. Retailers can utilize this information, along with other business-critical data such as Web logs from e-commerce Web sites and customer service records from call centers, to help them better understand the needs of their customers and make more informed business decisions (Pang-Ning *et al*, 2009).

Data mining techniques can be used to support a wide range of business intelligence applications such as customer profiling, targeted marketing, workflow management, store layout, and fraud detection. It can also help retailers answer important business questions such as "Who are the most profitable customers?" "What products can be cross-sold or up-sold?" and "What is the revenue outlook of the company for next year?"

Researchers in medicine, science, and engineering are rapidly accumulating data that is key to important new discoveries (Pang-Ning *et al*, 2009). For example, as an important step toward

improving our understanding of the Earth's climate system, NASA has deployed a series of Earth orbiting satellites that continuously generate global observations of the land surface, oceans, and atmosphere. However, because of the size and spatiotemporal nature of the data, traditional methods are often not suitable for analyzing these data sets.

2.1.1 What is Data Mining?

Data mining also called knowledge discovery in Databases (KDD), is the field of discovering novel and potentially useful information from large amount of data (Witten & Frank, 2005). Data mining is the process of automatically discovering useful information in large data repositories. Data mining techniques are deployed to scour large databases in order to find novel and useful patterns that might otherwise remain unknown. They also provide capabilities to predict the outcome of a future observation, such as predicting whether a newly arrived university students would be retained or not during his course of study. Not all information discovery tasks are considered to be data mining. For example, looking up individual records using a database management system or finding particular Web pages via a query to an Internet search engine are tasks related to the area of information retrieval. Although such tasks are important and may involve the use of the sophisticated algorithms and data structures, they rely on traditional computer science techniques and obvious features of the data to create index structures for efficiently organizing and retrieving information.

Nonetheless, data mining techniques have been used to enhance information retrieval systems (Rahman & Ramos, 2013).

2.1.2 Data Mining and Knowledge Discovery

Data mining is an integral part of knowledge discovery in databases (KDD), which is the overall process of converting raw data into useful information, as shown in Figure 2.1 This process consists of a series of transformation steps, from data preprocessing to post processing of data mining results.

The input data can be stored in a variety of formats (flat files, spreadsheets, or relational tables) and may reside in a centralized data repository or be distributed across multiple sites. The purpose of preprocessing is to transform the raw input data into an appropriate format for subsequent analysis. The steps involved in data preprocessing include fusing data from multiple sources, cleaning data to remove noise and duplicate observations, and selecting records and features that are relevant to the data mining task at hand. Because of the many ways data can be collected and stored, data preprocessing is perhaps the most laborious and time-consuming step in the overall knowledge discovery process (Bruno, 2009).

"Closing the loop" is the phrase often used to refer to the process of integrating data mining results into decision support systems. For example, in business applications, the insights offered by data mining results can be integrated with campaign management tools so that effective marketing promotions can be conducted and tested. Such integration requires a post processing step that ensures that only valid and useful results are incorporated into the decision support system. An example of post processing is visualization, which allows analysts to explore the data and the data mining results from a variety of viewpoints. Statistical measures or hypothesis testing methods can also be applied during post processing to eliminate spurious data mining results.

2.2 Motivating Challenges for Data Mining

Traditional data analysis techniques have often encountered practical difficulties in meeting the challenges posed by new data sets. The following are some of the specific challenges that motivated the development of data mining.

2.2.1 Scalability

Because of advances in data generation and collection, data sets with sizes of gigabytes, terabytes, or even petabytes are becoming common. If data mining algorithms are to handle

these massive data sets, then they must be scalable. Many data mining algorithms employ special search strategies to handle exponential search problems. Scalability may also require the implementation of novel data structures to access individual records in an efficient manner. For instance, out-of-core algorithms may be necessary when processing data sets that cannot fit into main memory. Scalability can also be improved by using sampling or developing parallel and distributed algorithms (Pang-Ning *et al*, 2009).

2.2.2 High Dimensionality

It is now common to encounter data sets with hundreds or thousands of attributes instead of the handful common a few decades ago (Pang-Ning *et al*, 2009). In bioinformatics, progress in microarray technology has produced gene expression data involving thousands of features. Data sets with temporal or spatial components also tend to have high dimensionality. For example, consider a data set that contains measurements of temperature at various locations. If the temperature measurements are taken repeatedly for an extended period, the number of dimensions (features) increases in proportion to the number of measurements taken. Traditional data analysis techniques that were developed for low-dimensional data often do not work well for such high dimensional data. Also, for some data analysis algorithms, the computational complexity increases rapidly as the dimensionality (the number of features) increases.

2.2.3 Heterogeneous and Complex Data

Traditional data analysis methods often deal with data sets containing attributes of the same type, either continuous or categorical. As the role of data mining in business, science, medicine, education and other fields has grown, so has the need for techniques that can handle heterogeneous attributes (Pang-Ning *et al*, 2009).

Recent years have also seen the emergence of more complex data objects (Pang-Ning *et al*, 2009). Examples of such non-traditional types of data include collections of Web pages containing semi-structured text and hyperlinks; DNA data with sequential and three-dimensional structure; and climate data that consists of time series measurements (temperature,

pressure, etc.) at various locations on the earth's surface. Techniques developed for mining such complex objects should take into consideration relationships in the data, such as temporal and spatial autocorrelation, graph connectivity, and parent-child relationships between the elements in semi-structured text and XML documents.

2.2.4 Data Ownership and Distribution

Sometimes, the data needed for an analysis is not stored in one location or owned by one organization. Instead, the data is geographically distributed among resources belonging to multiple entities. This requires the development of distributed data mining techniques. Among the key challenges faced by distributed data mining algorithms include: how to reduce the amount of communication needed to perform the distributed computation, how to effectively consolidate the data mining results obtained from multiple sources, and 3. how to address data security issues (Pang-Ning *et al.*, 2009)

2.2.5 Non-traditional Analysis

The traditional statistical approach is based on a hypothesize-and-test paradigm. In other words, a hypothesis is proposed, an experiment is designed to gather the data, and then the data is analyzed with respect to the hypothesis. Unfortunately, this process is extremely labor intensive. Current data analysis tasks often require the generation and evaluation of thousands of hypotheses, and consequently, the development of some data mining techniques has been motivated by the desire to automate the process of hypothesis generation and evaluation.

Furthermore, the data sets analyzed in data mining are typically not the result of a carefully designed experiment and often represent opportunistic samples of the data, rather than random samples. Also, the data sets frequently involve non-traditional types of data and data distributions (Pang-Ning *et al.*, 2009).

2.3 The Origins of Data Mining

Brought together by the goal of meeting the challenges of data mining, researchers from different disciplines began to focus on developing more efficient and scalable tools that could handle diverse types of data. This work, which culminated in the field of data mining, built upon the methodology and algorithms that researchers had previously used. In particular, data mining draws upon ideas, such as (1) sampling, estimation, and hypothesis testing from statistics and (2) search algorithms, modeling techniques, and learning theories from artificial intelligence, pattern recognition, and machine learning (Information Management Resource Association, 2015).

The usage of data mining started with Structured Query Language (SQL) developed at IBM by Donald D. Chamberlin and Raymond F. Boyce in the early 1970s (Date & Darwen, 1997). SQL was originally called Structured English Query Language (SEQUEL) and then gained reputation with the Multi State Antiterrorism Information Exchange (MATRIX) (Chamberlin & Boyce, 2016).

Data mining has also been quick to adopt ideas from other areas, including optimization, evolutionary computing, information theory and signal processing, visualization, and information retrieval (Pang-Ning *et al.*, 2009).

A number of other areas also play key supporting roles. In particular, database systems are needed to provide support for efficient storage, indexing, and query processing. Techniques from high performance (parallel) computing are often important in addressing the massive size of some data sets. Distributed techniques can also help address the issue of size and are essential when the data cannot be gathered in one location.

2.4 Data Mining Tasks

According to (Pang-Ning *et al.*, 2009), data mining tasks are generally divided into two major categories:

1. Predictive tasks. The objective of these tasks is to predict the value of a particular attribute based on the values of other attributes. The attribute to be predicted is commonly known as the target or dependent variable, while the attributes used for making the prediction are known as the explanatory or independent variables.
2. Descriptive tasks. Here, the objective is to derive patterns (correlations, trends, clusters, trajectories, and anomalies) that summarize the underlying relationships in data. Descriptive data mining tasks are often exploratory in nature and frequently require post processing techniques to validate and explain the results.

Predictive modeling:

There are several applications for Machine Learning (ML), the most significant of which is predictive data mining. Every instance in any dataset used by machine learning algorithms is represented using the same set of features (Kotsiantis *et al.*, 2007).

Predictive modelling refers to the task of building a model for the target variable as a function of the explanatory variables. There are two types of predictive modeling tasks: classification, which is used for discrete target variables, and regression, which is used for continuous target variables. For example, predicting whether a student will be retained or not is a classification task because the target variable is binary-valued. On the other hand, forecasting the future price of a stock is a regression task because price is a continuous-valued attribute. The goal of both tasks is to learn a model that minimizes the error between the predicted and true values of the target variable. Predictive modeling can be used to identify customers that will respond to a

marketing campaign, predict disturbances in the Earth's ecosystem, or judge whether a patient has a particular disease based on the results of medical tests.

***Association analysis*:** is used to discover patterns that describe strongly associated features in the data. The discovered patterns are typically represented in the form of implication rules or feature subsets. Because of the exponential size of its search space, the goal of association analysis is to extract the most interesting patterns in an efficient manner. Useful applications of association analysis include finding groups of genes that have related functionality, identifying Web pages that are accessed together, or understanding the relationships between different elements of Earth's climate system.

2.5 Machine Learning

Machine learning is a subfield of computer science that evolved from the study of pattern recognition and computational learning theory in artificial intelligence. Arthur Samuel (1959) defined machine learning as a field of study that gives computers the ability to learn without being explicitly programmed.

Turing, (1950) in his paper asked, Can machines think? The paper describes the "Imitation Game," which involves three participants: a human acting as a judge, another human, and a computer that is attempting to convince the judge that it is human. The judge would type into a terminal program to "talk" to the other two participants. Both the human and the computer would respond, and the judge would decide which response came from the computer. If the judge couldn't consistently tell the difference between the human and computer responses then the computer won the game.

The test continues today in the form of the Loebner Prize, an annual competition in artificial intelligence. The aim is simple enough: Convince the judges that they are chatting to a human instead of a computer chat bot program.

Arthur, (1959) defined machine learning as, "A field of study that gives computers the ability to learn without being explicitly programmed." Samuel is credited with creating one of the self-learning computer programs with his work at IBM. The focused was on games as a way of getting the computer to learn things. The game of choice for Samuel was checkers because it is a simple game but requires strategy from which the program could learn. With the use of alphabet evaluation pruning (eliminating nodes that do not need evaluating) and minimax (minimizing the loss for the worst case) strategies, the program would discount moves and thus improve costly memory performance of the program. This work was widely known in artificial intelligence, but this author was also noted for being one of the first programmers to use hash tables, and he certainly made a big impact at IBM.

2.5.1 Tom M. Mitchell

Tom, (1997), stated that: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with the experience E. The important thing here is that you now have a set of objects to define machine learning:

1. Task (T), either one or more
2. Experience (E)
3. Performance (P)

Machine learning is a branch of artificial intelligence. Using computing, we design systems that can learn from data in a manner of being trained. The systems might learn and improve

with experience, and with time, refine a model that can be used to predict outcomes of questions based on the previous learning.

