MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

Data Mining Techniques to Improve Predictions Accuracy of Students' Academic Performance: A Case Study with Xorro-Q

A thesis presented in a partial fulfilment of the requirements for
Master of Information Science (IT)
At
Massey University
Auckland
New Zealand
in 2018

Gomathy Suganya
(Supervisor: Dr. Teo Susnjak)
(Co-Supervisor: Dr.Anuradha Mathrani)

Abstract

Recent research in analytics has assisted policy makers capitalize on their ever-increasing data repositories and make data-driven predictions to create a vision for developing strategies to achieve their business targets. This is especially relevant in educational environments where data mining techniques can be applied to make predictions around students' academic performance. This can help educators align a teaching strategy which encourages and assists students with their learning. Suitable pedagogical support can be provided to enhance the overall student learning experience.

This study is in the educational domain where student-related course data has been used to extract insights on student performances over the study period. Extensive data collected from an educational tool (Xorro-Q) used in an engineering course delivery has aided this investigation. Data collected from Xorro-Q comprised student scores from real-time and self-paced activities set by educators over a 12-week semester period along with students' final Exam scores and scores from a compulsory prerequisite course. Popular data mining techniques have been applied to predict the academic performance of students based on data extracted from Xorro-Q. This is done by training the classifier using four different algorithms, namely, Naive Bayes, Logistic regression, K nearest neighbour and Random Forest. Process mining techniques have been applied along with the general features to find out the effectiveness, such as improvement in accuracy of predictions. The study has further implications in enhancing value of the role of analytics for predictive modelling by incorporating process mining features in the training set of data.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Dr. Teo Susnjak and Co-supervisor Dr. Anuradha Mathrani for providing valuable guidance, comments and suggestions throughout the course of my project. But for their help I would not have completed my project successfully.

I would like to acknowledge my client Pablo Garcia for providing me the data and for the countless hours he has spent in discussing with me the way forward and the potential outcomes of this project.

I wish to express my sincere thanks to Dr.James Lim for clarifying my doubts and giving valuable suggestions for correction modification and improvement of the project.

Finally Iam thankful and fortunate enough to get unconditional love and constant encouragement from my friend Rahila Umer and my little son Sai Dhanwin which helped me in successfully completing the project on time.

PUBLICATIONS AND PRESENTATIONS

Publications generated from this project so far:

Suganya, G., Susnjak, T., Mathrani, A., Lim, J., Garcia, P. (2017). Data Mining Techniques to Improve Predictions Accuracy of students' Academic Performance: A Case Study with Xorro-Q. *Proceedings of the 11th International conference on Data Mining. Computers, Communication and Industrial Applications.*

Presentations given from this project so far:

1. Suganya, G., Susnjak, T., Mathrani, A., Lim, J., Garcia, P. (2017). Data Mining Techniques to Improve Predictions Accuracy of Students' Academic Performance: A Case Study with Xorro-Q. *Oral presentation at the 11th International conference on Data Mining. Computers, Communication and Industrial Applications, Kuala Lumpur, Malaysia.*

2. Suganya, G., Susnjak, T., Mathrani, A., Lim, J., Garcia, P. (2017). Data Mining Techniques to Improve Predictions Accuracy of Students' Academic Performance: A Case Study with Xorro-Q. *Poster session presented at the 5th INMS Postgraduate students at Massey University, Auckland, New Zealand, 26th October 2017.*

3.Suganya, G.(2017).*Data Mining Techniques to Improve Predictions Accuracy of Students' Academic Performance: A Case Study with Xorro-Q.* Seminar delivered on 25th August 2017 to academic and industry personnel involved in this project.

ACHIEVEMENTS

Award won Suganya, G., Susnjak, T., Mathrani, A., Lim, J., Garcia, P. (2017). Data Mining Techniques to Improve Predictions Accuracy of Students' Academic Performance: A Case Study with Xorro-Q. Won the best session paper in *11th International Conference on Data Mining, Computers, Communication and Industrial Applications.*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Recent advancements in various fields have prompted the collection of large amounts of data that consequently is stored formats such as text records, images, files, and videos. This data is analysed to extract meaningful insights for decision-making processes; however, due to the vast amount of data stored, analysing it becomes complex and challenging. To effectively use such data for better decision-making, better methods of extracting knowledge from large repositories are required as a first step. Data mining can then be applied to obtain valuable and meaningful knowledge from large amounts of data(Mueen, Zafar, and Manzoor 2016).

Data mining, which is also referred to as Knowledge Discovery in Databases (KDD) is an emerging IT field of research which has much potential to improve decision-making processes and enable researchers to make useful discoveries from data. This is important especially for businesses as critical information retrieved from various data sources can inform the management on key decisions which can benefit the business. The main data source for the data mining are databases, data warehouses and other repositories. To enhance the accuracy of the mining function, some data preparation such as data integration, transformation and cleaning are done before mining(Ribeiro 2013)

(Giudici 2005, p.2) defines data mining as "the process of selection, exploration and modelling of large quantities of data to discover regularities or relations that are at first unknown with the aim of obtaining clear and useful results from the database". Detected patterns are then applied to new subsets of data to validate findings. One particular advantage of data mining is that it catalogues all the relationships found among the data without focusing on the source of the relationships(Nyce and CPCU 2007).

Data mining has been applied to numerous fields including banking, finance,

retail sales, the health sector and education. A new field of research gaining interest is known as Educational data mining (EDM) which applies data mining in educational environments(R. S. Baker, Corbett, and Koedinger 2004). In contrast to broader data mining applications, EDM methods often integrate methods from psychometrics, machine learning and data mining, to discover and exploit multiple levels of hierarchy within the educational data(R. Baker et al. 2010).

In today's competitive world, higher education institutions need to provide efficient and quality education to their students. To help them in this task there is a treasure trove of educational data about students and learning behaviour lying hidden in various databases in these institutions. This data can be leveraged for making informed decisions and is a vital resource for institutions to use for improving the quality of their teaching and understanding how they should deliver courses to current generation of students. By analysing this data using data mining techniques, educational institutions can gain useful knowledge to improve the learning experience of students who are enrolled in their courses.

EDM can help researchers discover distinct patterns within the voluminous data and create data-driven strategies which lead to enhanced teaching practices by educational institutions. To do this, machine learning and statistical and visualization techniques are used to analyse the information discovered and serve it up in a manner whereby it can be consumed and understood by education providers, students and other users of the data.

Data analysed in EDM includes all interactions made with an online educational platform which are recorded in an event log. This takes account of individual student interactions, such as their navigation behaviour, inputs to quizzes and interactive exercises, besides also gathering group data posted by participating students like text chat or discussion forums and teacher interaction like typing, mouse clicking. Other datasets which can be analysed are administrative data (e.g. school, school district) and demographic data (gender, school grades etc.)(Cristobal Romero and Ventura 2013).

Machine learning algorithms can be can be categorised as supervised or unsupervised learning, based on the type of input data. Supervised learning examines records that have a known outcome, for example, supervised learning is used to study the academic behaviour of students with the intent to link student behavioural patterns to academic history and other recorded information. Whereas unsupervised learning is used in a situation where the patterns are unknown. Unsupervised learning is utilized first to study the patterns and look for previously hidden patterns, and to understand, classify and code the objects of study before applying theories (Luan 2002). This project focus on supervised learning, particularly predictive analysis, where machine learning is utilized to predict future outcomes(Nyce and CPCU

2007).

Machine learning utilizes artificial intelligence to enlist rules and describe patterns that the analyst can apply to new data. Machine learning generally refers to the changes in system that perform tasks related with artificial intelligence. The tasks include recognition, diagnosis, planning, and predictions amongst others(Nilsson 1996). Once a model performs well on previously seen data, the analyst can feed in new data and the model can be used to predict and understand aspects of newly observed data(VanderPlas 2016). All the processes involved are automated to deliver accurate estimations quickly when compared to conventional behaviour prediction methods(Luan 2002). As for statistical approaches, background knowledge is required initially in the development stage, but operations are performed without human interference.

Process mining (PM) is one of the techniques used in EDM. The idea behind PM is to extract knowledge from event logs generated within educational environments and to identify, scrutinise and improve processes(Awatef HICHEUR Cairns et al. 2015). The data about every event contains the time stamp and is linked with the learning process which can be participating in student forums and chats, viewing of educational material or performing assessment activities through quizzes. This can be combined in to different groups based on the behaviour of the students(Grigorova, Malysheva, and Bobrovskiy 2017). PM generally involves construction of a process model and conformance checking. The process model is generally represented via a Petri net.

1.1.1 This study's context

There are many studies available that predict the academic performance of students by applying EDM techniques. The present study uses extensive data collected from an educational tool known as Xorro-Q(`www.xorro.com`), which was used to assist in the delivery of an engineering course. Xorro-Q is an audience communication tool that enables synchronous as well as asynchronous interactions with the presenter (teacher) and the viewers (students). The advantage in using Xorro-Q for the teacher and the institution is that Xorro-Q captures all classroom interactions providing them with evidence of student engagement activities, which in turn may help in answering questions pertaining to student retention in courses or how to bring about improvements in student learning outcomes.

This study focuses on EDM techniques to explore the data collected from Xorro-Q. Data comprising student activities over the duration of the course has been investigated with appropriate EDM techniques to share insights on effective teaching and learning strategies. The insights gained from this research can help teachers to

better understand student learning behaviours and also to identify links between learning behaviour and positive or negative outcomes. The information can also be beneficial if presented to students as it enables them to consider how effective their learning habits are based on their current performance.

1.1.2 Scope and study objectives

Primarily, EDM research is about obtaining a greater understanding of the key aspects that influence how students learn. Investigating student learning behaviours and their links to learning outcomes is good way to achieve this. As information and communication technology continues to advance at a rapid rate, the amount of data being collected and stored about students and learning processes continues to increase. This data can be analysed to obtain insights to help to improve learning outcomes. Therefore, EDM researchers analyse academic databases to identify patterns in the data and develop new learning strategies based on the data. This is what has motivated the present research study which focuses on identifying student behaviour related to the use of Xorro-Q.

The main objective of this thesis is to evaluate the effectiveness of EDM methods for predicting student academic performance using Xorro-Q and to incorporate process mining features to find out whether or not those features help in increasing the accuracy of predicting student performance. To achieve these objectives, a popular data mining technique, namely a classification technique, is utilized and various algorithms under classification have been tested to find the best algorithm to predict student academic performance. Indicators commonly used to assess the effectiveness of machine learning algorithms include precision, recall and F-measures(Powers 2011). These indicators can be utilized to evaluate the various predictive models. More details about these indicators can be found in later chapters.

1.1.3 Research Questions

- Is it possible to predict students' final course grades and outcomes based on data gathered through a synchronous and asynchronous in-class participation technology (Xorro-Q)? If so, which data mining algorithms provide most accurate predictions, and how early can we reliably predict a student's final course outcome?

- Is it possible to improve the predictive accuracy of machine learning algorithms which use features extracted from in-class participation technology (Xorro-Q), by combining with features extracted from process mining? If in-class participation technologies (such as Xorro-Q) improve machine learning outcomes

by generating valuable features, is there evidence in the captured datasets that indicate that there are benefits for students to using these technologies? Did students who performed poorly in the prior course, perform better in the course which used an in-class participation technology (Xorro-Q)? And, for those who performed better, was the extent to which they performed better, related in any way to their activity on Xorro-Q?