Machine learning is so pervasive today that you probably use it dozens of times a day without knowing it. Many researchers also think it is the best way to make progress towards human-level AI. Machine learning studies computer algorithms for learning to do stuff. A person might, for instance, be interested in learning to complete a task, or to make accurate predictions, or to behave intelligently. The learning that is being done is always based on some sort of observations or data. So in general, machine learning is about learning to do better in the future based on what was experienced in the past. The emphasis of machine learning is on automatic methods. In other words, the goal is to devise learning algorithms that do the learning automatically without human intervention or assistance. The machine learning paradigm can be viewed as "programming by example." Often we have a specific task in mind, such as spam filtering. But rather than program the computer to solve the task directly, in machine learning, people seek methods by which the computer will come up with its own program based on examples provided. Machine learning is a core subarea of artificial intelligence. Although a subarea of AI, machine learning also intersects broadly with other fields, such as statistics mathematics, physics and theoretical computer science. Some examples of machine learning problems are: optical character recognition: categorize images of handwritten characters by the letters represented; face detection: find faces in images (or indicate if a face is present); spam filtering -identify email messages as spam or non-spam; topic spotting - categorize news articles such as as to whether they are about politics, sports and entertainment spoken language understanding: within the context of a limited domain, determine the meaning and lastly not the least prediction of students performance in courses.

Machine learning algorithms are classified into two (2) distinct group supervised machine learning and unsupervised machine learning:

2.5.2 Supervised machine learning:

Supervised learning is a data mining task of inferring a function from labeled training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize decisions from the training data to unseen situations in a "reasonable" way. Supervised learning includes two categories of algorithms namely Classification: for categorical response values, where the data can be separated into specific "classes"; Regression: for continuous-response values;

Common classification algorithms include: Support vector machines (SVM),, Neural networks, Naïve Bayes classifier, Decision trees; Nearest neighbors (kNN) and common regression algorithms include: Linear regression; Logistics regression etc.

2.5.3 Unsupervised machine learning:

Unsupervised learning is the machine learning task of inferring a function to describe hidden structure from unlabeled data. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution. This distinguishes unsupervised learning from supervised learning.

Unsupervised learning is closely related to the problem of density estimation in statistics. However unsupervised learning also encompasses many other techniques that seek to summarize and explain key features of the data. Many methods employed in unsupervised learning are based on data mining methods used to preprocess data. The most common unsupervised learning method is cluster analysis, which is used for exploratory data analysis to find hidden patterns or grouping in data. The clusters are modeled using a measure of

similarity which is defined upon metrics such as Euclidean or probabilistic distance. Common clustering algorithms include: Hierarchical clustering, k-Means clustering and Gaussian mixture models etc.

2.6 Classification Machine Learning Algorithms

Supervised classification is one of the tasks most frequently carried out by intelligent systems. (Kotsiantis et al, 2006) Thus, a large number of techniques have been developed based on Artificial Intelligence (Logic-based techniques, Perceptron-based techniques) and Statistics (Bayesian Networks, Instance-based techniques).

The goal of supervised learning is to build a concise model of the distribution of class labels in terms of predictor features. The resulting classifier is then used to assign class labels to the testing instances where the values of the predictor features are known, but the value of the class label is unknown. This section describes various classification algorithms.

2.6.1 k – Nearest Neighbor

The k-nearest-neighbor method was first described in the early 1950s. The method is labor intensive when given large training sets, and did not gain popularity until the 1960s when increased computing power became available. It has since been widely used in the area of pattern recognition (Jiawei & Micheline 2006)

Nearest-neighbor classifiers are based on learning by analogy, that is, by comparing a given test tuple with training tuples that are similar to it. The training tuples are described by n attributes. Each tuple represents a point in an n-dimensional space. In this way, all of the training tuples are stored in an n-dimensional pattern space. When given an unknown tuple, a k-nearest-neighbor classifier searches the pattern space for the k training tuples that are closest to the unknown tuple. These k training tuples are the k "nearest neighbors" of the unknown tuple. "Closeness" is defined in terms of a distance metric, such as Euclidean distance (Natarajan *et al.*, 2012).

For k-nearest-neighbor classification, the unknown tuple is assigned the most common class among its k nearest neighbors. When k = 1, the unknown tuple is assigned the class of the

training tuple that is closest to it in pattern space. Nearest neighbor classifiers can also be used for prediction, that is, to return a real-valued prediction for a given unknown tuple. In this case, the classifier returns the average value of the real-valued labels associated with the k nearest neighbors of the unknown tuple.

But how can distance be computed for attributes that is not numeric, but categorical, such as color? The above discussion assumes that the attributes used to describe the tuples are all numeric. For categorical attributes, a simple method is to compare the corresponding value of the attribute in tuple X1 with that in tuple X2. If the two are identical (e.g., tuples X1 and X2 both have the color blue), then the difference between the two is taken as 0. If the two are different (e.g., tuple X1 is blue but tuple X2 is red), then the difference is considered to be 1. Other methods may incorporate more sophisticated schemes for differential grading (e.g., where a larger difference score is assigned, say, for blue and white than for blue and black).

How can I determine a good value for k, the number of neighbors? This can be determined experimentally. Starting with k = 1, A test set is used to estimate the error rate of the classifier. This process can be repeated each time by incrementing k to allow for one more neighbor. The k value that gives the minimum error rate may be selected. In general, the larger the number of training tuples is, the larger the value of k will be (so that classification and prediction decisions can be based on a larger portion of the stored tuples).

Nearest-neighbor classifiers use distance-based comparisons that intrinsically assign equal weight to each attribute. They therefore can suffer from poor accuracy when given noisy or irrelevant attributes. The method, however, has been modified to incorporate attribute weighting and the pruning of noisy data tuples. The choice of a distance metric can be critical. The Manhattan distance or other distance measurements, may also be used (Jiawei & Micheline , 2006).

2.6.2 Bayesian Classification

Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given tuple belongs to a particular class. Bayesian classification is based on Bayes' theorem (Jiawei & Micheline , 2006), described below. Studies comparing classification algorithms have found a simple Bayesian classifier known as the naïve Bayesian classifier to be comparable in performance with decision tree and selected neural network classifiers. Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases. Naïve Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computations involved and, in this sense, is considered "naïve." Bayesian belief networks are graphical models, which unlike naïve Bayesian classifiers, allow the representation of dependencies among subsets of attributes. Bayesian belief networks can also be used for classification (Sumathi, 2007).

2.6.3 Bayes' Theorem

Bayes' theorem is named after Thomas Bayes, a nonconformist English clergy man who did early work in probability and decision theory during the 18th century. Let X be a data tuple. In Bayesian terms, X is considered "evidence." As usual, it is described by measurements made on a set of n attributes. Let H be some hypothesis, such as that the data tuple X belongs to a specified class C. For classification problems, to determine P(H|X), the probability that the hypothesis H holds given the "evidence" or observed data tuple X. In other words, finding the probability that tuple X belongs to class C, given that the attribute description of X is known. P(H|X) is the posterior probability, or a posteriori probability, of H conditioned on X. For example, suppose the selected world of data tuples is confined to customers described by the attributes age and income, respectively, and that X is a 35-year-old customer with an income of 40,000. Suppose that H is the hypothesis that our customer will buy a computer. Then P(H|X)

reflects the probability that customer X will buy a computer given that we know the customer's age and income.

In contrast, P(H) is the prior probability, or a priori probability, of H. For our example, this is the probability that any given customer will buy a computer, regardless of age, income, or any other information, for that matter. The posterior probability, P(H|X), is based on more information (e.g., customer information) than the prior probability, P(H), which is independent of X.

Similarly, P(X|H) is the posterior probability of X conditioned on H. That is, it is the probability that a customer, X, is 35 years old and earns 40,000, given that we know the customer will buy a computer. P(X) is the prior probability of X. Using our example, it is the probability that a person from our set of customers is 35 years old and earns 40,000.

"How are these probabilities estimated?" P(H), P(X|H), and P(X) may be estimated from the given data, as we shall see below. Bayes' theorem is useful in that it provides a way of calculating the posterior probability, P(H|X), from P(H), P(XjH), and P(X). Bayes' theorem is

$$P(H|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|H)P(H)}{P(\boldsymbol{X})} \tag{2.1}$$

2.6.4 Rule-Based Classification

In Rule-based classifiers the learned model is represented as a set of IF-THEN rules. An examination of how such rules are used for classification is considered; then a study of how such rules are generated is also considered; either from a decision tree or directly from the training data using a sequential covering algorithm (Jiawei & Micheline , 2006).

Using IF-THEN Rules for Classification

Rules are a good way of representing information or bits of knowledge. A rule-based classifier uses a set of IF-THEN rules for classification. An IF-THEN rule is an expression of the form IF condition THEN conclusion.

An example is rule R1, R1: IF age = youth AND student = yes THEN buys computer = yes.

The "IF"-part (or left-hand side) of a rule is known as the rule antecedent or precondition. The "THEN"-part (or right-hand side) is the rule consequent. In the rule antecedent, the condition consists of one or more attribute tests (such as age = youth, and student = yes) that are logically ANDed. The rule's consequent contains a class prediction (in this case, predicting whether a customer will buy a computer or not). R1 can also be written as R1: (age = youth) ^ (student = yes)) (buys computer = yes).

If the condition (that is, all of the attribute tests) in a rule antecedent holds true for a given tuple, then the rule antecedent is satisfied (or simply, that the rule is satisfied) and that the rule covers the tuple.

A rule R can be assessed by its coverage and accuracy. Given a tuple, X, from a class labeled dataset, D, let n covers be the number of tuples covered by R; n correct be the number of tuples correctly classified by R; and |D| be the number of tuples in D. A definition of coverage and accuracy of R is given as : coverage(R) = ncovers / |D|; accuracy(R) = ncorrect / ncovers.

That is, a rule's coverage is the percentage of tuples that are covered by the rule (i.e. whose attribute values hold true for the rule's antecedent). For a rule's accuracy, it is essential to look at the tuples that it covers and see what percentage of them the rule can correctly classify..

2.6.5 Decision Tree

Decision tree induction is the learning of decision trees from class-labeled training tuples. A decision tree is a flowchart-like tree structure, where each internal node (non-leaf node) denotes

a test on an attribute, each branch represents an outcome of the test, and each leaf node (or terminal node) holds a class label. The topmost node in a tree is the root node.

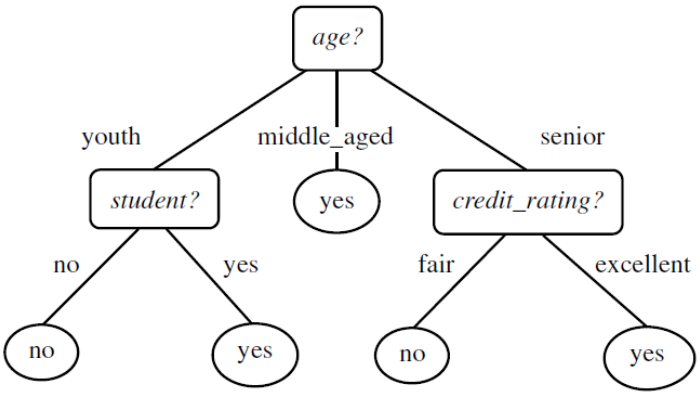


Figure 2.1: An example of a Decision Tree

A typical decision tree is shown in Figure 2.1. It represents the concept buys computer, that is, it predicts whether a customer at is likely to purchase a computer. Internal nodes are denoted by rectangles, and leaf nodes are denoted by ovals. Some decision tree algorithms produce only binary trees (where each internal node branches to exactly two other nodes), whereas others can produce non binary trees.

"How are decision trees used for classification?" Given a tuple, X, for which the associated class label is unknown, the attribute values of the tuple are tested against the decision tree. A path is traced from the root to a leaf node, which holds the class prediction for that tuple. Decision trees can easily be converted to classification rules.