1.1.4 Research contributions

The contributions of this thesis are as follows:

- Provide insights and information about how EDM methods could be utilized to gain knowledge from the data gathered from Xorro-Q, an education tool.
- Perform a detailed analysis of the features that can identify student academic performance using different machine learning algorithms. Then conduct statistical testing to determine whether or not the performance of all the classifiers which were used for this experiment are the same.
- Integrate process mining, or more specifically conformance checking, to measure the effectiveness of the general features that help to increase the accuracy of predicting student academic performance
- Identify specific student learning behaviour in the Xorro-Q that leads to positive and negative outcomes.
- Obtain pedagogical knowledge that both teachers and students can use to help enhance student performance.

1.1.5 Thesis outline

This thesis is organized as follows:

Chapter 1 introduced the key concepts of data mining and educational data mining. The scope of this study was stated, the research questions were defined and finally, the research objectives were revealed.

Chapter 2 propose a summary of educational data mining research, the various types of data used in data mining, applications of educational data mining, related studies in educational data mining and various data mining methods used in higher education.

Chapter 3 focusses on process mining research, various types of process mining, the tool which is implemented for this research and finally related study in process mining.

Chapter 4 explains the two data sources that are used for this project.

Chapter 5 describes the research approach and methodology, the different steps involved in pre-processing the data is explained, and also talked about the various machine learning procedures involved in analysis of data mining.

Chapter 6 presents the results of the study and a discussion of the results.

Chapter 7 concludes the thesis. In addition, limitations of this study are discussed and ideas for future work are proposed.

Chapter 2

Educational Data Mining Research

2.1 Introduction

Poor academic performance of students is a concern in the educational sector, especially if this leads to students being unable to meet minimum course requirements. In an attempt to resolve this issue, a range of procedures, both formal and informal, depend on qualitative and quantitative techniques have been employed by higher education institutions; however, these attempts have been far from successful(Abaidullah, N. Ahmed, and Ali 2015; Delavari, Phon-Amnuaisuk, and Beikzadeh 2008). Such attempts to improve academic performance are mostly based on analysis of the data via predefined queries and charts. But the methods used do not reveal useful hidden information and they give a simplistic view of the problem domain(Abaidullah, N. Ahmed, and Ali 2015). Decision makers either do not receive or are not capable of retrieving rich and useful information from the data. This can be overcome by applying EDM on these data(Hanna 2004).

This chapter reviews recent research conducted in the area of EDM and provides a guideline for the research undertaken in this thesis. Section 2.2 covers the different classes of data used in educational environment. Section 2.3 describes the various types of data which are used in EDM research, followed by goals on EDM research on section 2.4. Section 2.5 talks about the benefits and success factors of applying EDM. The applications of EDM are explained in section 2.6. Reviews about the EDM methods and techniques which are commonly used are discussed in the subsequent sections. Finally, this chapter concludes with a review of some of the related work in EDM research.

2.1.1 Different classes of Educational environments

Across the educational domain, in traditional education and computer-based education, a broad range of educational environments and information systems exist (Ref

Figure 2.1: Different types of traditional and computer-based educational environments and systems

(Source: Cristobal Romero and Ventura 2013)

fig 2.1). Each environment and system have different types of data sources that must be pre-processed in different ways. Therefore, DM methods used must take into account the type of data and how it is stored and any other issues specific to the data source.

Traditional Education

Traditional educational systems predominantly feature face-to-face interaction among teachers or instructors and students organized via formal lectures, tutorials, small groups etc. Information collected per course includes student attendance, student marks, and educational program objectives. The educational institutions also store administrative data in traditional databases including student information, instructor information, and course information. The computer based educational system is also used as a complementary tool by the traditional education provider during the face-to-face sessions(Cristobal Romero and Ventura 2013).

Computer based education system

Computer-based education systems use computing devices to provide direction, and instructions, and manage instructions given to the students. They include web-based

systems such as e-learning systems, e-training systems. Commonly used computer-based education platforms are learning and management systems that incorporate course-delivery functions which record student activities such as reading, writing, taking tests, test results, and commenting on events with peers. An intelligent tutoring system (ITS) is another platform which records all relevant student–teacher interactions (mouse clicks, typing) from test and quiz system, and forums etc(Cristobal Romero and Ventura 2013).

2.1.2 Types of data used in EDM research

Data is saved by online educational learning systems and tools in different formats, specific to the tools used in those systems. The data objects can be text records, instances, observations etc. and there is a good possibility that the data is stored in a database(Janecek 2009). However, the type of data stored, rely on the type of database and the setup of the database itself as current databases can efficiently store data in multiple formats, for example, .dat, .text or log files (Rajibussalim 2014). Data may be stored in an educational system in formats such as text format, numerical format, web server log files and learning software log files (Black, Dawson, and Priem 2008).

Recorded Text data

Various researchers have utilized data mining techniques on text data captured in learning management systems and computer-supported collaborative learning systems. High level information has been obtained from this especially rich source of data. This type of data has shown significant promise in the educational domain; however, utilizing such data via a machine learning algorithm is far from easy because the manual coding requirements are very time (Black, Dawson, and Priem 2008).

Web server log files

Another source of data which is generated by educational systems are web server logs from educational systems that are web-based or run on web servers. These are vast collections of data made up of records which are created when users access particular web pages that are part of or linked to an educational application or website. Several data mining techniques have been applied by various researchers to obtain useful knowledge from web server (Black, Dawson, and Priem 2008).

Learning Management System (LMS) log files

A most promising source of online learning data is Learning Management Systems (LMS) log files, because students normally sign in to LMS and, therefore, monitoring of individual users and sessions, which is a key problem when exploring web server logs, can be completed automatically. Several research studies have mined data from LMS log files to analyze groups in web-based learning(Black, Dawson, and Priem 2008).

2.1.3 Goals of EDM research

Researchers have begun to examine the potential of applying data mining techniques to educational data. To provide high quality education to their students, advanced educational institutions need a good knowledge of their student base and student study patterns. A goal of EDM is to obtain that information to give these institutions an insight into their students with a view to enhancing learning and reducing failure rates (Feng et al. 2008).

Another conceivable goal is to explore new models and identify possible changes to the current models of education, for Example, to outline the educational content to determine ideal instructional sequences and strategies to support student learning models (Castro, Nebot, and Mugica 2007; Rajibussalim 2014).

2.1.4 Benefits and success factors of education data mining

A number of issues and problems in the educational domain can be addressed by using quality data and knowledge retrieved from educational systems. Several data mining techniques could be utilized for this purpose. For example, comprehensive characteristics of students can be analysed by data mining tasks like clustering, while classification and regression are useful in forming policies and initiatives to reduce student dropout rates and to promote success and positive learning outcomes. Information gathered by EDM techniques can be used to enhance the efficiency of the educational system, reduce costs and even help tailor educational products to create more personalised education services for students (Zhang et al. 2008).

A centralized system for collecting all educational data for an institution is a must for the successful application of data mining in higher education. With this system in operation, the factors that are important to assess and improve the quality of service provided by the institution can be monitored (Chalaris et al. 2014).

Apart from having such a system as an important part of their infrastructure, the higher education provider could use an analysis model as a road map for the institution. A data mining analysis could help to determine parts of the education

delivery process which could be improved and methods to use to achieve that goal.

2.1.5 Main applications of EDM

One important area of application is to build student models. These models can give researchers complete information about student characteristics like knowledge, motivation and attitude. One key theme in EDM research is to model the individual differences between the students. From this modelling, researchers can make very good inferences about student behaviour. For Example, when a student "slips" (makes a blunder despite having good skills), and when a student is performing well in an activity. Also, these models have enabled the researchers to predict the future performance of students, their knowledge and the factors that influence the choices they make in learning (R. Baker et al. 2010; Shih, Koedinger, and Scheines 2011).

A second key area concerns the models of knowledge structure of the domain with the intent to improve them. Various educational data mining techniques exist to rapidly discover a model directly from the data it generates. These methods consolidate a psychometric modelling framework with advanced space-searching algorithms. Using different domain models, they act as prediction problems for model discovery, such as predicting if individual actions will be right or wrong(R. Baker et al. 2010).

A third key area is to study pedagogical support for students and discover which pedagogical support is most effective. This has been a key area for educational data miners(R. Baker et al. 2010; Beck and Mostow 2008).

A fourth key area of application of educational data mining is to gain knowledge about learning and learners. The created models and learning decomposition techniques are used to conduct research into learning and learners(R. Baker et al. 2010).

2.1.6 EDM methods

Increasing use of information and communication technologies, along with a continuously growing body of research have led to an exponential growth in the amount of data, and complexity of data, being stored by educational institutions. Furthermore, without a direct relationship between inputs and outcomes, analysis and modelling of this data is challenging. Machine learning algorithms are, therefore, often used to extract information from these datasets and, likewise, are used in data discovery and analysis (Livieris, Tampakas, Drakopoulou, Pintelas, Mikropoulos, 2018). The following are the predominant categories of machine learning algorithms proposed by Baker (Rajibussalim, 2014).

- Prediction involves Classification, Regression and Density estimation.
- Clustering
- Relationship mining comprises Association rule mining, Correlation mining, Sequential pattern mining and casual data mining
- Distillation of data for human management
- Discovery with models.

Of the above EDM methods which are prominently used by researchers, according to Baker (2010), prediction, clustering and relationship data mining are more popular compared to the other two categories.

2.1.7 Commonly used data mining techniques in EDM

Clustering, classification and association rule mining are the most commonly used EDM techniques. When these data mining techniques are applied to educational data, information can be extracted that can assist educators to make informed decisions regarding teaching methodology or to modify or optimise the content of a course. Data mining techniques can also be used to predict and analyse student performance. The following section provides an overview of data mining techniques that are commonly used in EDM research.

Clustering

A cluster, also known as unsupervised learning, is a group of alike data objects that share a similar attribute. The process of discovering groups and data clusters is called clustering and if two objects belong to the same cluster, the degree of association between them is higher than if they were in different clusters. When the most common categories or attributes classes of data objects are unknown, clustering can be especially useful. The dataset is separated into classes, depend on some attribute similarity, and then the cluster receives a distinctive label. The principle advantage of clustering is that it is a dynamic process and it adjusts rapidly to changes. Additionally, it effectively magnifies the attribute, that cluster members have in common, on which the cluster is based (Jacob et al. 2015). The fundamental goal of the clustering process is to divide the dataset so that the distance between data objects within a cluster should be minimal whereas the inter-cluster distance should be maximal(Şuşnea 2009). Using clustering in EDM can help institutions, for example, to group individual students by similar types of learning behaviour(Goyal and Vohra 2012). Statistical metrics such as the Bayesian Information Criterion are

used to assess the appropriateness of a set of clusters, with regard to how well the set of clusters fits the data, relative to chance(R. Baker et al. 2010).

Clustering techniques used in EDM include:

Partitioning Methods organizes the object of a set in to user defined number of clusters. k-means, k-medoids algorithm, CLARANS are some of the Examples of partitioning methods.

Hierarchical clustering which seeks to build a hierarchy of clusters, is utilized to obtain commonly used items from a large database (Priyadarshini 2017). The two types of hierarchical clustering are agglomerative and divisive where agglomerative is a "bottom up" approach which starts by considering every object as a cluster and successfully combines the two most similar clusters and continues until only one cluster is left out. Divisive is a "top down" approach which starts by considering all objects to belong to a single cluster and then splits to remove less similar objects are repeatedly performed, moving down the hierarchy(Ribeiro 2013).

Association rule mining

Association rule mining refers to a method that helps to identify relationships between different variables in the data. To do this, it searches the dataset for patterns and frequently occurring variables. The two measures of rule interestingness are support and confidence.

The two-step process involved in association rule mining is:

- Find all the frequent item sets, based on a predetermined minimum support count.
- Generate strong association rules that satisfy the minimum support and confidence from the frequent item sets.

In the education sector, association rule mining can help a student to search for useful learning behaviour patterns, guiding them to discover the best model for learning for them (Priyadarshini 2017).

Classification

Along with clustering and association rule mining, classification is another frequently used EDM technique. A classification algorithm classifies a given instance into a set of discrete categories. The value of a (categorical) attribute (the class) is predicted depends on the values of other attributes (the predicting attributes). A model can be built by using a training dataset where all the attributes are known including the class attribute and verified via a test dataset where the class attribute is unknown. In general, a classification technique works in a two-step process:

- learning step
- classification step.

In the learning step, the classification algorithm determines the classification rules and develops the classifier from the training set and the class labels.

Whereas, in the classification step, the classifier is used to classify test data to verify the accuracy of the classification rules that were determined in the first step. If deemed acceptable, the verified rules could be applied to the new data(A. B. E. D. Ahmed and Elaraby 2014).

This method is also called supervised learning since the labels or categories of the given instances are known beforehand, in contrast to unsupervised learning where the labels are unknown(Bishop 2006). The accuracy of the classification model can be evaluated by some validation techniques such as Hold-out, cross validation, bootstrap. Fundamentally, the evaluation of a model comprises of utilizing a testing data for predicting the known classes. The evaluation of the accuracy can be done by checking the correct and the incorrect predictions.

Classification techniques have been widely used in EDM, for instance, to classify students into different categories by their final marks depending on the outcome of activities they performed in a learning system (Vahdat 2017). There are a vast variety of classifiers in the literature but choosing the best classifier is not a simple task because they differ in numerous aspects such as learning rate, robustness, amount of data for training etc(Osmanbegović and Suljić 2012).

In this research study, the classifier has divided students into three classes based on their Exam scores: High risk, Medium risk, and Low risk. Four algorithms, namely, K nearest neighbours, Logistic regression, Random forest, and Naïve Bayes, were used for classification analyses. In this study, the main objective was to classify the final exam performance of students. The resultant model can be used to predict the final exam performance of new students. Such a classification method can be remarkably important in supporting education and learning. The classification algorithms which are used for this project are explained below.

Naïve Bayes

Naïve Bayes, based on Bayes rule, is a popular classification algorithm due to its simplicity, efficiency and performance applied to real-world problems. The approach is referred to as "naive" due to the assumption of independence between the various attribute values. The class with the highest probability is considered the most likely class and is also called Maximum A Posteriori (MAP)(Polamuri 2017). The variables are assumed to be independent variables so only the variances of the variables for every class needs to be determined, as opposed to the complete covariance matrix

(Pandey and Pal 2011). To make a prediction, Naïve Bayes uses the probabilities of each attribute belonging to each class. This model is used to predict an output value based on a set of input values. Bayes theorem gives a relationship between the posterior probability and the prior probability.

$$P(A\backslash B) = \frac{P(B\backslash A)P(A)}{P\,(B)}$$

Where $P(A\backslash B)$ is defined as the probability of observing A given that B occurs and is called the posterior probability. $P(B\backslash A)$, P(A) and P(B) are called prior probabilities. $P(B\backslash A)$ is the likelihood, i.e. the probability of data d for a given hypothesis h which was true; P(A) is the class prior probability, i.e. the probability of hypothesis h being true irrespective of the data; and P(B) is the predictor prior probability, i.e. the probability of the data(Shaw 2017).

The Naïve Bayes algorithm uses the Bayesian approach to classify a new instance in a training dataset. Given the input attributes, the Bayes rule is applied to find the probability of observing each output class and then the class with the highest probability and is assigned to an instance. Probability values used in the algorithm are obtained from counts of attribute values in the training set.

Strong independence assumptions are asserted by the feature model used by the Naïve Bayes classifier. Thus, the existence of a particular feature of a class is independent of every other feature(Nikam 2015). For instance, a fruit is deemed to be a banana if it is yellow and long. The Naïve Bayes classifier considers these two properties, along with any other properties, to independently contribute to the probability that a given fruit is a banana. Furthermore, all the attributes in the instance must be discrete and no attributes are allowed missing values as this would make it difficult to calculate the attribute's probability values. The Naive Bayes classifier can be easily trained on a small dataset. Since Naive Bayes considers all the features to be unrelated, it can learn only the importance of individual features and cannot determine the relationship between features.

Random Forest

A Random forest (RF) is a collection of trees built up with some element of random choice. Therefore, its component trees randomly differ from one another. Individual tree predictions are, thus, not correlated and, consequently, generalization is improved (Criminisi, Shotton, Konukoglu, et al. 2012). When creating a random forest, the trees are grown to maximal depth and an independent classification is performed for each tree. RF works by generating many trees to analyse the data then it combines all the output from the trees and then through the process of the vote to obtain the (Polamuri 2017). Some important features of RF are

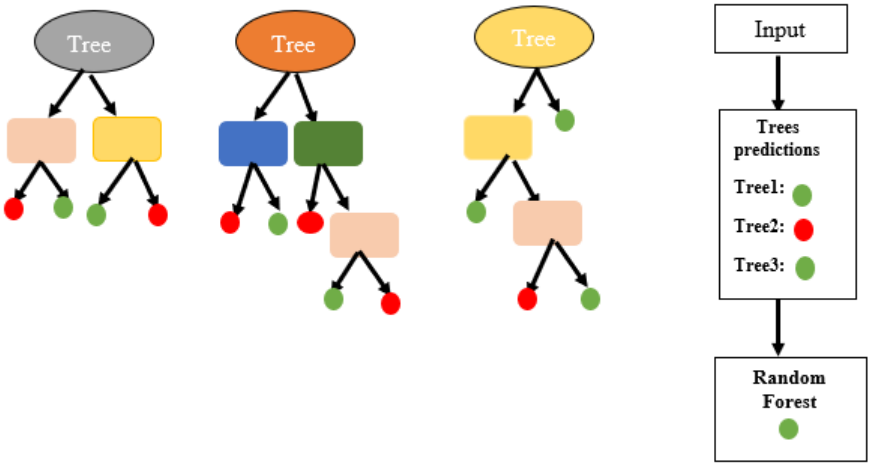


Figure 2.2: Randomly selecting features

- It has an adequate method for evaluating missing data
- It uses weighted random forest (WRF) for balancing errors in unbalanced data
- The significance of the variables used in the classification is estimated by RF.

RF has high robustness for large datasets. Thus, even if a significant portion of an attribute value is missing, often the desired accuracy is still achieved. Although RF is much more difficult to interpret compared to a single decision tree, it is still possible to provide explanatory knowledge in terms of its input variable relevance(Breiman 2001).

RF randomly selects K features from the total M features, where K¡¡ M. Among the K features, the root node D is calculated using the best split point and a tree with a node is formed and finally randomly created trees are generated. These randomly created trees form the random forest. Each randomly created tree is used to predict the outcome. Then the vote for each predicted target is calculated and the predicted target with the maximum vote is considered to be the final prediction for the classification and takes the average for the regression analysis. RF can be used for feature engineering while identifying the most important features out of the available features from the training dataset.

2.1.8 Logistic regression

Logistic regression (LR) is an extension of linear regression for cases with a binary dependent variable. Situations involving in categorical variables are common in practise(Dayton 1992). Well suited, in most cases, to explain and test hypotheses associated with relationships between a categorical outcome variable and one or more categorical or continuous predictor variables(Peng, Lee, and Ingersoll 2002),

Figure 2.3: Logistic regression curve
(Source: `:http://www.saedsayad.com`)

logistic regression is used to predict the probability that an event will occur. Better results are obtained where there exists some linear dependency among the data.

The output of a LR is a logistic curve, limited to values between 0 and 1. Although similar to a linear regression, but the curve produced by a logistic regression is based on the natural logarithm of the "odds" of the target variable, where the odds are the ratios of the probabilities of an event occurring to it not occurring.

However, the predictors don't need to be normally distributed, nor do they need to have equal variance in every group

The logistic function is also called as a Sigmoid function ("S" shape).

The aim of logistic regression is to predict the probability that an event will occur. The algorithm does not allow any missing attribute values. The prediction accuracy, which is also the error, is the absolute difference between the predicted output via the regression equation and the actual observed output (Polumetla 2006). The fitted model can be used, for example, to predict the probability that a new student will pass their final exam but also, importantly, to identify key factors that could influence a successful outcome, or otherwise(Hoffait and Schyns 2017).

Multi-Class Classification using Logistic Regression

Logistic regression is widely used for binary classification (e.g., email is spam or not, whether to give loan or not etc.). However, logistic regression can be extended for multi-class problems (e.g., shape is square or circle or triangle, vehicle is car or bus or truck or bike). Each point in the training dataset belongs to one of N different classes. The goal is to construct a function to correctly predict the class for new data point.

Two approaches are generally followed for Multi class classification using Logistic Regression.

One versus All

This is the simplest form of solving multi-class problem by building multiple

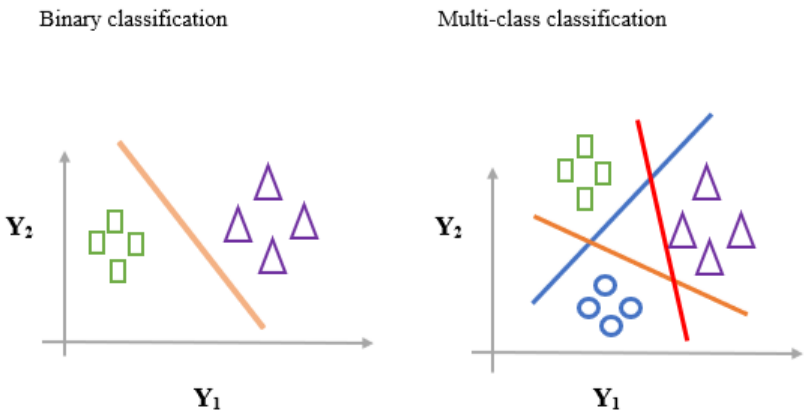


Figure 2.4: One Vs All method

binary classifiers. A binary classifier is built for each class to classify that class against all others. In the prediction process, a probability is obtained from each class. The highest probability calculated will indicate that the test observation should belong to that particular class. Suppose, a classifier was built to label input data into one of three classes:

- class 1
- class 2
- class 3

Following steps would be followed

- Build a classifier I with class1 as positive and class 2, class 3 put together as negative
- Build a classifier II with class 2 as positive and class 1, class 3 put together as negative
- Build a classifier III with class 3 as positive and class 1, class 2 put together as negative
- For a new input data, predict the probability of belonging to class 1, class 2, class 3 using classifier I, classifier II and classifier III respectively
- Classify the data based on the class which has the highest probability.