"Why are decision tree classifiers so popular?" The construction of decision tree classifiers does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high dimensional data. Their representation of acquired knowledge in tree form is intuitive and generally easy to assimilate

by humans. The learning and classification steps of decision tree induction are simple and fast. In general, decision tree classifiers have good accuracy.

However, successful use may depend on the data at hand. Decision tree induction algorithms have been used for classification in many application areas, such as medicine, manufacturing and production, financial analysis, astronomy, and molecular biology. Decision trees are the basis of several commercial rule induction systems.

Most of the decision tree based systems use some form of general procedure for decision tree induction such as ID3, C45 and CART. Typically the tree is built recursively by picking a feature *f* at each step and dividing the training the collection into two sub collections, one containing *f* and the other not containing *f*, until only documents of a single category remains – at which point a leaf node is generated. The choice of a feature at each step is made by some information theoretic measure such as entropy.

ID3, C45, and CART adopt a greedy (i.e., non-back tracking) approach in which decision trees are constructed in a top-down recursive divide-and-conquer manner (Jiawei & Micheline , 2006). Most algorithms for decision tree induction also follow such a top-down approach, which starts with a training set of tuples and their associated class labels. The training set is recursively partitioned into smaller subsets as the tree is being built. A basic decision tree algorithm is summarized below:

Algorithm: Generate decision tree. Generate a decision tree from the training tuples of data partition D.

Input:

a. Data partition, D, which is a set of training tuples and their associated class labels;
b. attribute list, the set of candidate attributes;
c. Attribute selection method, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of a splitting attribute and, possibly, either a split point or splitting subset.

Output: A decision tree.

Basic Steps:

1. create a node N;
2. if tuples in D are all of the same class, C then
3. return N as a leaf node labeled with the class C;
4. if attribute list is empty then
5. return N as a leaf node labeled with the majority class in D; // majority voting
6. apply Attribute selection method(D, attribute list) to find the "best" splitting criterion;
7. label node N with splitting criterion;
8. if splitting attribute is discrete-valued and multiway splits allowed then
   // not restricted to binary trees
9. attribute list attribute list; splitting attribute; // remove splitting attribute
10. for each outcome j of splitting criterion
    // partition the tuples and grow sub trees for each partition
11. let Dj be the set of data tuples in D satisfying outcome j; // a partition
12. if Dj is empty then
13. attach a leaf labeled with the majority class in D to node N;
14. else attach the node returned by Generate decision tree(Dj, attribute list) to node N;
    end for
15. return N;

2.7 Review of related work

A plethora of studies on predicting students' retention have been carried out by researchers. In the early nineties, research work became more oriented to building systems to analyze data, usually using techniques related to data mining. An example is the research by (Druzdzel & Glymour, 1994). The purpose was to find the possible causes of low freshman retention in USA colleges. The TETRAD II system was developed to inspect a database and find a set of causal structures using linear regression. The data source was a set of statistics from administrative data concerning 204 higher education institutions. Experiments showed that, of all variables considered, only test scores and class attendance are directly causally related to freshman retention. They were able to prove that the average test scores have high predictive power for student retention and graduation rates and for the 41 top ranking colleges, average test scores in combination with class attendance explains 68.3% of the variance in freshman retention rate and 77% of the variance in graduation rate. This research, therefore, associates the quality of students to attrition probabilities.

Research on student attrition using specific systems progressed with Sanjeev & Zytkow (1995). The Forty-Niner discovery system (an automated discovery system that explores databases in search for knowledge) was used to discover knowledge useful in understanding the university enrolment and to find ways to increase it. The system searched the databases for knowledge about enrolment, using a pattern discovery process developed by Zytkow and Zembowicz (1993) in form of regularities (statements in the form "Pattern P holds for data in range R"). Using demographic and academic information from the student database of Wichita State University (USA), from 1986 and 1987, researchers concluded that, although good high school students are the ones attending classes the most, some of these students drop out, causing significant enrolment losses.

The development of these systems led data mining techniques to begin to be widely applied in finding causes for student attrition. Research by Massa and Puliafito (1999) addresses directly the problem by defining clusters of students associated with different drop out risk degree. An approach based on Markov chains was used to identify clusters of people with a homogeneous behavior. The data set consisted of academic data from 15000 students of the University of Genoa (Italy) from 1978 to 1998. The main conclusion was that the proposed modelling technique can be used to define groups of students where attrition it most likely risk of happening, while not providing any explanation for it.

Náplava and Šnorek (2003) had the purpose of discovering whether and how a particular student would be successful in university studies using a method of modelling of the GMDH (Group Method of Data Handling) type neural network algorithms. Administrative data from freshman students of the Faculty of Electrical Engineering, Czech Technical University of Prague (Czech Republic) was used. This included data related to the student's personality, type and results (grades) from high school and entrance examination. Conclusions mainly helped to demystify some preconceived ideas so far. The first being that the type of high school is not a rather dominant factor for the success in the 1st semester. The second that the quality of the results is not directly related to the student's age. On a methodological perspective it proves that the students' results can be predicted by means of these methods with a high degree of accuracy.

Minaei-Bidgoli (2003) attempted an approach to classify students in order to predict their final grade based on features extracted from logged data in an education web based system. An ensemble of classifiers was designed, implemented and evaluated on an online programme dataset in the Michigan State University (USA), relating to 2002 data. Conclusions from a methodological viewpoint show that the combination of multiple classifiers led significant improvement in classification performance (as research in ensemble methods confirms) and

that this method may be considerably useful in identifying students at risk early, especially in very large classes, and allow the instructor to provide appropriate advising in a timely manner.

(Dekker *et al*., 2009), research aimed to predict the Electrical Engineering (EE) students drop out after the first semester of their studies and identify success-factors specific to that programme. Data was collected, over the period from 2000 to 2009 selecting 648 freshman students of Eindhoven University of Technology (Netherlands). Then comparisons were made between two decision tree algorithms, a Bayesian classifier, a logistic model, a rule-based learner and random forest. The OneR classifier was also considered as a baseline and as an indicator of the predictive power of particular attributes. Experimental results show that decision trees give a useful insights with accuracies between 75 and 80%. Conclusions delve on showing ways of further improvement in prediction without having to collect additional data about the students.

Kabakchieva (2012) wanted to reveal the high potential of data mining applications for university management in order to contribute to more efficient university enrolment campaigns and attracting the most desirable students. The research was focused on the development of data mining models for predicting student performance, based on their personal, pre-university and university-performance characteristics. The algorithms tried were OneR rule learner, a decision tree, neural network and k-nearest neighbour (k-NN) classifiers. The dataset used for the research includes data about students admitted to the University of National and World Economy of Sofia (Bulgaria) in three consecutive years. The main conclusions were that the highest accuracy is achieved for the neural network model, followed by the decision tree model and the k-NN model. The data attributes related to the students' university admission score and number of failures at the 1st year university exams are among the factors influencing most the classification process.

Bogard et al (2012) developed a first time, first year student retention model to understand the determinant factors. Three years of data for 1st time, 1st year degree-seeking students at Western Kentucky University (USA). SAS REnterprise Miner was used to implement an ensemble model consisting of logistic regression, decision trees, neural networks using the default average method. Each of the algorithms were also evaluated independently. After performance comparison, decision trees were preferred, based on reasoning. Risk factors identified were high school GPA and programme loads.

Kaur et al (2015) focused on identifying slow learners among students and displaying them by a predictive data mining model using classification based algorithms. Real world dataset was collected from a high school and filtration of desired potential variables was done using the WEKA workbench. They carried out experiments on the dataset of student academic records by applying classification algorithms such as Multilayer perceptron, Naïve Bayes, SMO, J48 decision tree, and using the WEKA (an open source tool). As a result, statistics were generated based on the above listed classification algorithms and comparison of all classifiers using the accuracy measure was also done. It is also reported in this work that Multilayer Perceptron classifier performs best among all.

From a purpose point of view, majority of researches conducted in this field attempts an approach at predicting the phenomenon of student attrition. Behind this goal, however, different motivations are revealed as it turns out to be a two dimensional problem. The first with emphasis in assessing the effectiveness of data mining techniques and the second with emphasis in finding the causes that can help explain why student attrition happens.

The focus of this dissertation was to build a first year at risk model for detecting, at an early stage students at risk of attrition.

CHAPTER THREE: PREDICTION MODELS DEVELOPMENT

In this chapter the prediction models were developed using selected classification algorithms. In the process, the following steps were discussed: data collection, data cleaning, data exploration, data normalization, data transformation, and data splitting.

3.1 Data Collection

Data collection is the process of gathering and measuring information on targeted variables in an established systematic fashion, which then enables one to answer relevant questions and evaluate outcomes. A formal data collection process is necessary as it ensures that data gathered are both defined and accurate and that subsequent decisions based on arguments embodied in findings through data are valid.

The primary focus of this work is to build a first year at risk model for students in A.B.U Zaria that will enable the university detect at an early stage the students at risk of attrition. Therefore data is collected from the university's information system. The data gathered emerged from two primary sources namely: exam processing software (exam-logic) and students' portal. Data sources used in this work were not centralized as this has made the data collection phase of this dissertation tedious. Regardless of the field of study or preference for defining data (quantitative, qualitative), accurate data collection is essential to maintaining the integrity of research; also the data collection phase of this work was carried out with utmost thoroughness to ensure nonexistence or high reduction in errors recorded.

In this phase, standard and notable data collection tools have been identified; the first is PHP MyAdmin which we used in extracting subset of data from the database; the second is Microsoft Excel which we used in carryout preliminary data analysis task.

3.1.1 PHPMyAdmin

phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. phpMyAdmin supports a wide range of operations on MySQL databases. It is frequently used for operations involving managing databases, tables, columns, relations, indexes, users and permissions. These operations can be carried out via the user interface, while you still have the ability to directly execute any SQL statement through the command line.

Specific features of phpMyAdmin used in this work as supported by MySQL database are:

1. Browse and drop databases, tables, views, fields and indexes;
2. Create, copy, drop, rename and alter databases, tables, fields and indexes;
3. Import and Export data from CSV and SQL;

In the data collection phase, SQL (Structured Query Language) pronounced SEQUEL has been very handy. SQL is accessible through the phpMyAdmin graphical user interface. SQL is used to create, transform and retrieve information from RDBMS (Relational Database Management Systems). SQL was developed during the early 70's at IBM and it is relatively simple but very powerful.

3.1.2 Microsoft Excel

Microsoft Excel is a spreadsheet application developed by Microsoft for Windows, Mac OS X, and iOS. It features calculation, graphing tools, pivot tables, and a macro programming language called Visual Basic for Applications. Some basic features of Microsoft Excel include basic operations, macro programming and charts. Microsoft Excel has the basic features of all spreadsheets, using a grid of cells arranged in numbered rows and letter-named columns to organize data manipulations like arithmetic operations. It has a set of supplied functions to answer statistical, engineering and financial needs. In addition, it can display data as line

graphs, histograms and charts, and with a very limited three-dimensional graphical display. It gives the data miner the power to do all sort of preliminary data analysis.

The Microsoft Excel spreadsheet has been used to view and organize dataset efficiently in this work. Microsoft Excel was used to rearrange data items and filter our data to match certain criteria. All of these options have enhanced data reorganization and preliminary data analysis during this work.

3.2 Data Cleaning

Real-world data tend to be incomplete, noisy, and inconsistent. Data cleaning (or data cleansing) routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data.

The primary operation carried out during data cleaning was ***discrepancy detection***. Discrepancies can be caused by several factors, including poorly designed data entry forms that have many optional fields, human error in data entry, deliberate errors (e.g., respondents not wanting to divulge information about themselves), and data decay (e.g., outdated addresses). Discrepancies may also arise from inconsistent data representations and the inconsistent use of codes.