Multinomial Logistic Regression

In case of multinomial logistic regression, a multi class function is built. While function like Sigmoid is used for binary classification, function like Softmax is used for multi-class classification.

Given an input vector z, softmax does two things. First it exponentiates (elementwise) e^z , forcing all values to be strictly positive.Then it normalizes so that all values sum to 1. Following the softmax operation for k classes computes the following

$$softmax(z) = \frac{e^z}{\sum_{i=1}^{k} e^Z i}$$

The exponential (e-power) of the given input value and the sum of exponential values of all the input values are computed. Then, the output of the softmax function is the ratio of the exponential of the input value and the sum of the exponential values.

The properties of the softmax function are as follows.

- The calculated probabilities will be in the range of 0 to 1.
- The sum of all the probabilities is equals to 1.

k-Nearest Neighbours

The k-nearest neighbour (kNN) algorithm is a machine learning algorithm. It is classified as an instance-based learner, or a lazy learner, because there is no model building step and the generalization process does not occur until after the classification process is complete (Bhavsar and Ganatra 2012). The kNN algorithm remembers the whole training dataset and the test object is classified only if its attributes exactly match one of the training sample objects. The concept that underpins the k-nearest neighbour algorithm is that instances within a dataset with similar properties are more likely to occur close together.

The three important characteristics of this approach are a set of stored records, a similarity metric to calculate the distance between the objects, and k, the number of nearest neighbours.Euclidean distance, which is the length of the path connecting the two points, is used to determine the distance between the unlabelled object and its neighbours (Saxena 2016).

To classify the unlabelled object, first, calculate the distance between it and the labelled object, then identify its k-nearest neighbours, and use the class labels of these nearest neighbours to determine the class label of the unlabelled object. Once the list of nearest objects is obtained, the unlabelled object is classified based on the class of the majority of its nearest neighbours(Wu et al. 2008).

$$y = argmax = \sum_{(Xi,yi)Dz} I(v = yi)$$

Where v is the class label, yi is the class label for the ith nearest neighbour and I () is a function that returns the value 1 or 0 otherwise. D is the set of k training objects and test object z= (). The distance between z and every object (X,y) D is computed d(X, x), Dz D, the set of k closest training objects to Z.

Choosing the right value for k is important. If the value of k is too small then the result can be sensitive to noise points, but if it is too large, then the neighbourhood will consider too many points from the other classes. Therefore, k is usually a small odd (to avoid tied votes) positive number but to find the best fit, run through every possible value and test the results. Figure 2.5 shows how the value of k can affect the outcome of the kNN algorithm. If k is set to 7 or less, as in Figure 2.5(1), a majority count of the nearest neighbours of the object returns a diamond as the class. In contrast, if k=1 or 9, as in Figure 2.5(2), the output class is a circle.

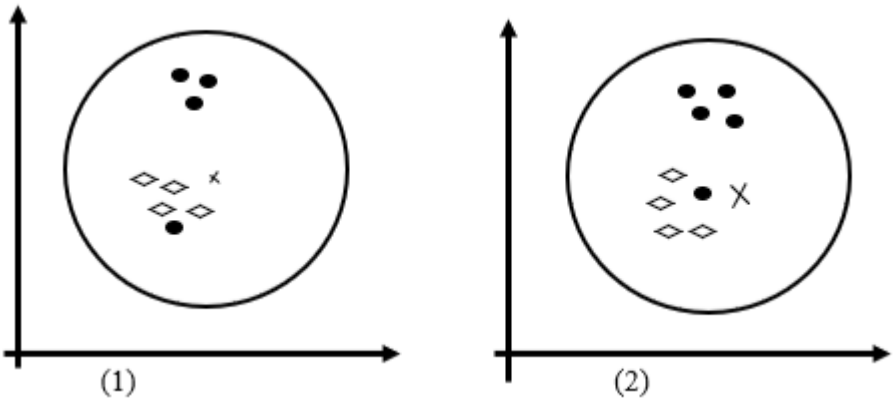


Figure 2.5: kNN classifier

Classifying based on majority vote is a problem if the neighbours in the nearest neighbours list are too far away and instead, the closer neighbours would more reliably determine the class of the object. In such a case, each neighbour object's vote is weighted by its distance from the object to be classified.

$$y = argmax = w \sum_{(Xi,yi)Dz} I(v = yi)$$

The attributes for a kNN classifier must be scaled to prevent one of the attributes from dominating the distance measures. For example, consider a data record where person's height varies from 5.4 to 6.5 feet, their weight varies from 60 to 120 kg, and their personal income varies from $5,000to1{,}000{,}000$. If there is no scaling of the distance measure, the income attribute will lead the computation of distance and a class will be assigned based on that(Wu et al. 2008).

If a new training pattern is added to the current training set, retraining is not needed as kNN does not require any prior knowledge of the data in the training set,

but testing will be time consuming as, for every test data object, the distance will need to be calculated between the test data object and each object in the training dataset(Sarkar and Leong 2000). The kNN algorithm is very well suited for multi-model classes and where an object has many class labels. The algorithm does not have any principled method by which to choose k, with the exception of through cross-validation or similar.

2.1.9 Related study

Predicting Academic performances

.

Data mining is a powerful tool that utilizes a blend of an explicit knowledge and domain-specific knowledge to reveal trends and patterns which shape predictive models that empower analysts to gain new knowledge and understand from existing data(Priya and A. S. Kumar 2013). Several studies have been reported success in predicting student academic performance via use of data mining techniques. In most of the studies, the authors have used either Weka or MATLAB in predicting the students' performance.

A study by Kaur, Singh and Josan (2015) attempted to identify slow learners using data mining techniques. A dataset of 152 students was created from high school academic records and five classification algorithms were applied to the data. Statics were generated using each of the five classification algorithms: multilayer perception, naïve Bayes, SMO, J48 and REPTREE. A comparison of these five classifiers was conducted to gauge prediction accuracy and to determine the best classification algorithm. To evaluate the attribute, they used ChiSquared attribute, Infogain attribute, Symmeterical Unvert attribute and Relief F attribute evaluator to evaluate the attribute. High potential variables were selected by applying the Rank search method technique of WEKA and final ranks were achieved by taking averages. Then results of all five classifiers were tested using WEKA an open source tool. Of all the classifiers, the multilayer perception proved to be the best with F-measures of 82%. This research helped the institution to identify slow learners which gave them the opportunity to provide those learners with additional assistance.

In another study, Badr, Algobail, Almutairi and Almutery (2016) predicted the future performance of students in a programming course based on grades obtained in other courses, specifically Maths and English. An association rule algorithm (CBA) was the basis of a classifier that evaluated student performance in the programming course and results indicated that their prior performance in English courses had a significant influence on the ability of the researchers to predict programming course outcomes(Badr et al. 2016).

Laci, Sergio and Geraldo (2014) present an approach that uses EDM techniques to discover drop out risks from a group of engineering students. The study conducted in a Brazilian public university looked at real data from three undergraduate engineering courses. Classifiers used for this experiment were Naïve Bayes, Multilayer perceptron, Support vector machine with polynomial Kernel and Decision table, and results showed that the Naïve bayes classifier produced the best predictive outcomes across all three datasets(Manhães, Cruz, and Zimbrão 2014).

Based on survey data collected from 189 students, Erman, Serhat, Kılıç (2014), used data mining techniques to predict which students would be likely to drop out of a course. Online questionnaires (demographic survey, online technologies self-efficacy scale, prior knowledge questionnaire, Locus of control scale, and Readiness for Online Learning Questionnaire) were used for data collection and four data mining approaches, namely Naïve bayes, k-Nearest Neighbor, Decision Tree and Neural Network, were applied to classify dropout students. The resultant highest predictive precision was 87% from the k-NN classifier, followed by 79.7%, 76.8%, and 73.9% from the Decision Tree, Naïve Bayes, and Neural network classifiers, respectively.

In a 2007 study that aimed for early discovery of potential issues, Vandamme classified first-year students into low, medium or high risk, based on analysis using decision trees, neural networks and linear discriminant analysis. Background information such as demographics and academic history was found to be significantly related to academic success for the first-year students. Yet three classification methods used in the study did not perform particularly well, with the best classification accuracy of 57.35% obtained via linear discriminant analysis.

A study was undertaken by Kabakchieva (2013) to determine if their personal and pre-university characteristics of university students' can be used to predict their performance at university. Based on a dataset of 10330 students, the CRISP-DM model (cross industry standard process for data mining) was run on a dataset of 10330 student records using WEKA software. Study results revealed that the best performer was the decision tree classifier J 48 followed by the rule learner JRP and the KN classifier, however overall classifier accuracy was less than 70%. In addition, for different categories of students, such as excellent, average, bad etc., classifier performance differed. Of the student characteristics studied, those found to be significant in regard to predicting university performance were university admission score and number of failures (Kabakchieva, 2013).

Marquez-Vera et al. (2013) used real data records to predict the dropout rate of 670 high school students. To predict the final performance of the students, various classification methods were trialled. They proposed using a genetic programming model to generate accurate classification rules that were also all-inclusive. The study

found that by choosing the finest attributes and using cost sensitive classification and data balancing methods, improvements in prediction accuracy could be gleaned.

Data mining techniques were applied by Mohammed and M.EI-Halees (2012) to fifteen years of graduate student data from a college of science and technology to classify the data and identify associations, clusters and outliers. The students were graded as excellent, very good, good, and average. Results showed that when applying association rule to the dataset there is a 75% probability that a student with poor in matriculation results in secondary school will get an average grade at college. During the classification, rule induction and Naïve Bayes were applied, and rule induction showed a better accuracy of 75% compared to Naïve Bayes in predicting a student's grade. Then the K-means clustering was performed to cluster the students in to groups. Finally, outlier detection was used to detect all the outliers in the data, using a distance-based approach and a density-based approach. The purpose of the research was to improve the performance of graduate students.

Abdous, He and Yen performed an EDM study in which regression analysis was chosen to analyse live video streaming of students' online behaviour. Insights gained from the video analysis were compared with student academic performance for the 298 selected for the study. However, no correlation was established between the academic success of students and their online behaviour such as questions asked by students, chat messages and total login times (Abdous, He, Yen, 2012).

An analysis by Mustafa (2016) using data mining techniques on course evaluation questionnaire data found the most important variables that separate "satisfactory" and "not satisfactory" instructor performances based on student perception. A set of 2850 course evaluation scores was collected from a randomly selected department of Marmara University. Then, 1995 of those records were randomly selected to train the classifier model, leaving 855 records to be used as test data. The evaluation data comprised 26 variables, of which variable q26 is the course evaluation variable that is the focus of the value prediction task. A value of "satisfactory" assigned by a student means that the student found the course to be positive, whereas "not satisfactory" assumes a negative review. In this study seven classifier models, namely C5.0, CART, SVM, ANN-Q2H, ANN-Q3H, ANN-M, and DA were trialled. Performance measures of all the classifiers were found to be approximately 90% against the test data.

The Random tree and J48 classification methods were created by Mishra, Kumar and Gupta (2014) to develop a student performance prediction model. The model was based on the social and academic integration, and various emotional skills of the students. Third semester students' performance is taken in to consideration and it classifies as Below Average (BAVG), Average (AVG), Above Average (ABVG) and Excellent (EXCL). A sample of 215 instances was collected and 10-fold cross valida-

tion was performed. Random tree gives better accuracy (94.41%) when compared to J48(88.37%). From J48 and Random tree analysis, the researchers found that the students second semester result is key to influencing their third semester result. Overall, good academic performance in the second semester is a good indication of good performance in the third semester. Furthermore, the emotional attributes of the students that were found to most affect their performance were leadership and drive.

Koviac (2010) presented an EDM case study which focused on predicting the success of information system students from New Zealand's Open Polytechnic, based on their enrolment data. CHAID and CART algorithms were applied to students' enrolment data to construct two decision trees to classify the students into two groups: successful and unsuccessful. Prediction accuracy achieved using CHAID was 59.4% and using CART was 60.5% (Kovačić, 2010).

Goga, Kuyoro and Goga (2015) designed a tool to predict students grade by providing various parameters as input. A framework Models were developed using ten classifiers (OneR, Random Forest, ZeroR, random tree, Decision stump, REPTree, JRip, J48, PART and Decision table) and a multilayer perception learning algorithm by operating on WEKA. A framework is designed for an intelligent recommender system which recommends suitable action for improvement. The work is based on the background factors that predicts tertiary first year academic performance of the students. The data for the student obtained from Babcock University, Nigeria and the background factors were collected through in-depth interview. Mothers educational qualification, fathers' educational qualification, sponsor, family size, marital status of the parents, father's occupation, mother's occupation, and average family income are the discovered background factors. The study showed that Random tree outperforms other classifiers with an accuracy of 99.9% for 10-fold cross validation and 99.8% for holdout method

A classification model was developed by Qasem, Emad and Mustafa (2006) to discover and examine the main attribute that affects student performance in university courses with the aim to enhance the quality of university courses. The data were collected via a questionnaire directed at undergraduate students of a University in Jordan and 12 attributes were filtered out of 20 attributes and used for the analysis. The student grade was considered to be the class attribute. The Weka tool was used to apply three classification models, namely ID3, C4.5 and Naïve Bayes and they carried out both hold out and 10-fold cross validation. But the classification accuracy achieved was not very high for any of the three classifiers and hence, the researchers concluded that the sample and the attributes were not large enough or comprehensive enough to construct a satisfactory classification model.

Chapter 3

Process mining research

This chapter concentrates on the process mining research. The process mining approach is not just tied to an individual. It extends analytics beyond an individual's data to track the processes followed by that individual in attaining some defined objective. This chapter describes about the various techniques that are available also the implementation of the techniques in the process mining research and highlights about the various uses of process model. This chapter also explains about the tool which has been used for this research. The chapter summarizes with a review on related research works undertaken in the process mining area.

3.1 Introduction

EDM technique helps to identify the patterns among the student's performance in a test or a predicting students' performance in a test from large volumes of data acquired through the use of various information systems. Most of the data mining techniques focus on the simple pattern or data dependencies and not the entire educational process. This can be overcome by using process mining techniques which seek to discover and retrieve process-based data from information system event logs(Pechenizkiy et al. 2009). Commonly used event logs in the educational domain are student enrolment and registration procedures, course attendance by students, examination traces and e-learning activity logs(Awatef Hicheur Cairns et al. 2015).

Process mining (PM) is a new area of research that lies between machine learning and data mining on one hand, and process modelling and analysis on the other hand. PM extracts data from event logs and uses this information to identify, monitor and potentially improve processes(Van 2011). Various tools, and methods have been employed on event logs to discover process information and models. The nature of discovered models and the efficiency of discovery techniques differ according to such criteria as fitness, precision, generalization and simplicity

The vast majority of the IT frameworks used in practice are likely to keep some sort of record, in event logs, of the actions that occur during execution of the processes they support. The main advantage of PM methods is to gain access to objectively compiled data, i.e. they collect information about actual events and processes, directly from an event log of an organization.Information systems event logs are accessed and searched by PM techniques to identify, monitor and even improve processes. They can also detect bottle necks and ensure processes adhere to set procedures.

Also known as workflow mining(Bogarın, Cerezo, and Cristóbal Romero 2018; Trcka and Pechenizkiy 2009) ,in process mining, most of the work concentrates on the discovery of Petri net representations of workflows, i.e. a process model that defines the set order of events and activities involved in a process. Process models that broadly describe the events and processes extracted from event log data are generated from the raw event logs via these process mining methods(Bogarın, Cerezo, and Cristóbal Romero 2018; Reimann, Markauskaite, and Bannert 2014).

Figure 3.1: Positioning of main classification of process mining
(Source: Van 2011)

Event logs directly record and represent the process, step by step, therefore, process mining techniques mine these logs to identify and verify or potentially modify and improve process models(Munoz-Gama et al. 2014). Sometimes, these models are utilized to glean knowledge, to record or study the process, for instance, they may

be used in simulations to test proposed changes to the process. The models may also be employed to support the process, for instance, as part of a workflow management system. Regardless, a model that represents the actual process as closely as possible is useful to quantify the value addition of various embedded processes and to see if processes can be extended to other business areas

PM technology can be utilized in two ways. Firstly, PM can be utilized to mine a process model from the relevant event log where the event log is collated manually, and subsequently, the behaviour of the mined model will be compared with the relevant log. Secondly, instead of manually collating the event logs, the log can be recorded automatically by an information system and the PM can be used in a similar way(Wen, Wang, et al. 2010).

Process models are used for

Insight: help to view the process in different angle

Discussion: used for structure discussion

Verification: errors in the system or procedures can be find out by analyzing process models

Configuration: models can be utilized to configure a system.

Performance analysis: The factors involving response time, service levels, etc can be understand by utilizing techniques like simulation (Van 2011).

3.2 Event logs

Event logs are the primary object that any PM technique will interact with. An event log can be in the form of a spreadsheet, a database table, or multiple tables, or a simple text file, that contains a sequence of events and each event has a set of attributes (refer to Table 3.1). The events in an event log are ordered sequentially, with each row corresponding to one event. Normally, the event logs need to be transformed into a specific format such as XES (eXtensible Event Stream) or MXML (Mining eXtensible Markup Language) which can be used by a process mining tool(Van der Aalst 2016).

Event attributes from an event log that may be used in process mining include:

- Trace – process instance id of the event
- Activity – name of the action done in an event
- Time stamp – moment of the event completing,creating an order of events
- Resource – name of the resource originating or completing the event
- Data – data feature associated to the event.

Case id	activity	Start time	Finish time
1	A	2017-03-15 07:00:23	2017-03-15 07:12:10
2	A	2017-03-15 08:12:34	2017-03-15 09:10:02
1	B	2017-03-15 10:05:01	2017-03-15 10:45:05
3	A	2017-03-15 12:12:12	2017-03-15 12:45:01
2	C	2017-03-15 14:15:40	2017-03-15 14:59:45
3	B	2017-03-15 14:45:01	2017-03-15 14:59:45
1	D	2017-03-15 16:00:20	2017-03-15 16:45:45
2	D	2017-03-15 17:01:34	2017-03-15 17:56:03
3	D	2017-03-15 18:23:34	2017-03-15 20:23:34

Table 3.1: Sample event log

Figure 3.2: Various process mining techniques in terms of input or output
(Source: Rudnitckaia n.d.)

Of these five attributes, the first three constitute the minimal requirement for process mining.

Event logs can be utilized to conduct three types of process mining: discovery, conformance, and enhancement.

3.3 Process discovery

Process discovery is the first type in process mining, whereby a process model is build based entirely on an event log by capturing the behavioural aspects of the process which are revealed in the event log (Bogarın, Cerezo, and Cristóbal Romero 2018). A process tree is constructed from the event log and this tree can be directly transformed into a Petri net to explain the behaviour recorded in the log. Based on the discovery technique, process models can be modelled corresponding to many process modelling forms. Petri net, Fuzzy models, Business Process Modelling Notation(BPMN) are most commonly used in process discovery.

Process discovery can be break down in to four perspectives.

Control-flow perspective involves with aspects related to the process discovery. All the techniques that build process models from the observed behaviour described in the process data. This perspective aims to deliver a Petri net or some other notations that illustrates the control-flow of activities based on the flow of cases in a log. The attention is on the ordering of the activities and the dependencies between the activities(Van 2011; Aslan 2017).

The resource or organizational perspective involves aspects related to the resources and organizational structure of the business process. For instance, a technique which derives a social network from an event log is an example of a resource perspective.

Time perspective concerns with situation associated to the timing of the process events. Dotted chart, a technique that represents process events over time is an Example of the time perspective.

Data perspective is involved with aspects related to the properties of process instances or process events. When an event log has information about the timestamp of events, with respect to the time perspective, we can discover bottlenecks within the process, analyse the service times of activities and predict the remaining time for the running cases. An Example is the decision minor where the properties of the process events are taken in to consideration to characterize specific choices in the process model(Van 2011; Aslan 2017).

ProM has a plug-in that mines a Petri net directly using an Inductive Miner technique. The process model (refer to Figure 3.4) was generated from the event log. A Petri net consists of places denoted by circles and it contains token denoted by closed black circles. Furthermore, the squares with an inscription represent a transition, for example, an activity undertaken by the students. The squares without an inscription refer to invisible activities, there is no observable activity attached to it, instead mainly distribute token in the Petri net(Aalst, Bolt, and Zelst 2017). A transition moves the token from the input location to the output location and one token from each incoming arc is consumed by a transition which also yields one token on each outgoing arc. The choice split is the place that the choices are made, where one place is linked to multiple transitions and each transition is competing for a token, whichever transition fires first consumes the token, hence disabling the other transition, choice join where ever split a choice must join to synchronize again. Some behaviours are infrequent when compared to others.

The log-based ordering relations are utilized to discover patterns in the corresponding process models. Figure 3.3 shows the various process patterns that occur in the event log.

Figure 3.3 is explained as follows: (a) describes the sequence pattern when both activities A and B occur in sequence. XOR-split (b) occurs when there is a choice

((a)) Sequence pattern

((b)) XOR-split

((c)) XOR-join

((d)) AND-split

((e)) AND-join

Figure 3.3: Various process patterns
(Source: Van 2011)

between activities B and C after the activity A has occurred. Here A can be followed by B and C, but B will not be followed by C and vice versa. This is known by XOR-Split. After the occurrence of either B and C activities, if the activity D happens, this is referred as XOR-join shown in (c). Or, if after A, both B and C can be executed in parallel then this is known as AND-split (d). And if after a split there is a join, then itis referred as AND-join shown in (e). All these patterns combine together to form the process discovery model.

A simple example of a Petri net for the above event log is as follows(Figure 3.4).