Discrepancies was detected in this work by examining previous knowledge already obtained regarding properties of the data. Such knowledge or "data about data" is referred to as metadata. For instance:

1. What is the domain and data type of each attribute?

2. What is the acceptable values for each attribute?

3. What is the range of the length of values?

4. Do all values fall within the expected range?

5. Are there any known dependencies between attributes?

In this step, Python programming language (pandas library) scripts were written to find noise, outliers, and unusual values in our dataset. As more information about data was unveil during cleaning, this was used for updating the metadata to reflect this knowledge as that has helped in speeding up data cleaning on future versions of the same data store.

3.2.1 Python Programming Language

Python is a high-level, object oriented, prototyping language. Python is a scripting language which means it is interpreted every time it is run. Some advantages of Python over other (scripting) languages are that it is extensible and embeddable in applications. Python is free, open source, easy to learn and well readable.

Python has emerged over the last couple decades as a first-class tool for scientific computing tasks, including the analysis and visualization of large datasets. This may have come as a surprise to early proponents of the Python language: the language itself was not specifically designed with data analysis or scientific computing in mind. The usefulness of Python for data mining stems primarily from the large and active ecosystem of third-party packages particularly NumPy for manipulation of homogeneous array-based data, Pandas for manipulation of heterogeneous and labeled data, SciPy for common scientific computing tasks, Matplotlib for high-quality visualizations, IPython for interactive execution and sharing of code, Scikit-Learn for machine learning, and many more tools.

3.2.2 IPython

The IPython project began in 2001 as Fernando Pérez's side project to make a better interactive Python interpreter. In the subsequent 15 years it has grown into what's widely considered one of the most important tools in the modern scientific Python computing stack (McKinney, 2013).

The IPython project now encompasses a great deal more than just an enhanced, interactive Python shell. It also includes a rich GUI console with inline plotting, a web-based interactive notebook format, and a lightweight, fast parallel computing engine. And, as with so many other tools designed for and by programmers, it is highly customizable.

In this work, IPython Notebook has been used to edit, run and comment codes in python PL directly from a web browser and this has assisted in speeding up data manipulation, re-organization and transformation task carried out in this work.

3.2.3 Pandas

Pandas is an open source Python-based data analysis library. It provides high-performance and easy-to-use data structures and data analysis tools built with the Python programming language. The Pandas library brings many of the good things from R, specifically the Data Frame objects and R packages such as plyr and reshape2, and places them in a single library that you can use in your Python applications (Heydt, 2015). Some library features of Python pandas we have exploited in this work are:

1. High performance array and table structures for representation of homogenous and heterogeneous data sets: the Series and Data Frame objects
2. Flexible reshaping of data structure, allowing the ability to insert and delete both rows and columns of tabular data
3. Powerful grouping capabilities and a functionality to perform split-apply-combine operations on series and tabular data
4. Sub setting of data sets, including intuitive, flexible merging, and joining of data with SQL-like constructs
5. Extensive I/O facilities to load and save data from multiple formats including CSV, Excel, relational and non-relational databases, HDF5 format, and JSON

A popular means of using pandas is through the use of IPython Notebooks. IPython Notebooks provide a web-based interactive computational environment, allowing the combination of code, text, mathematics, plots, and right media into a web-based document. IPython Notebooks run in a browser and contain Python code that is run in a local or server-side Python session that the notebooks communicate with using WebSockets. Notebooks can also contain markup code and rich media content, and can be converted to other formats such as PDF, HTML, and slide shows. All python scripts written in this work are done in the web browser interface of IPython Notebook

3.3 Data Exploration

Data exploration is the first step in data analysis and typically involves summarizing the main characteristics of a dataset. It is commonly conducted using visual analytics tools, but can also be done in a spreadsheet software, such as Microsoft Excel. The data visualization feature of Microsoft Excel has been used for data exploration because it enabled us quickly and simply view most of the relevant features in the dataset collected. From this step, variables having interesting observations were identified. By displaying data graphically; for example, through scatter plots or bar charts we ascertained variables that correlate and determine if they are good candidates for further in-depth analysis.

In this work, taking a close look at the dataset collected of selected departments in A.B.U Zaria, there are 12 attributes in all, and a very noteworthy attribute is the class attribute (categorical attribute with values: RETAINED or DROPPED); which aided in a principal data exploration task carried out in this work as shown in Figure 1.1.

Table 3.1: Attributes selected from dataset

S/N	varname	vartype
1	Gender	categorical
2	age	continous
3	State	categorical
4	bloodgroup	categorical
5	marital status	categorical
6	residence	categorical
7	aggregated olevel result	continous
8	ume/putme score	continous
9	current cgpa	continous
10	religion	categorical
11	entrymode	categorical
12	genotype	categorical

3.4 Data Integration.

Data mining often requires data integration: the merging of data from multiple data sources. The data may also need transformation into forms appropriate for mining. This section describes both data integration and data transformation performed on the dataset collected for the purpose of this research.

During our data gathering phase, data integration was carried out which combines data from multiple sources into a coherent data store. These sources include the database of the students' portal and that of the exam processing software (exam-logic). There are a number of issues to consider during data integration such as schema integration, object matching, and redundancy. How can equivalent real-world entities from different data sources be matched up? (This is referred to as the entity identification problem). For example, how can we be sure that student id in one database and student id number in another refer to the same attribute? The metadata

of each attribute including the name, meaning, data type, and range of values permitted for the attribute, and null rules was considered for overcoming this. Metadata can be very helpful as it used to avoid errors in schema integration. The metadata may also be used to help transform the data (e.g., where data codes for grade in one database may be "A" and "1", in another). Hence, this step also relates to data cleaning, as described earlier in section 3.2.

Redundancy is another important issue. An attribute (such as student cgpa) may be redundant if it can be "derived" from another attribute or set of attributes. In this work, the case of redundancy was handled by employing derived attributes and this assisted in taking out missing values on attributes that can be derived from others.

One important issue in data integration is the detection and resolution of data value conflicts. For example, for the same real-world entity, attribute values from different sources may differ. This may be due to differences in representation, scaling, or encoding. For instance, an attribute in one system may be recorded at a lower level of abstraction than the "same" attribute in another. Therefore, when matching attributes from one database to another during integration, special attention must be paid to the structure of the data. This is to ensure that any attribute's functional dependence and referential constraint in the source system match those in the target system.

3.5 Data Transformation

This dissertation cores on classification and the objective of classification is to analyze the input data and to develop a model for each class using the features present in the data; as a result semantic heterogeneity of features in data pose great challenges in any classification task. A careful transformation of heterogeneous features to homogenous can help reduce classification errors in the resulting data set. This can help improve the accuracy and speed of the subsequent mining process.

In this stage of the work, the data got transformed or consolidated into forms appropriate for mining. Data transformation can involve the following:

1. Normalization of attribute scales values so as to fall within a small specified range, such as - 1:0 to 1:0, or 0:0 to 1:0. In this work the status of a student's instance was defined to be either of **RETAINED** / **DROPPED** and this is determined using *smoothing* technique. The final cumulative grade point (cgpa) of the student was used to determine if the student will be retained or dropped. This is done by tagging cgpa < 0 as dropped and otherwise as retained.
2. Aggregation, where summary or aggregation operations are applied to the data. For example, the total credit unit of courses offered by a student is computed. This step is typically used in constructing a data cube for analysis of the data at multiple granularities.
3. Generalization of the data, where low-level or "primitive" (raw) data are replaced by higher-level concepts through the use of concept hierarchies. For instance values for numerical attributes, like age, may be mapped to higher-level concepts, like youth, middle-aged, and senior. In this dissertation work categorical attributes, like a student's local government area is generalized to higher-level concepts, like state of origin. Similarly, we mapped values for student entry level like 100, 200 and 300 level to freshman, direct entry and special admission respectively.
4. Attribute construction (or feature construction), where new attributes are constructed and added from the given set of attributes to help the mining process. In this work, newer attributes were constructed from existing ones; an example is deriving age at entry from date of entry and date of birth of the student; another example is deriving the value cumulative grade point from attributes like credit unit and grade point of a particular course the student offered.

3.5.1 Numerical Coding of nominal attributes

For classification of data, it is extremely important that the nature of data is known which is either numeric, continuous or mixed data type, this information is essential because different algorithms and techniques are more suitable for different types of data. Many distance learning algorithms such as k nearest neighbors learning algorithms, locally weighted learning algorithm e.tc depends upon a good distance metric to be successful. In this kind of algorithms a key problem is how to measure distance between each pair of instance.

In this work, a technique for converting nominal attributes to numeric was proposed for easy computation of similarity measure between two instances using standard distance measures like Euclidean distance and Manhattan distance. A form of rank size distribution in transforming nominal attributes to numeric was employed. For any given nominal attribute in the dataset we have also considered the frequency of identical values to influence our ranking decision.

A rank in statistics is a relationship between set of data values such that, for any two items the first is either 'ranked higher than', 'ranked lower', 'ranked equal' to the second. Nominal data cannot be sorted according to identical values but can be grouped according to identical values.

For any nominal attribute identical data values have been grouped and assigned numbers from 1 to n as positions. A unique rank is computed for each group of identical values that is equal to the average of the positions of this data values. Table 3.2 and Table 3.3 shows how the transformation was accomplished using a chosen nominal attribute and converting to numeric using a variation of rank frequency distribution

Table 3.2: An example of numeric and nominal attribute in our dataset

age	bloodgroup
21	A
23	A

41	O
19	O
26	O
17	O
21	AB
19	AB
22	AB

We are tasked with assigning a rank to the data values in the blood group attribute, as this will simplify the computation of Euclidean distance metric in our chosen nearest neighbor instance learning algorithm.

Table 3.3: Rank of nominal data – difference in the cardinality of data values

Value	Position	Rank	Value	Position	Rank	Value	Position	Rank
A	1	1.5	O	1	2.5	AB	1	2.0
A	2	1.5	O	2	2.5	AB	2	2.0
			O	3	2.5	AB	3	2.0
			O	4	2.5			

Table 3.2 clearly express that to each data value in our bloodgroup column, respective rank were assigned, also showing that more identical values will have higher rank than others with less. Also in this work we have provided a pseudo code for accomplishing the modified rank frequency distribution expressed in this section.

3.5.2 Algorithm Pseudo code for converting Nominal to numeric values using rank frequency distribution

```
#Input: Array[1,n]; Output: Array[1,n]
A ← {}
B ← Unique(A)
C ← {}
For i ← 1 to len(B)
        sum ← 0
        average ← 0
        position ← 0
        for j ← 1 to len(A)
                if(B[i] = A[j])
                        position ← position + 1
                        sum ← sum + position
        average ← sum / position
        C ← (B[i], average)
For (key, val) in C
        for k ← 1 to len(A)
                if(A[k] = key)
                        A[k] ← val
return A[k]
```

At this point, it is concluded that for any given nominal attribute in the dataset, a rank can be assigned in form of a numeric value.

3.6 Data Splitting

Data partitioning also known as data splitting is an approach used to divide the entire dataset into subsets of data. In an ideal situation, more independent data sets can be collected or simply and inexpensively repeat an experiment to collect new ones. Also independent datasets can be used for learning, model selection and even an assessment of the prediction performance. In such situations there exist no reason to split any dataset. But in situations when only one dataset is available and newer ones cannot be collected then there is need for some strategy to perform prediction task effectively based on the dataset collected. In this section, several data splitting strategies were reviewed and predominantly the chosen strategy for this work.

The dataset for building our model in a classification task can be divided into subsets of train and test data.