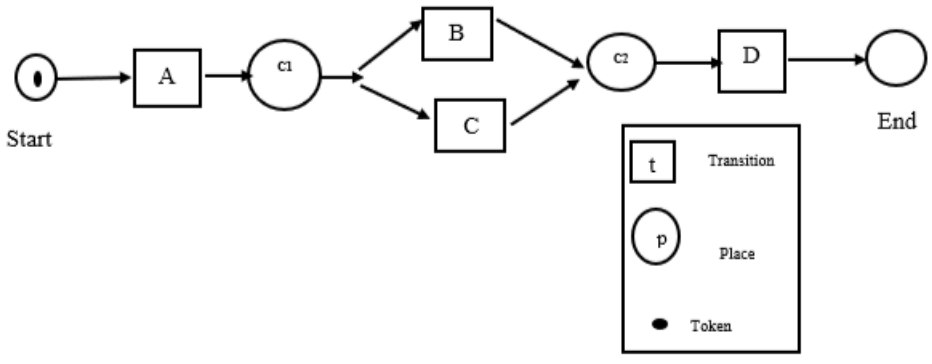


Figure 3.4: Process model in Petri net notation

Considering the above example, first the activity, A, has one input place and a token. Therefore, the corresponding activity, known as firing, is enabled and thus can occur. The transition triggered, during firing, consumes one token from each of its input places and yields one token for each of its output places. Thus, one token is produced for the output place C1. In parallel, the token from C1 enables both activities B and C. The activity B thoroughly removes token from C1 and disables activity C. Similarly, the occurrence of activity C disables activity B, i.e. there is a choice between the two activities. Lastly synchronization occurs before activity D is enabled. The process ends after activity D(Van 2011).

The main advantages of using this approach are

- All models discovered by process mining correspond to a simple workflow or Petri net system.
- The model can produce the traces in the log(Ghawi 2016).

3.4 Conformance checking

Conformance checking is done after the process discovery stage. It takes both an event log and a model as input (see Figure 3.5) and the process model is compared with the event log of the process that the model is describing(Aalst, Bolt, and Zelst 2017).

The aim of the conformance analysis is to check if the processes modelled actually do match the processes that were observed in the log. Thus, if the traces on the model were replayed, this would reveal any differences between the log and the model (Van der Aalst 2013b).Metrics like fitness (possible observed behaviour according to the model) and appropriateness (model typical of the observed behaviour) can be used to evaluate the conformance of the model.

Figure 3.5: Rediscovering process model
(Source: Van 2011)

The plug-in used for this analysis accepts a Petri Net and an event log and replays the log to compare the trace and the net. The fitness value between the trace and the net is calculated and used for further analysis.

To replay an event log on a process model, ProM used the plugin 'Replay a Log on Petri Net for Conformance Analysis'. The replay is needed to perform two tasks:

- To check for rediscoverability, i.e. that the model can recreate the event log from which it was created.
- To classify the unseen traces as fitting or not fitting with respect to the discovered (Ghawi 2016).

The two forms of conformance checking are log–model conformance checking and model–model conformance checking. In log–model conformance checking the model is compared against an event log. This type of conformance checking also provides a deep understanding of the real behaviour of an organization, by pointing to where in the model any deviation occurs and which traces deviate from the model(Leemans, Fahland, and Aalst 2016). In model–model conformance checking, the model is compared with the model of the system and can be used to check whether the process model has been adjusted to a design made before. In log–model conformance checking, the reality is considered to be represented by the event log, whereas in model–model conformance checking, a portrayal of the system is assumed to be available and to represent reality(Leemans, Fahland, and Aalst 2016). For this experiment, log–model conformance checking was conducted.

The four important dimensions used to compare the model and the log are fitness, simplicity, precision and generalization (see Figure 3.6). Fitness refers to how well the observed behaviour from the log file matches up with the process model. In addition, fitness detects any divergence between the specification of the process and logged events and activities. Perfect fitness of a model indicates that the model can replay all the traces in the log step by step. The best model is the simplest possible model that can describe the logged events and activities. A model which does not allow much leeway in behaviour is precise and is a model that does not allow any behaviour which is not related to the behavior seen in event logs. Generalization expresses the possibility that future logged behaviour will be able to be represented by the model and simplicity indicates the simplicity in a model to show its behaviour. It is a challenge to balance these four quality dimensions. For example, a model which is extremely simple is likely to lack fitness and precision, whereas a model with ideal fitness often lacks simplicity.

Fitness is measured by aligning traces in the event log to traces of the process model. Transition represents the move in the model and it should be labelled because

Figure 3.6: The four quality dimensions:fitness,simplicity, generalization,precision (Source: Rudnitckaia n.d.)

there could be multiple transition. If a move in the model cannot imitate by a move in the log or vice versa then a(>)(>)("no move") appears as shown below in the figure 3.8 (Van der Aalst 2013a).

For any process-mined model and corresponding log, the fitness value should be close to 1 which indicates a good fit(Wen, Aalst, et al. 2007). The top row indicates the movement in the log and the bottom row indicates the movement in the model. When there is a movement in the model and the corresponding move is not observed in the log, then it is called move on model only, and if is observed in the log and not the model, then it is called move on log only. Synchronous moves occur when the movement in the log and the model match, i.e. both movements occur simultaneously. To select the appropriate alignments cost is associated to undesirable moves and the alignment with lowest total costs will be selected.

Figure 3.7 is a simple example showing to capture the deviations from the model. Aligning traces with the process model .

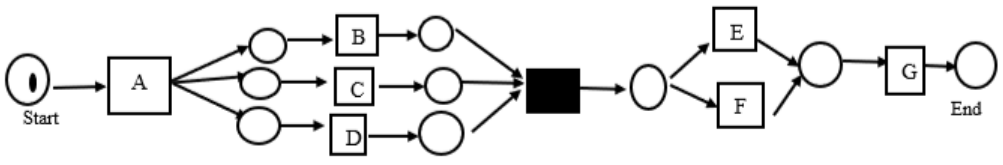


Figure 3.7: Aligning traces with the model

In the above model first, the activity A is executed. Activity A is fired synchronously by both the Trace and the Model and hence the name synchronous move, the same synchronous move was observed with activity B and C. Then the process model observes the activity D, but Trace didn't observe activity D. The

move of D in the model only and not in the Trace is called Move on model only. Next the activity E was fired by both the Trace and Model. The activity F was observed only by Trace and the Model remains the same at activity E, hence we see ¿¿ at the Model table. The move of F in the log only and not in the model is called Move on Log only. Finally, activity G is executed synchronously before the process gets over.

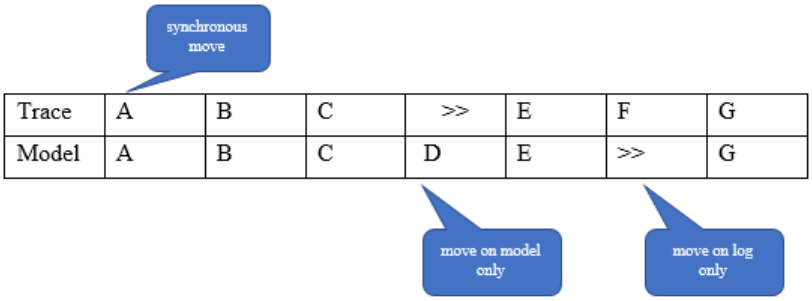


Figure 3.8: Event log and process model aligning

Generally, conformance checking is used for

- Improve or realign business processes, information systems or organizations.
- Repair models.
- Evaluate the process discovery algorithm.
- Connect the event log and process model and hence enable further analysis.
- Explore multiple options to find the optimal alignment.
- Allow flexible costs for activities and move types (e.g. a model move can be preferred over a log move if the cost is less)(Rudnitckaia n.d.).

3.5 Enhancement

Like conformance checking, the model enhancement process also considers both the process model and the event log as inputs. Enhancement involves refining a process model by making use of additional process information stored in the event log and this could mean repairing the model to better reflect observed behaviour. Say, for example, that two activities are modelled as occurring subsequently, yet they can occur in any order, then the model might be rectified to reflect this reality. Other option is to extend the model, whereby additional process information from the log

can be used to add a new perspective to the model thus extending the process model. For example, by consuming a time stamp from the event log in the "request for compensation" process, an extension can be made to the process model to highlight bottlenecks and report throughput times and frequencies(Van der Aalst 2016).

3.6 The PROM Framework

Various software tools which can implement PM are Celonis Process Mining, Disco, Perceptive Process Mining, ProM, ProM Lite, and RapidProM (Grigorova, Malysheva, and Bobrovskiy 2017). Figure 3.9 is an example of ProM framework.

The ProM Framework was the tool used for this experiment. Of the process mining tools available, the ProM framework(Van Dongen et al. 2005) is the most useful and powerful tool that is designed for process mining and discovery, and process analysis. Techniques ranging from process discovery, conformance analysis and model extension are supported by ProM, along with conversion tools and tools to import or export plug-ins and more.

A plug-in is a software application that can either be added to or removed from the framework without affecting functionality. A wide range of models, from a Petri Net to LTL formulas, and also tools can be loaded via the import plug-in feature. A mining plug-in is used for the process mining task, the results of which are stored as a frame. These results are analyzed via an analysis plug-in and/or converted into another format via a conversion plug-in, for example, to convert an EPC into a Petri net or vice versa. The ProM framework also features an event log filter plug-in which filters the more complex event log and enables only a subset of approved activities in an event log. ProM provides two ways to establish a relationship.

- Using a mining plug-in from the framework, process mining is used to create a Petri net from the event log. In such a case, a relationship is established between the mined transitions in the Petri net and the event classes in the log.

- The Petri net in combination with the event log is loaded into the framework. In this case, the user must specify the individual transitions in the Petri net that should be related to specific event classes in the event log(Hornix 2007).

The standard format to store and exchange event logs is MXML (Mining Extensible Markup Language) or XES (eXtensible Event Stream) and these formats are adopted in the ProM framework(Awatef Hicheur Cairns et al. 2015).

The main advantages of using the ProM framework for the development of PM techniques are outlined as follows:

Figure 3.9: ProM framework
(Source: Günther 2009)

- ProM framework presents a common application framework which facilitate the development of user interfaces for the mining techniques and enables an extensive set of functionalities to be accessible in one place.
- ProM is a plug-able framework; hence all the plugins can be merged in to the framework both from source and binary packages, considering restrictive or commercial extensions.
- The framework presents a wide range of model sort implementations, which can be utilized as input or output for plugins. Further, the framework presents default visualization for the given model types, e.g. a Petri net.
- ProM depends on the idea of a shared common object pool which can be accessed by each plugin in the framework. A plugin can be applied to any subset of objects in that pool as input data and can exchange the objects it creates back into the pool. This empowers plugins to use the functionality of other plugins, by being executed in sequence.

ProM framework architecture is displayed in Figure 3.10. The object pool is shown in the centre of the model which contain many objects from a variety of types(Günther 2009).

3.7 Goals of process mining in Educational domain

The basic idea of PM is detecting, monitoring and making improvements to the real process by discovering information and extracting knowledge from automatically

Figure 3.10: ProM framework overview architecture
(Source: Günther 2009)

updated event logs. This same approach can also be applied to solve problems in the educational domain.