Machine Learning deals with programs learning from data sets and as a result the ***training data*** is the data on which the machine learning programs learn to perform correlational tasks (classify, cluster, learn the attributes) while the ***testing data*** is the data, whose outcome is already known and is used to determine the accuracy of the machine learning algorithm, based on the training data (how effectively the learning happened). Figure 3.1 shows how a single data set is divided into instances of train and test set as given to the machine learning algorithms.

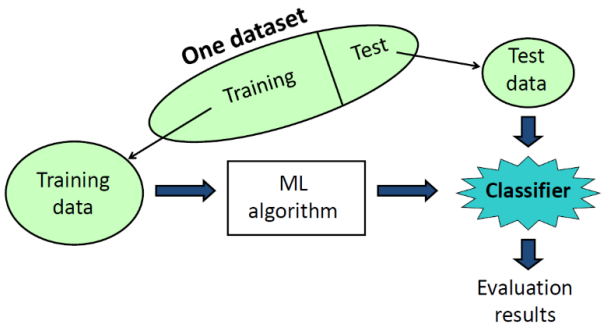


Figure 3.1: Data partition (Training and test set).

3.7 Model validation

Model Validation is the process of assessing how well a data mining model performed against real data. It primarily applies the model to test data in order to determine whether the model, built on a training set, is generalizable to other data. In particular, it helps to avoid the phenomenon of over fitting, which can occur when the logic of the model fits the build data too well and therefore has little predictive power.

3.7.1 Cross validation

Cross-validation is a technique to evaluate predictive models by partitioning the original sample into a training set to fit the model, and a test set to evaluate it. Cross validation is mostly used when a limited amount of data is available, in order to achieve an unbiased estimate of the model performance.

3.7.2 Holdout method

The holdout method is the simplest kind of cross validation. The data set is separated into two sets, called the training set and the testing set. The machine learning algorithm fits a classifier using the training set only. Then the classifier is asked to predict the output values for the data in the testing set (it has never seen these output values before). The errors it makes are accumulated as before to give the mean absolute test set error, which is used to evaluate the model. The advantage of this method is that it is usually preferable to the residual method and takes no longer to compute. However, its evaluation can have a high variance. The evaluation may depend heavily on which data points end up in the training set and which end up in the test set, and thus the evaluation may be significantly different depending on how the division is made.

3.7.3 K-fold cross validation

K-fold cross validation is one way to improve over the holdout method. The data set is divided into k subsets, and the holdout method is repeated k times. Each time, one of the k subsets is used as the test set and the other k-1 subsets are put together to form a training set. Then the average error across all k trials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set k-1 times. The disadvantage of this method is that the training algorithm has to be rerun from scratch k times, which means it takes k times as much computation to make an evaluation.

Given a set of m training examples, a single run of k-fold cross validation proceeds as follows:

1. Arrange the training examples in a random order.
2. Divide the training examples into k folds. (k chunks of approximately m=k examples each.)
3. For i = 1,2,3,..,k:
    a. Train the classier using all the examples that do not belong to fold i.
    b. Test the classier on all the examples in fold i.



Figure 3.2: 10 fold cross validation

3.7.4 Leave-one-out cross validation

Leave-one-out cross validation is K-fold cross validation taken to its logical extreme, with K equal to N, the number of data points in the set. That means that N separate times, the classifier is trained on all the data except for one point and a prediction is made for that point. The evaluation given by leave-one-out cross validation error (LOO-XVE) is good, but at first pass it seems very expensive to compute.

However, dataset used in this work is partitioned into two (train and test set) using the k-fold cross validation (where k = 10) this technique was applied to all machine learning algorithms used for classification purpose; Ninety percent (90%) of the data was chosen to be the training set, and 10% to be the testing set in each iteration as shown in Figure 3.2

3.8 Attribute Weighting

It has been frequently observed that the classification of data depends more upon some attributes than others. Finding those relevant attributes is called attribute selection and making a method in which those relevant attributes are given more value is called attribute weighting.

In recent years, instance based learning methods, have emerged as a promising approach to machine learning, with researchers reporting excellent results on many real world induction task (David, 1992). The basic approach involves storing training cases and their associated class in memory and then when given a test instance, finding the training cases nearest to the test instance and using them to predict the class. One drawback of such method is that when combined with a naïve distance metric that weight attributes equally, they can suffer from the curse of dimensionality in which the number of attributes needed to maintain a given error rate grows rapidly with the number of features.

This observation has led some researchers to augment nearest neighbors methods with techniques for determining distinct weight for each feature. For instance (Cost & Salzberg, 1993) uses difference in probability distribution across classes to modify the distance metric for nominal attributes, whereas Dietrich (1994) uses mutual information to compute coefficient on numeric attributes. All of these researches reports improvement over the simple version of nearest neighbor that gives attributes equal weight.

The usage of feature selection and ranking of attributes in classification problems have shown very good results in the recent past; this concept has taken its power from information theory. The purpose of these techniques is to discard irrelevant or redundant features from a given feature vector. The following attribute evaluations are used: information gain (IG), gain ratio, symmetrical uncertainty, relief-F, one-R and chi-squared.

In this work a univariate measure of attribute weighting has been considered. Univariate means that the measurement is carried out for a single variable. The practical usefulness of IG attribute evaluation was considered by applying it as weight of attributes in the classification problem. The effect of using IG as the basis of weights assigned to features in the dataset used are evaluated using the naïve k – nearest neighbor algorithm.

3.9 Feature Selection

Dataset can have a very large number of features that can lead to the need of reducing the dimensionality. The term dimensionality reduction is often reserved for those techniques that reduce the dimensionality of a data set by creating new attributes that are a combination of the old attributes.

However, the reduction of dimensionality by selecting new attributes that are a subset of the old is known as feature subset selection or feature selection. Another way to reduce the dimensionality is to use only a subset of the features (Pang-Ning *et al*., 2009). While it might seem that such an approach would lose information, this is not the case if redundant and irrelevant features are present. Redundant features duplicate much or all of the information contained in one or more other attributes. Irrelevant features contain almost no useful information for the data mining task at hand. For instance, students' ID numbers are irrelevant to the task of predicting students' grade point averages.

Redundant and irrelevant features can reduce classification accuracy and the quality of the clusters that are found. While some irrelevant and redundant attributes can be eliminated immediately by using common sense or domain knowledge, selecting the best subset of features frequently requires a systematic approach. The ideal approach to feature selection is to try all possible subsets of features as input to the data mining algorithm of interest, and then take the subset that produces the best results. This method has the advantage of reflecting the

objective and bias of the data mining algorithm that will eventually be used. There are three standard approaches to feature selection namely: embedded, filter, and wrapper.

3.9.1 Embedded approaches

Feature selection occurs naturally as part of the data mining algorithm. Specifically, during the operation of the data mining algorithm, the algorithm itself decides which attributes to use and which to ignore. Algorithms for building decision tree classifiers, often operate in this manner.

3.9.2 Filter approaches

Features are selected before the data mining algorithm is run, using some approach that is independent of the data mining task.

3.9.3 Wrapper approaches

These methods use the target data mining algorithm as a black box to find the best subset of attributes. It is also possible to encompass both the filter and wrapper approaches within a common architecture. The feature selection process is viewed as consisting of four parts: a measure for evaluating a subset, a search strategy that controls the generation of a new subset of features, a stopping criterion, and a validation procedure. Filter methods and wrapper methods differ only in the way in which they evaluate a subset of features. For a wrapper method, subset evaluation uses the target data mining algorithm, while for a filter approach, the evaluation technique is distinct from the target data mining algorithm.

Conceptually, feature subset selection is a search over all possible subsets of features. Many different types of search strategies can be used, but the search strategy should be computationally inexpensive and should find optimal or near optimal sets of features. lt is usually not possible to satisfy both requirements, and thus, tradeoffs are necessary.

3.9.4 Information Gain Attribute Selection

In this work, filter method of feature selection and ranker search technique were used to assign ranks to eleven (11) attributes retrieved after carrying out data preprocessing task. The

information gain (IG) technique of feature selection has been chosen since it evaluates the worth of an attribute by measuring the information gained with respect to the class attribute; Information gain is a very effective metric in selecting features and can be defined as a measure of the effectiveness of an attribute in classifying the training data. If we split training data on attribute values, then information gain gives the measurement of the expected reduction in entropy after the split. The more an attribute can reduce entropy in the training data, the better the attribute in classifying the data. Information Gain of a binary attribute A on a collection of samples S is given by Equ 3.1:

$$Gain(S, A) \equiv Entropy\ (S) - \sum_{v \in Values(A)} \frac{|S_v|}{S} Entropy(S_v) \quad \text{Equ 3.1}$$

where Values(A) is the set of all possible values for attribute A, and Sv is the subset of S for which attribute A has value v. In our case, each binary attribute has only two possible values (Retained, Dropped). Entropy of subset S is computed using the following equation:

$$Entropy\ (S) = -\frac{p(s)}{n(s) + p(s)} log_2 \frac{p(s)}{n(s) + p(s)} - \frac{n(s)}{n(s) + p(s)} log_2 \frac{n(s)}{n(s) + p(s)} \quad \text{Equ 3.2}$$

Where p(S) is the number of positive samples in S and n(S) is the total number of negative samples in S. Computation of information gain of a continuous attribute is a little tricky, because it has infinite number of possible values. One approach followed by (Quinlan, 1993) is to find an optimal threshold, and split the data into two halves. The optimal threshold is found by searching a threshold value with highest information gain within the range of values of this attribute in the dataset.

In this work attributes have been ranked in line to their information gain using WEKA's ranker search method as shown in Table 3.3.

Table 3.4: Attribute selection using IG and Ranker Search Method

Attribute Name	Attribute Description	Info Gain	Rank
semester2	**cgpa (first year, second semester cgpa)**	**0.181**	**1**
semester1	**cgpa (first year, first semester cgpa)**	**0.148**	**2**
state	**state of origin**	**0.01**	**3**
stayinhostel	**Student staying incampus or offcampus**	**0.009**	**4**
age	**age of student at entry or admission**	**0.0055**	**5**
entrymode	**entry level of student (100,200,300)**	**0.004**	**6**
gender	**Male/Female**	**0.003078**	**7**
genotype	AB, O, B, AB	0.000851	8
blood group	AA, AS, SS	0.000554	9
country	nationality – Nigerian / Foreigner	0.000133	10
residence	resident in Nigeria or not at entry	0.000111	11

3.10 Proposed Algorithm for k Nearest Neighbor using Information as Weight

A primary contribution of this work to knowledge is to enhance k nearest neighbor technique by transforming nominal attributes to continuous using ranks and weighting attributes using univariate measure of information gain with respect to the class.

Below is our proposed algorithm:

1. Given a dataset X containing *n* instances and attributes; X = x_{11}, x_{22}, x_{33},...,x_{mn}
2. **Convert attributes to numeric (if nominal) using a Modified Numerical Coding of Nominal Attributes**
3. **Compute attribute weight (*Wxi*), using Information gain of attribute *xi* as *W***
4. Partition X into a training set D = ($\boldsymbol{x}$, y) and a test set z = ($\boldsymbol{x}^i$, y^i)
5. Compute $d(\boldsymbol{x}^i,\boldsymbol{x})$, i.e the distance between z and every object ($\boldsymbol{x}$, y) ∈ D

6. Select $D_z \subset D$, the set of k closest training objects to *z*
7. Return the set of objects in D_z with majority class votes

3.11 Tools and Techniques used

In this dissertation work, classification techniques used for prediction of students at risk of attrition were employed. Also prediction models were developed during this phase. Dataset are tested and analyzed with six (6) classification algorithms namely:

1. Multi-Layer Perceptron
2. Naïve Bayes
3. J48 Decision Tree
4. Sequential minimal optimization
5. K-Nearest Neighbor
6. Modified K-Nearest Neighbor

For implementation of all these classification algorithms. WEKA (Waikato Environment for Knowledge Analysis) and Python Scikit Learn Library were used.

3.11.1 WEKA

The WEKA the pronounced to rhyme with Mecca is an acronym for **Waikato Environment for Knowledge Analysis**; the WEKA workbench is a collection of state-of-the-art machine learning algorithms and data preprocessing tools (Witten & Frank, 2005). It includes virtually all the algorithms described in this work. It is designed so that you can apply existing methods on new datasets in flexible ways. It provides extensive support for the whole process of experimental data mining, including preparing the input data, evaluating learning schemes statistically, and visualizing the input data and the result of learning. As well as a wide variety of learning algorithms, it includes a wide range of preprocessing tools. This diverse and

comprehensive toolkit is accessed through a common interface so that its users can compare different methods and identify those that are most appropriate for the problem at hand.

The easiest way to use WEKA is through a graphical user interface called the Explorer. This gives access to all of its facilities using menu selection and form filling. Behind these interactive interfaces lies the basic functionality of WEKA. It comprises of four different user interfaces: the Explorer, the Knowledge Flow, the Experimenter, and the command-line interface.

3.11.2 Scikit learn Python Library

Scikit-learn is a free machine learning library for the Python programming language. It features various classification, regression and clustering algorithms. Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

The library is built upon the SciPy (Scientific Python) that must be installed before you can use scikit-learn. This stack that includes:

1. NumPy: Base n-dimensional array package
2. SciPy: Fundamental library for scientific computing
3. Matplotlib: Comprehensive 2D/3D plotting
4. IPython: Enhanced interactive console
5. Sympy: Symbolic mathematics
6. Pandas: Data structures and analysis

The vision for the library is a level of robustness and support required for use in production systems. This means a deep focus on concerns such as ease of use, code quality, collaboration, documentation and performance in machine learning.

3.12 Model Evaluation

Evaluation is the key to making real progress in data mining (Witten & Frank, 2005). Comparing the performance of different machine learning methods on a given problem is another matter that is not as easy as it sounds. In data mining, performance on training set is definitely not a good indicator of performance on test set. Since data preprocessing has been carried out; model that perfectly fit only the training data is not desirable, but instead a model that is reliable even with test data is needed. Therefore, model evaluation is a key part of this work. Two phases of a model evaluation were applied in this work. In the first phase model evaluation was performed in order to estimate the parameters of the model during the learning phase. This is a part of the model selection where a model with the best results is chosen. This phase is also called the validation phase. It does not necessarily that a model that best fits a particular set of data is selected. The well learned model captures only the underlying phenomenon, not the noise. A model that captures a noise is called an over-fitted model. In the second phase, the selected model is evaluated in order to assess the real performance on new unseen data. The process steps are model selection, model learning (training phase), model validation (validation phase), model assessment (testing phase).

3.13 Evaluation measures

In the field of data mining there is a large variation of performance measures for evaluating a classifier or predictor. However, a measure good for evaluating a model in a particular domain could be inappropriate in another domain and vice versa. The choice of an evaluation measure depends on the domain of use and the given problem. Moreover, different measures are used for classification problems. In this work, evaluation metrics are selected (accuracy, precision and recall measures) as used in the work of (Parneer *et al.*, 2015).

3.13.1 Measures for classification evaluation

The basis for analyzing classifier performance is a confusion matrix (Witten & Frank, 2005). The confusion matrix describes how well a classifier can recognize different classes. A confusion matrix is an n×n table, in which the (i,j)th entry indicates the count of instances of the class i classified as j. It means that correctly classified instances are on the main diagonal of the confusion matrix. The simplest and the most common form of the confusion matrix is a two class matrix as it is shown in the Figure 3.3 Given two classes, Positive and Negative refer to the classes. True Positives are positive instances that were correctly classified, True Negatives are also correctly classified instances but of the negative class. On the contrary, False Positives are incorrectly classified positive instances and False Negatives are incorrectly classified negative instances. TP (True Positives) TN (True Negatives) FP (False Positives) FN (False Negatives).

		Predicted class	
		yes	no
Actual class	yes	true positive	false negative
	no	false positive	true negative

Figure 3.3: Confusion matrix

The first and the most commonly used measure is the accuracy denoted as Acc(X). The accuracy of a classifier on a given set is the percentage of correctly classified instances. We can define the accuracy as:

$$\textbf{Acc(X)} = \frac{correctly-classified}{all\ instances} \qquad \text{Equ 3.3}$$

Or in two class problem as:

$$\textbf{Acc(X)} = \frac{TP+TN}{TP+TN+FP+FN} \qquad \text{Equ 3.4}$$

Since accuracy is defined, we can define the **error rate** (which is the percentage of incorrectly classified instances) of a classifier as:

Err(X) = 1 - **Acc(X)**. Equ 3.5

Once you have a model that you believe can make robust predictions you need to decide whether it is a good enough model to solve your problem. Accuracy alone is typically not enough information to make this decision (Brownlee, 2014). Alternatives for the accuracy measure are:

Sensitivity (also called *True Positive Rate* or *Recall*) - the percentage of truly positive instances that were classified as positive.

Sensitivity(X) = $\frac{TP}{TP+FN}$ Equ 3.6

Specificity (also called *True Negative Rate*) - the percentage of truly negative instances that were classified as negative.

Specificity (X) = $\frac{TN}{TN+FP}$ Equ 3.7

Precision is the percentage of positively classified instances that are truly positive.

Precision (X) = $\frac{TP}{TP+FP}$; Equ 3.8

Recall is the percentage of positively classified instances that are recalled as positive.

Recall (X) = $\frac{TP}{TP+FN}$ Equ 3.9

It can be shown that the accuracy is a function of the sensitivity and specificity as given below:

Acc(X) = Sensitivity(X) = $\frac{TP}{TP+FN}$ + **Sensitivity(X)** = $\frac{TN}{TN+FP}$ Equ 3.10

Finally is the Harmonic mean which is appropriate for situations when the average of rates (a ratio between two related quantities) is desired. Harmonic mean also referred to as F-measure and is given as:

F-measure = $N/(1/a_1+1/a_2+1/a_3+1/a_4+.....+1/a_N)$. Equ 3.11

3.14 System Architecture

This section shall give a pictorial presentation of the architecture of the intended system. The entire architecture can be decomposed into three major components. These components are; data pre-processing, data mining and results validation. Figure 3.4 lays out the architecture of this system.

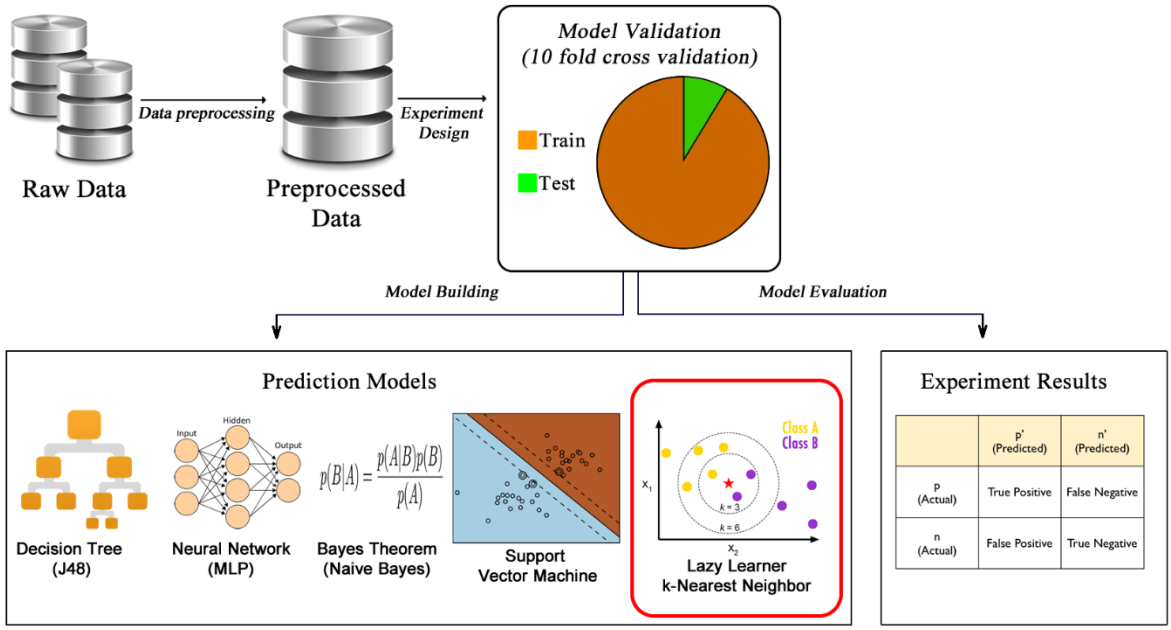


Figure 3.4: System Architecture

CHAPTER FOUR: RESULTS AND DISCUSSIONS

In this chapter, the results of experiments carried out in the model build up process were discussed; also shown is the experiments involving cross validating our prepared dataset with selected classification algorithms. This chapter also shows statistics of results obtained after experiments using tables, charts and the knowledge flow in WEKA.

4.1 Data Set characteristics

Before describing the experimental results in this section, a description of how the data is partitioned into six (6) separate sets is shown. First of all, the dataset was partitioned based on departments. The selection is driven by the availability of data in which the following departments are chosen: Department of Mathematics, Faculty of Pharmacy, Department of Building Engineering, Department of Mass Communication, Faculty of Law and Faculty of Agricultural Science all in A.B.U Zaria.

4.2 Experiments

The next step was to conduct separate experiments for the partitioned dataset using the 10 fold cross validation in WEKA and for this purpose, the following classification algorithms were applied: Multilayer perception, Naïve Bayes, J48, SMO, kNN and modified kNN.

Table 4.1: Experiment Mapping to Faculties and Departments

Experiments	**Faculty/Department Name**	**No of Instances**
Experiment 1	Department of Mathematics	1975
Experiment 2	Department of Building Engineering	561
Experiment 3	Faculty of Agricultural Science	477
Experiment 4	Department of Mass Communication	1053
Experiment 5	Faculty of Pharmacy	853
Experiment 6	Faculty of Law	1560

4.2.1 Experiment 1 (Department of Mathematics)

In this experiment, training and testing of six (6) classifiers were conducted separately using Multilayer perception, Naïve Bayes, J48, SMO, k Nearest Neighbor and modified k Nearest Neighbor algorithms; The 10 fold cross validation is applied, allowing all instances to participate in both the training and testing process.

Table 4.2 shows all statistics based on the evaluation measures we have selected (accuracy, precision, recall and f-measure);

Table 4.2: Statistical Analysis of Classifiers with Cross Validation (Mathematics)

Algorithm	Class	TP Rate	FP Rate	Precision	Recall	F-measure	Accuracy
KNN	Retained	0.963	0.494	0.953	0.963	0.958	92.3%
	Dropped	0.506	0.037	0.565	0.506	0.534	
mKNN	Retained	0.981	0.424	0.96	0.981	0.971	94.5%
	Dropped	0.576	0.019	0.744	0.576	0.649	
SMO	Retained	0.989	0.477	0.956	0.989	0.972	94.8%
	Dropped	0.523	0.011	0.818	0.523	0.638	
J48	Retained	0.983	0.355	0.967	0.983	0.975	95.3%
	Dropped	0.645	0.017	0.787	0.645	0.709	
Naïve Bayes	Retained	0.934	0.163	0.984	0.934	0.958	92.5%
	Dropped	0.837	0.066	0.548	0.837	0.662	
Multilayer Perceptron	Retained	0.968	0.39	0.963	0.968	0.966	93.7%
	Dropped	0.61	0.032	0.648	0.61	0.629	

A comparison of accuracy of all classifiers used have been carried out and finally it has been investigated that the J48 decision tree algorithm performed best with an accuracy of 95.3%.