The main goal of applying PM in education domain are to:

- Extract process-based knowledge from large educational event logs to generate process models which inform on key performance indicators or assist in setting up educational program design templates.
- Examine educational process models and their conformance with established educational program constraints, instructor's hypothesis and prerequisites.
- Upgrade educational process models with performance indicators such as bottlenecks, execution time, decision points etc.
- Personalize educational processes by means of suggesting of the best course units for a particular student or recommending specific learning methods to students based on their target skills, profiles or their preferences together with the automatic detection of possible prerequisite violations or any other relevant requirements or restrictions(Grigorova, Malysheva, and Bobrovskiy 2017).

3.8 Related works

3.8.1 Application of Process mining techniques on educational datamining

Pechenizkiy et al. (2009) proposed the use of various data mining techniques like process discovery, conformance checking and performance analysis techniques to examine student behaviour during the examination period. The data from two online multiple-choice examinations were used for the analysis. All possible information like grade, correctness, time spent answering the questions, time spent reading questions, whether a question was skipped and whether an answer was revised were collected from the two examinations. First, a dotted chart analysis was done, and it was found that mostly students answered the test questions in order and they correctly answered more questions in the first section of questions than in the last set of questions. Moreover, only a few students skipped questions after reading them. Then process discovery was performed and a conformance checking performance analysis was conducted and analysis revealed no mismatches between the answering pattern specified and the actual exam data. Performance of the answering process was done by using Petri net under performance analysis and it was found that 35% of the students had answered the first few questions correctly with more confidence. Almost all the students asked for feedback and checked their answers. The answering time was generally short and the students who answered the questions with high confidence spent more time reviewing the feedback.

In another study, Bogarín et al. (2014) applied two process mining techniques, namely conformance checking and performance analysis, to detect how prescribed workflows in a student registration model differed from the actual process instances. The data obtained consisted of an online course with 84 undergraduate students on the Moodle platform. The students were grouped firstly from data based on Moodle usage summary and the student's final marks were used. Then, to obtain more specific and accurate models of students' behaviour, the Moodle log data for each group of students was evaluated separately. Findings of the study revealed that the fitness of the specific models was greater than the fitness of the general model generated from the whole dataset(Bogarın, Cerezo, and Cristóbal Romero 2018).

Process mining techniques were used by Southavilay, Yacef and Calvo (2010) to analyse a collaborative writing process. The study looked into correlations between the writing process and the final document, specifically looking the quality and semantic features. A conformance analysis was also performed on a set of predefined pattern templates to extract pattern-driven education models from students' examination traces(Southavilay, Yacef, and Callvo 2010).

Tr ˇcka and Pechenizkiy (2009) developed a technique that depends on a set of pre-defined pattern templates which aims to merge the domain knowledge to EPM and to assist with interactive process mining. The outline structure helped the educators analyse the educational process in a principled way from the students' examination traces. This software prototype was built on a ProM framework and was developed for academic curriculum mining. This framework monitors the stream of educational curriculum in real time and automatically advises the students if prerequisites are not fulfilled(Trcka and Pechenizkiy 2009).

Chapter 4

Data sources

The datasets utilized for this project comes from two data sources: data collected from Xorro-Q (educational tool) which contained data about the activities taken by the students during the year 2016 and 2017 for a course conducted over one semester. In addition, the data about the student's assessment score for the year 2016 and 2017 was acquired from the course facilitator who taught the course. These two datasets were combined and used while performing the experiments. The final Exam score for the 2017 data was not available initially but was made available in November 2017 for this study.

A classification model was created using the 2016 dataset and the performance of the classifiers was tested using the 2017 dataset. Nearly 240 students from a University have participated. Briefly present an outline of the tool and its attributes. All these data were presented in MySQL. The tables which are required for the analysis were extracted using the SQL queries.

4.1 Dataset 1-Description.

Xorro-Q is a web-based audience interaction tool which enables synchronous and asynchronous interaction with the speaker(teacher)and the audience (students). The participants can use their own web enabled devices to connect with the facilitator. The facilitator uses Xorro-Q to ask the audience a wide range of questions and manages their participation (responses) using a simple dashboard. Question is a part of an activity which makes participant to respond in some way. Using Xorro-Q it is easy for the facilitator to find out whether the student is understanding the concepts or not by asking questions to the students while participating in the class. The questions can be posed spontaneously (also known as Realtime or synchronous session) where they are put forth during a live in-class session and involve the facilitator working with an audience in real time, or, questions can be posed asynchronously (or Self-Paced) where the participants can answer at any time and

at their own pace. Self-Paced questions are used mostly for self-practise or assignment activities. Xorro-Q can also give instant automated feedback and assessment to every participant. Moreover, Xorro-Q does not involve any downloads or plugins and the participants can access the activities through any browser. In Xorro-Q each facilitator will have a unique URL where the activities will be created. Participants will be able to reach activities being run by that facilitator or by scanning the QR code. The activities can be set in various ways, for Example enabling questions to be repeated thereby allowing students to re-visit the question again, or activities can be set to a minimum target level to encourage students to attain that level thereby helping students acquire confidence and familiarity with the fundamental concepts. Xorro-Q can be set to give feedback after every question or at the end of the activity or both to the participants. The facilitator can prepare Realtime activities comprising of a Batch or Question Batch in which a group of multiple questions are asked one at a time by the facilitator. Rather than asking each question, the facilitator may aggregate the questions into a Batch and then ask students to access the Batch.

Xorro-Q uses wide choice of question types. The question types can be either Multiple choice (having both single and multiple selection), text response, numeric response, hotspot image (single or multiple zone), labelling questions or likert scale questions. Some Examples of question type are given below

Single multiple-choice type where the students must select one answer.

Figure 4.1: Example of a multiple-choice type

Multichoice type Example where the students have to select more than one answer.

Figure 4.2: Example of multiple-choice type with more than one answer

Numeric type example.

Figure 4.3: Example of a numeric type

An overview of database schema extracted from Xorro-Q shown inFig:4.4. Next in Figure 4.5, each entity has had been described in more detail to show their attributes and datatypes. Also, the relationship between the entities is made clearer with relationship names and their corresponding multiplicities.

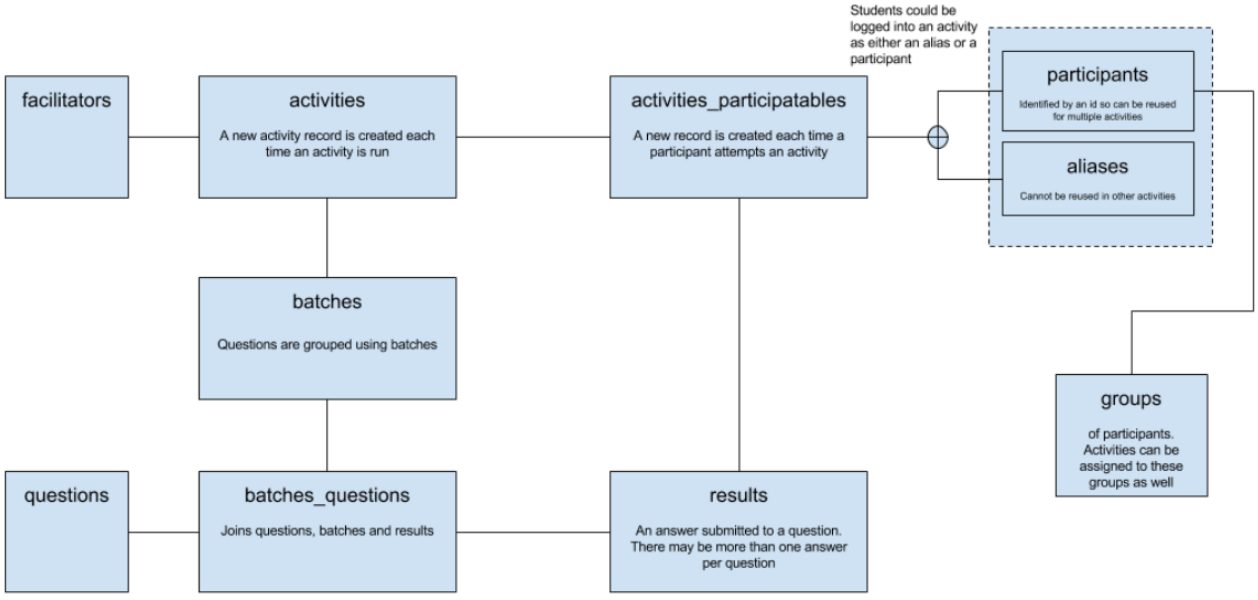


Figure 4.4: Database schema of Xorro-Q

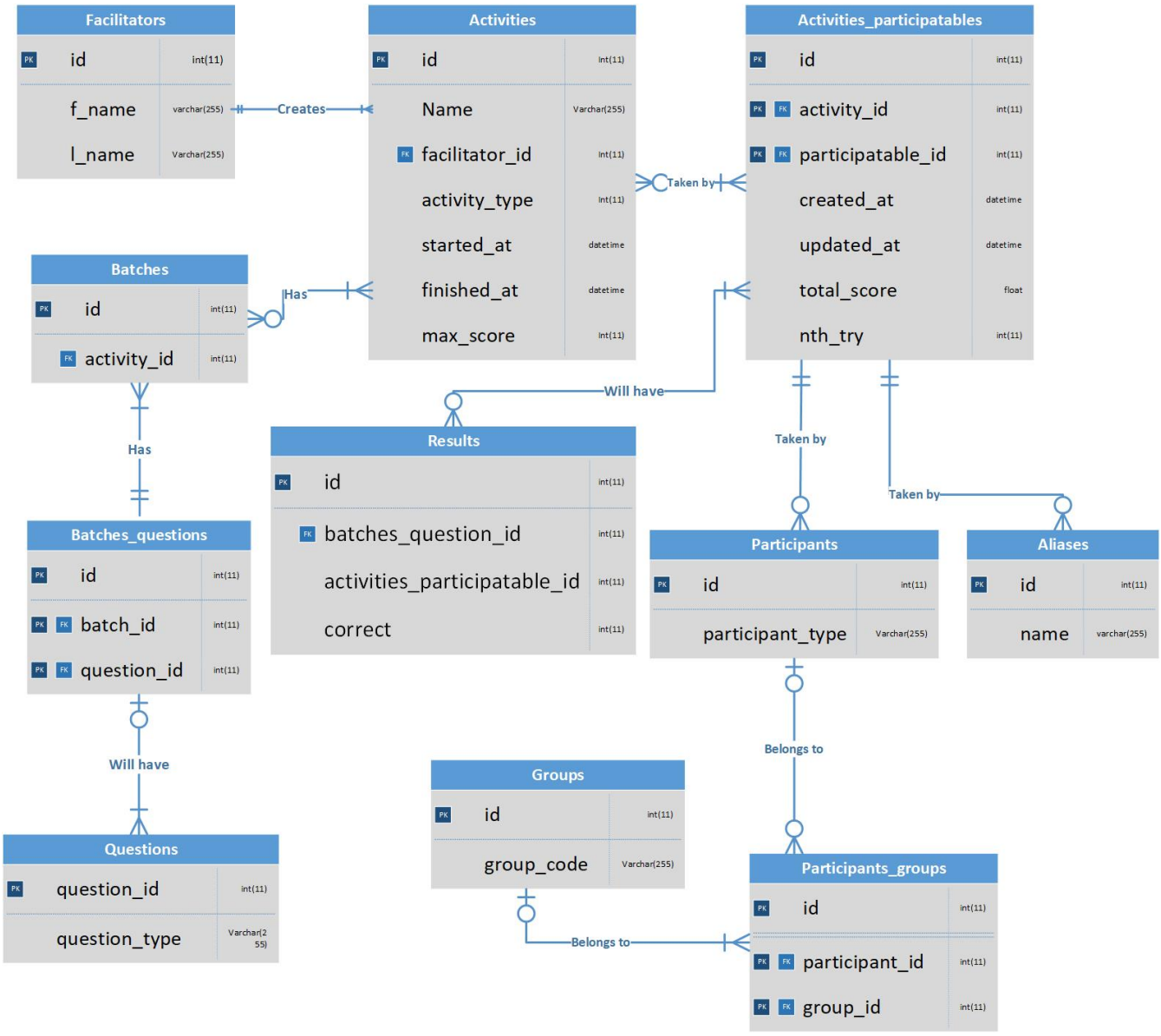


Figure 4.5: Database schema of Xorro-Q with the attributes

The following table describes the relationship between the entities.