4.2.2 Experiment 2 (Department of Building)

In this experiment, training and testing of six (6) classifiers were conducted separately using Multilayer perception, Naïve Bayes, J48, SMO, k Nearest Neighbor and modified k Nearest Neighbor algorithms; The 10 fold cross validation is also applied, allowing all instances to participate in both the training and testing process. Table 4.3 shows all statistics based on the evaluation measures we have selected (accuracy, precision, recall and f-measure);

Table 4.3 Statistical Analysis of Classifiers with Cross Validation (Building)

Algorithm	Class	TP Rate	FP Rate	Precision	Recall	F-measure	Accuracy
KNN	Retained	0.981	0.5	0.976	0.981	0.979	95.9%
	Dropped	0.5	0.019	0.565	0.5	0.531	
nKNN	Retained	1	0.4	0.978	1	0.989	97.9%
	Dropped	0.538	0	1	0.538	0.7	
SMO	Retained	0.998	1	0.954	0.998	0.975	95.1%
	Dropped	0	0.002	0	0	0	
J48	Retained	0.998	0.231	0.989	0.998	0.993	98.7%
	Dropped	0.769	0.002	0.952	0.769	0.851	
Naïve Bayes	Retained	0.929	0.115	0.994	0.929	0.960	92.6%
	Dropped	0.885	0.071	0.377	0.885	0.529	
Multilayer Perceptron	Retained	0.989	0.269	0.987	0.989	0.988	97.6%
	Dropped	0.731	0.011	0.76	0.731	0.745	

A comparison of accuracy of all classifiers used have been carried out and finally it has been investigated that the J48 decision tree algorithm performed best with an accuracy of 98.7%.

4.2.3 Experiment 3 (Faculty of Agricultural Science)

In this experiment, training and testing of six (6) classifiers were conducted separately using Multilayer perception, Naïve Bayes, J48, SMO, k Nearest Neighbor and modified k Nearest Neighbor algorithms; The 10 fold cross validation is also applied, allowing all instances to participate in both the training and testing process. Table 4.4 shows all statistics based on the evaluation measures we have selected (accuracy, precision, recall and f-measure);

Table 4.4: Statistical Analysis of Classifiers with Cross Validation (*Agric Science*)

Algorithm	Class	TP Rate	FP Rate	Precision	Recall	F-measure	Accuracy
KNN	Retained	0.985	0.647	0.976	0.985	0.980	96.2%
	Dropped	0.353	0.015	0.462	0.353	0.4	
mKNN	0.998	0.765	0.972	0.998	0.985	0.998	97.05%
	0.235	0.002	0.8	0.235	0.364	0.235	
SMO	Retained	1	0.824	0.97	1	0.985	97.0%
	Dropped	0.176	0	1	0.176	0.3	
J48	Retained	0.998	0.231	0.989	0.998	0.993	98.7%
	Dropped	0.769	0.002	0.952	0.769	0.851	
Naïve	Retained	0.967	0.294	0.989	0.967	0.978	95.7%
Bayes	Dropped	0.706	0.033	0.444	0.706	0.545	
Multilayer	Retained	0.991	0.412	0.985	0.991	0.988	97.6%
Perceptron	Dropped	0.588	0.009	0.714	0.588	0.645	

A comparison of accuracy of all classifiers used have been carried out and finally it has been investigated that the J48 decision tree algorithm performed best with an accuracy of 98.7%.

4.2.4 Experiment 4 (Department of Mass Communication)

In this experiment, training and testing of six (6) classifiers were conducted separately using Multilayer perception, Naïve Bayes, J48, SMO, k Nearest Neighbor and modified k Nearest Neighbor algorithms; The 10 fold cross validation is also applied, allowing all instances to participate in both the training and testing process. Table 4.5 shows all statistics based on the evaluation measures we have selected (accuracy, precision, recall and f-measure);

Table 4.5: Statistical Analysis of Classifiers with Cross Validation (*Mass Comm*)

Algorithm	**Class**	**TP**	**FP**	**Precisio**	**Recall**	**F-**	**Accuracy**
KNN	Retained	0.985	0.596	0.97	0.985	0.977	95.6%
	Dropped	0.404	0.015	0.583	0.404	0.477	
mKNN	0.987	0.538	0.972	0.987	0.98	0.987	96.1%
	0.462	0.013	0.649	0.462	0.539	0.462	
SMO	Retained	0.996	0.596	0.97	0.996	0.983	96.6%
	Dropped	0.404	0.004	0.84	0.404	0.545	
J48	Retained	0.981	0.096	0.995	0.981	0.988	97.7%
	Dropped	0.904	0.019	0.712	0.904	0.797	
Naïve	Retained	0.943	0.058	0.997	0.943	0.969	94.3%
Bayes	Dropped	0.942	0.057	0.462	0.942	0.62	
Multilayer	Retained	0.983	0.462	0.976	0.983	0.98	96.1%
Perceptron	Dropped	0.538	0.017	0.622	0.538	0.577	

A comparison of accuracy of all classifiers used have been carried out and finally it has been investigated that the J48 decision tree algorithm performed best with an accuracy of 97.7%.

4.2.5 Experiment 5 (Faculty of Pharmacy)

In this experiment, training and testing of six (6) classifiers were conducted separately using Multilayer perception, Naïve Bayes, J48, SMO, k Nearest Neighbor and modified k Nearest Neighbor algorithms; The 10 fold cross validation is also applied, allowing all instances to participate in both the training and testing process. Table 4.6 shows all statistics based on the evaluation measures we have selected (accuracy, precision, recall and f-measure);

Table 4.6: Statistical Analysis of Classifiers with Cross Validation (*Pharmacy*)

Algorithm	Class	TP Rate	FP Rate	Precision	Recall	F-measure	Accuracy
KNN	Retained	0.99	0.625	0.994	0.999	0.996	99.2%
	Dropped	0.375	0.001	0.75	0.375	0.5	
mKNN	1	0.375	0.996	1	0.998	1	99.64%
	0.625	0	1	0.625	0.5	0.625	
SMO	Retained	1	0.625	0.994	1	0.997	99.4%
	Dropped	0.375	0	1	0.375	0.545	
J48	Retained	0.999	0.25	0.998	0.999	0.998	99.6%
	Dropped	0.75	0.001	0.857	0.75	0.8	
Naïve Bayes	Retained	0.998	0	1	0.998	0.999	99.7%
	Dropped	1	0.002	0.8	1	0.889	
Multilayer Perceptron	Retained	0.999	0.375	0.996	0.999	0.998	99.5%
	Dropped	0.625	0.001	0.833	0.625	0.714	

A comparison of accuracy of all classifiers used have been carried out and finally it has been investigated that the Naïve Bayes algorithm performed best with an accuracy of 99.7%.

4.2.6 Experiment 6 (Faculty of Law)

In this experiment, training and testing of six (6) classifiers were conducted separately using Multilayer perception, Naïve Bayes, J48, SMO, k Nearest Neighbor and modified k Nearest Neighbor algorithms; The 10 fold cross validation is also applied, allowing all instances to participate in both the training and testing process. Table 4.7 shows all statistics based on the evaluation measures we have selected (accuracy, precision, recall and f-measure);

Table 4.7: Statistical Analysis of Classifiers with Cross Validation (*Law*)

Algorithm	Class	TP Rate	FP Rate	Precision	Recall	F-measure	Accuracy
KNN	Retained	0.993	0.833	0.987	0.993	0.99	98.01%
	Dropped	0.167	0.007	0.267	0.167	0.205	
mKNN	0.999	0.792	0.988	0.999	0.993	0.999	98.7%
	0.208	0.001	0.714	0.208	0.323	0.208	
SMO	Retained	1	1	0.985	1	0.992	98.46%
	Dropped	0	0	0	0	0	
J48	Retained	0.992	0.833	0.987	0.992	0.99	97.94%
	Dropped	0.167	0.008	0.25	0.167	0.20	
Naïve Bayes	Retained	0.961	0.167	0.997	0.961	0.979	95.8%
	Dropped	0.833	0.039	0.25	0.833	0.385	
Multilayer Perceptron	Retained	0.993	0.667	0.99	0.993	0.991	98.2%
	Dropped	0.333	0.007	0.421	0.333	0.372	

A comparison of accuracy of all classifiers used have been carried out and finally it has been investigated that the modified k nearest neighbors algorithm performed best with an accuracy of 98.7%.

The Knowledge Flow interface which renders configurations for streamed data processing as applied to all experiments is shown in Figure 4.1

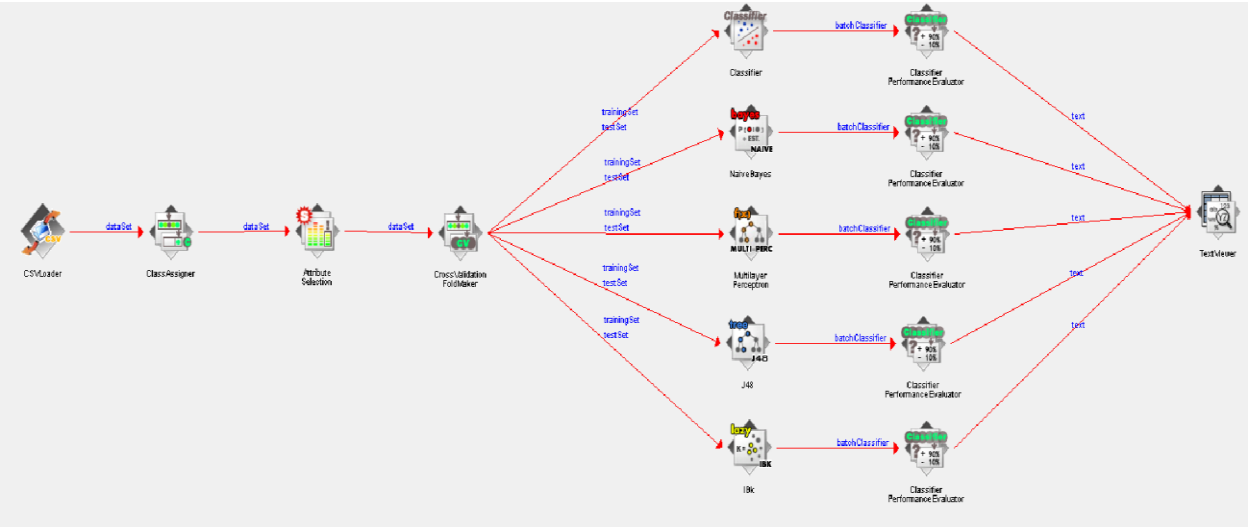


Figure 4.1: WEKA Knowledge Flow Model for all experiments

The results of experiments carried out with respect to the accuracy metric is summarized in Table 4.9 and Figure 4.2

Table 4.8: Summary of accuracy measure for all classifiers used

	Maths	**Building**	**Agric**	**MassComm**	**Pharmacy**	**Law**	**Average**
KNN	92.3%	95.9%	96.2%	95.6%	99.2%	98%	96.2%
mKNN	94.5%	97.9%	97.1%	96.1%	99.6%	98.7%	97.3%
SMO	94.8%	95.1%	97.0%	96.6%	99.4%	98.5%	96.9%
J48	95.3%	98.7%	98.2%	97.7%	99.6%	97.9%	97.9%
NaiveBayes	92.5%	92.6%	95.7%	94.3%	99.7%	95.8%	95.1%
MLP	93.7%	97.6%	97.6%	96.1%	99.5%	98.2%	97.1%

Table 4.8 shows a summary of accuracy (%) achieved by the selected classifiers in building the models. In Table 4.8, the average of each classifier's accuracy metric recorded during the experiments was reported (datasets are divided by academic faculties or departments). The J48 decision tree algorithms performs best based on this data with a final average of 97.9%.

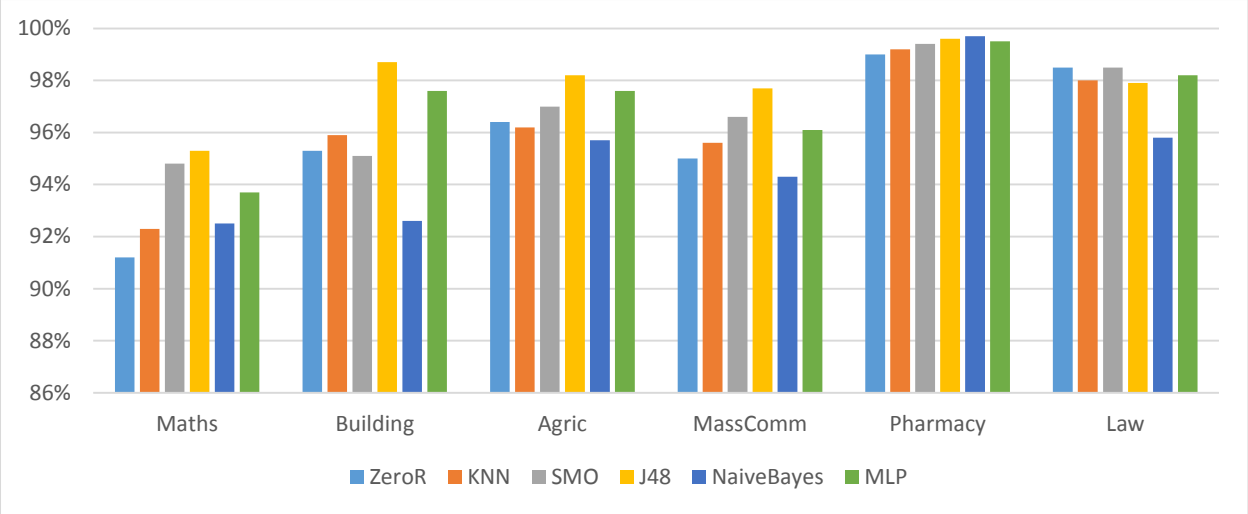


Figure 4.2: Accuracy Comparison Chart

Figure 4.2 is a bar chart showing a pictorial representation of the result summaries recorded in Table 4.8.

CHAPTER FIVE: SUMMARY AND FUTURE WORK

5.1 Summary

Nowadays, there are several machine learning algorithm and so, EDM users normally use knowledge discovery tools for data collection, pre-processing, mining, pattern evaluation and knowledge presentation. This dissertation has applied structured query language for data collection and selection, python PL pandas library for data preprocessing, python PL scikit learn library/WEKA for building classifiers and confusion matrix for performance evaluation of classifiers. These tasks are most important in classification experiments because of its interestingness, usefulness and applicability. In this study, machine learning algorithms were considered, specifically a modified k nearest neighbor algorithm was proposed.

5.2 Conclusion

In conclusion, this dissertation has employed classification techniques for prediction on students' datasets, to build models that will assist in the early detection of those at risk of attrition. The models were developed based on: 1) selected students' related inputs variables collected from two data sources of A.B.U Zaria information system 2) selected supervised machine learning algorithms. We have used six (6) classification algorithms which include Multi-Layer Perceptron, Naïve Bayes, J48 Decision Tree, Sequential minimal optimization, K-Nearest Neighbor and modified K-Nearest Neighbor; among all classifiers used K-Nearest Neighbor classifier was selected and modified due to its simplicity and robustness.

Among six (6) data mining classifiers used, J48 decision tree algorithm performed well with an average accuracy of 97.9%; also our modified kNN classifier recorded a better performance than the traditional kNN with an accuracy of 97.3%. Consequently, the J48 algorithm proves to be potentially effective and efficient based on the datasets retrieved for this work.

A primary contribution of this dissertation is the extension of experiments carried out by (Parneer *et al.*, 2015) on student academic record with a proposed **modified k nearest neighbor** algorithm for building a model that will detect at an early stage students at risk of attrition within the context of A.B.U Zaria.

A limitation encountered during this work is the inability to collect and utilize all variables that influence students' attrition, due to sparse and unclean data storage.

5.3 Future Work

This work is a starting point of Educational Data Mining research in Ahmadu Bello University, Zaria and can be further developed in various ways by:.

1. Using different techniques to learn the classification models, or a combination of classifiers, i.e. ensemble classifiers or meta-classifiers, to improve the performance results.
2. Using more data: the research used datasets of six (6) departments for the model development. The inclusion of other departments in the model development process can bring a different perspective and allow the University to gain a better understanding of the attrition challenge.
7. Integrating these data mining techniques with university databases, portals, and e-learning platforms.

References

Arthur, S. (1959, February). *Some Studies in Machine Learning Using Game of Checkers*. Retrieved May 16, 2016, from Great Works in Computer Science: https://www.cs.virginia.edu/~evans/greatworks/samuel1959.pdf

Ascend Learning, L. (2016, 02 07). *Student attrition: Consequences, contributing factors, and remedies.* Retrieved from www.atitesting.com/Libraries/pdf/Attrition_whitepaper_ATI_2.sflb.ashx

Baker, R., & Yacef, K. (2009). The State of Educational Data Mining in 2009: A Review and Future Visions. *Journal of Educational Data Mining*, 3.

Bean, J. (1990). Using Retention Management in Enrollment Management. *San Francisco,*.

Boufardea, E., & Garofalaskis, J. (2012). A Predictive System for Dsitance Learning Based on Ontologies and Data Mining. *The Fourth International Conference on Advanced Cogntive Technologies and Applications*, 151-158.

Brownlee, J. (2014, March 21). *Classification Accuracy is Not Enough: More Performance Measures You Can Use*. Retrieved from Machine Learning Mastery: http://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/

Chamberlin, D. D., & Boyce, R. F. (2016, 02 10). *SEQUEL: A Structured English Query Language*. Retrieved from IBM: http://www.almaden.ibm.com/cs/people/chamberlin/sequel-1974.pdf

Cost, S., & Salzberg, S. (1993). A Weighted Nearest Neighbor Algorithm For learning with Symbolic Features. *Machine Learning (10)*, 57-78.

Date, C. J., & Darwen, H. (1997). *A Guide to the SQL standard : a users guide to the standard database language SQL.* USA: Addison Wesley.

David, A. W. (1992). Tolerating noisy, irrelevant and novel attributes in instance based learning methods. *International Journal of Man Machine Studies (Vol 36)*, 267-287.

Dekker, G., Pechenizkiy, M., & Vleeshouwers, J. (2009). Predicting Students Drop out: A Case study. *2nd International Educational Data Mining Conference (EDM09)*, 41-50.

Dietrich, W. (1994). *PHD Thesis: A Study of Distance Based Machine Learning Algorithms.* Oregon: Oregon State University.

Druzdzel, M., & Glymour, C. (1994). Application of the TETRAD II Program to the Study of Student Retention in US Colleges. *Workshop on Knowldege Discovery in Databases*, 419–430.

Gorunescu, F. (2011). *Data Mining: Concepts, Models and Techniques.* Berlin: Springer.

Heydt, M. (2015). *Learning Pandas.* Birmingham: Packt Publishing Ltd.

Information Management Resource Association. (2015). *Human Rights and Ethics: Concepts, Methodologies, Tools, and Applications.* Hershey P.A, USA: Information Science Reference (An imprint of IGI Global).

International EDM Society. (2011, July). *Educational Data Mining*. Retrieved March 20, 2016, from Educational Data Mining: http://www.educationaldatamining.org

Jiawei , H., & Micheline , K. (2006). *Data Mining: Concepts and Techniques.* San Francisco: Elsevier Inc.

Jusoff, K., Samah, A., & Isa, P. M. (2008). Promoting University Community's Creative Citizenry. *World Academy of Science*, (pp. 1-6).

Kabakchieva, , D. (2012). Student Performance Prediction by Using Data Mining Classification Algorithm. *International Journal of Computer Science and Management Research, 1*(4), 686-690.

Kotsiantis, S. B., Zaharakis, I. D., & Pintelas, P. E. (2006). Machine learning: a review of classification. *Artif Intell Rev (2006), 26*, 159–190.

Kotsiantis, S. B., Zaharakis, I. D., & Pintelas, P. E. (2007). Machine learning: a review of classification and combining techniques. *Springer: Artif Intell Rev*, 159.

Kuyoro et al. (2013). Framework for Recommendation System For Tetiary Institution. *International Journal of Comuter and Information Technology*, 648-657.

Lawal, M. (2015, April 15). ABU, Zaria, develop in-house software applications to become e-compliant-- V-C. *New Agency of Nigeria*.

Luan, J. (2016, 02 10). *Data Mining Applications in Higher Education*. Retrieved from www.spss.com: http://www.spss.com/home_page/wp2.htm

Massa, S., & Puliafito, P. (1999). An application of data mining to the problem of the university. *3rd European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'99)*, 51-60.

Minaei-Bidgoli, B. (2003). Predicting student performance: an application of data mining methods. *33rd ASEE/IEEE Frontiers in Education*, 1-6.

Náplava, P., & Šnorek, M. (2003, October). Modeling of Student's Quality by Means of GMDH Algorithms. *Systems Analysis Modelling Simulation, 43*, 1415–1426.

Natarajan , M., Nabendu, C., & Dhinaharan, N. (2012). *Advances in Computer Science and Information Technology. Computer Science and Engineering.* Bangalore, India: Springer Berlin Heidelberg.

Ozga, J., & Sukhnadan, L. (1998). Undergraduate Non-Completion: Developing an Explanatory Model. *Higher Education Quarterly; , 52*(3), 1-3.

Pang-Ning, T., Steinbach, M., & Kumar, V. (2009). *Introduction to Data Mining.* Boston: Pearson Addison Wesley.

Parneer, K., Manpreet, S., & Gurpreet, S. (2015). Classification and prediction based data mining algorithms to predict slow learners in the educational sector. *3rd International Conference on Recent Trends in Computing* (pp. 500-508). Elsevier B.V.

Ramist, L. (1981). *College Student Attrition and Retention.* New York: College Entrance Examination Board.

Sanjeev, A. P., & Zytkow, J. M. (1995). Discovering Enrollment Knowledge in University. *1st International Conference on Knowledge Discovery and Data (KDD 95).*

Seidman, A. (2005). *College Student Retention: Formula for Student Success.* ACE available through Greenwood Publishing.

Stillman, M. (2016, February 06). *Making the case for the importance of student retention*. Retrieved from http://www.pacrao.org: http://www.pacrao.org/docs/resources/writersteam/MakingtheCaseforStudentRetention.pdf

Sumathi, S. (2007). *Fundamentals of relational database management systems.* Berlin Heidelberg: Springer.

Tinto. (2004). *Student Retention and Graduation: Facing the Truth, Living With the Consequences.* Washington, DC: The Pell Institution for the Study of Opportunity in Higher Education.

Tinto, V. (1975). Dropout from higher education: A theoretical synthesis of recent research. *Review of educational research*, 89-125.

Turing, A. M. (1950). *Computing Machinery and Intelligence.* Mind.

Witten, I. H., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques.* San Francisco, CA: Morgan Kaufmann Publishers(An Imprint of Elsevier).

Wong, S., & Li, C.-S. (2006). *Life Science Data Mining.* Singapore: World Scientific Publishing Co. Pte. Ltd.

Yong, J. P., Ci , L. F., & Yang, P. (2014). Research on Database and Information Technology in the Decision Tree of Communication Construction Scheme. *Advanced Materials Research; Vol 1046*, 461-464.