Entity	Relationship	Connectivity	Entity
Facilitator	1:m	creates	Activities
Activities	0:m	taken by	Activities_participatables
Activities	0:m	has	Batches
Batches	1:m	has	Batches_questions
Questions	1:m	Will have	Batches_questions
Participants	0:m	Taken by	Activities_participatables
Aliases	0:m	Taken by	Activities_participatables
Participants	0:m	belong to	Participants_groups
Groups	1:m	belong to	Participants_groups
Results	1:m	Will have	Activities_participatables

Table 4.1: Entity relationship

While the above schema shows many tables, the tables which were mainly used for the analysis purposes are the Activities table, Activities participatable, Result, Groups and Final grades. These have been explained next.

Activities.xls

An activity is the set of questions posted by the facilitators and which have been answered by the students. The activity table lists all the activities run by Xorro-Q. This includes the activity id which is unique, the name of the activity, maximum score which varies among the activities, the starting and the finishing time of the activities and finally the activity type. There are two activity types namely Real-time and Self-paced. Realtime activities are those activities which happened live in the class so there is no chance of re-doing the activities. Typically, Realtime activities are done to boost the student's participation during the teaching session and to keep their attention focussed on the subject being taught. Next comes the Self-paced activities which can be endeavoured again and again by the students to complete a minimum threshold. A target score set for 70% was set for this project. Attempting an activity many times will help the students to progressively increment their score which will increase their motivation. Self-paced activities do not involve the facilitator to be logged in. Self-paced activities have been further divided into mandatory and voluntary activities. Mandatory are the compulsory activities have to be taken while voluntary activities depend upon the student's inclination and may or may not be taken.

Activities participatables.xls

An activities participatable has the details about all the participatable who has participated in the activities. It has the participatable id which is the id of the

participant who has taken the activities. The participant can use an Alias or use their id. For this study's analysis, only participant's details without the Alias attempts have taken into consideration because if a participatable logs in under an alias name, we will not be able to extract their reports over multiple sessions. Also, if these Alias group participants exit the activity and later re-enter the activity, they will be recognized as a different participant.so participants group alone taken. Activities participatable has the total score which are the scores taken by the students for an activity and the number of attempts made by the students.

Results.xls

Results is a set of data relating questions asked to the participants and the participant answers for this specific activity. A Result can refer to the records of all the questions asked to the students. The answers can be either correct, partially correct or entirely wrong. A result is not created if there were no response from the students. Thus, Results provides a log or report of what questions were asked and what answers were given by which participants. Results stores the scoring information for each individual question.

Final grades.xls

In addition to these database tables there is Final grades tables. This comprises of the previous course grade (which is a prerequisite course requirement for entry into this course), Test 1 score (or the first test taken by the students in the fifth week of the semester), Test2 (or the second test taken by the students which happened on 10th week) and the final Exam score which is the target variable which should to be predicted. The activities were done by the students over a 12- week semester period and the number of participants were reduced to 5 after the 10th week i.e., after the Test2 was scheduled. Hence, the 12th week activities have not been taken into consideration for the analysis purposes. Both the Realtime and Voluntary activities were done in the second and third week by the students. The first week activities were taken by Alias participatable, so these activities have not been taken into consideration. Therefore, only the activities which started from second week have been used for the analysis. Moreover, it is worth mentioning that during the fifth and seventh week, activities comprised of survey questions which were asked to the students and for which no score was allocated; hence these weeks too have been left from the analysis.

The graph (Figure 4.6) displays the number of activities per week. As can be seen from the graph more activities were done initially, but as the weeks progressed over the semester, the number of activities taken by the participants had also become

less and the least was on the 8th week where only one activity was taken by the students. The Test 1 happened after week 4 so all the week 4 data has been clubbed with the Test 1 score and this has been considered for week 5. Similarly, the Test 2 happened after week 9 so the week 9 data and the test 2 score have been together considered for week 10.

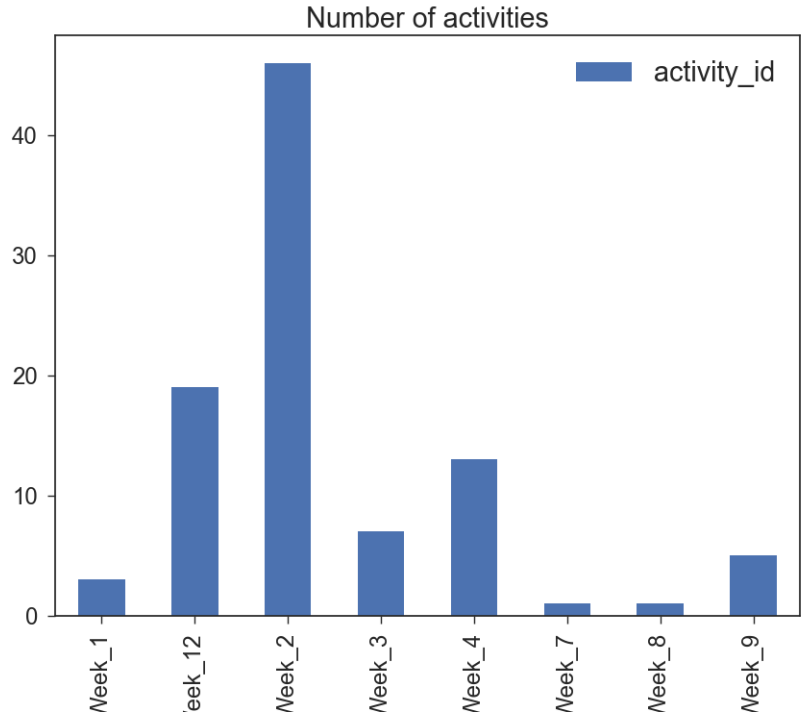


Figure 4.6: Activities over weeks

4.1.1 Dataset features- Xorro-Q activities

This dataset contains all the attributes which have been acquired from Xorro-Q database (refer Table 4.2). The data obtained from Xorro-Q were spread more than a few worksheets in Microsoft Excel. There were many activities taken by the students so taking score of individual activities and utilizing it for the analysis is not feasible hence for the analysis purposes the average of the activities score are used. Subsequently the students can take numerous attempts to accomplish the minimum threshold of 70%, therefore student's minimum and maximum scores in these activities have additionally be considered as one of the attributes for the prediction. It should be noted this is applicable only with the Self-paced activities. The number of attempts also varies for the students some students might achieve their target score of 70% in first attempt and for some students it may go up to 5 or even more attempts. Using this reasoning, the number of attempts is a better attribute for prediction purposes, so the average number of tries for every student

have also been considered. The Test1 and Test2 score play an important part in students' final Exam score, and this study found that the Test2 score especially showed a good relationship with the final Exam score. Therefore, an average of Test1 and Test2 score were taken. The pre-requisite course grade has also considered.

The activities taken by the students for the year 2016 and 2017 looks almost similar but the activity structure looks different. Most of the 2017 activities were divided into 2to 3 smaller activities just to make it easy for the students and to prevent them from taking too many attempts. For the year 2016 the number of Realtime activities taken by the students were 5 but for the year 2017 it was only one activity. Moreover, number of participants for the year 2017 were much more when compared to 2016.

S.no	Feature description
1	Average lowest score taken in the activities
2	Average highest score taken in the activities
3	Mean score of the activities
4	Average number of attempts to do an activity
5	Total number of answers correct
6	Previous course grade score
7	Test1 score
8	Test2 score
9	Average of Test1 and Test2

Table 4.2: General features description obtained from Xorro-Q database

4.2 Dataset2 features - Process Mining

Process mining features was used for the dataset 2. A dataset has been generated using event logs of weekly Xorro-Q activities. Generating event log from Xorro-Q dataset is the first task in process mining. For doing so the Xorro-Q data has been mapped to an event log. Two important things need to be consider while mapping. These are: What constitutes an event and what makes a case (sequence of events). For each case the data available about the student is stored i.e. the activities taken by the student and the corresponding time stamp for that activity.

The Table 4.3 below displays an sample of an event log created from Xorro-Q. The participant id is considered as the case for this experiment,name is the activity name which are taken by the students.Started and finished are the starting and finishing time of each activity by the students.

The students have next been categorized as High risk, Low risk and Medium risk students and the event logs were created separately for these three categories. The event log which is in CSV format are transformed in to XES format an event log

format reinforced by tools like ProM.

	participant_id	name	Started	Finished
0	11986	Beam_Can1_P_001 - Improved	2016-07-25 08:54:00	2016-07-25 09:04:00
1	11986	Beam_Can1_P_003 - Improved	2016-08-02 21:56:00	2016-08-02 22:00:00
2	11986	Beam_Can1_U_001 - Improved	2016-08-02 09:17:00	2016-08-02 09:39:00
3	11986	Beam_Can1_U_007 - Improved	2016-08-03 00:09:00	2016-08-03 00:19:00
4	11986	Beam_Sim1_P_001 - Improved	2016-07-25 08:35:00	2016-07-25 08:46:00

Table 4.3: Event log generated from process mining

Techniques

The most commonly used techniques in process mining are discovery, conformance checking, dotted chart analysis, and social network analysis. Discovery and conformance checking are the two methods which are used for this project.

4.2.1 Process discovery using Inductive miner

After generating event log, the Inductive miner technique is utilized to observe the process model for the event logs. With Inductive minor technique, the process model for the above-mentioned event logs was generated for all the weeks stating from week 2 to week 10 because one of the research question is to find out if there is any improvement in accuracy of predicting students' performance over the weeks. The figure 4.7 below shows the process model of Low risk students which illustrates all the activities taken up by the Low risk students. Similarly, the process model was obtained for High risk and Medium risk students too. The model describes that most of the activities are in parallel i.e, can be implemented in any arbitrary order.

After generating a process model for High risk, Low risk and Medium risk students replay an event log on these models is performed This is achieved by generating conformance checking.

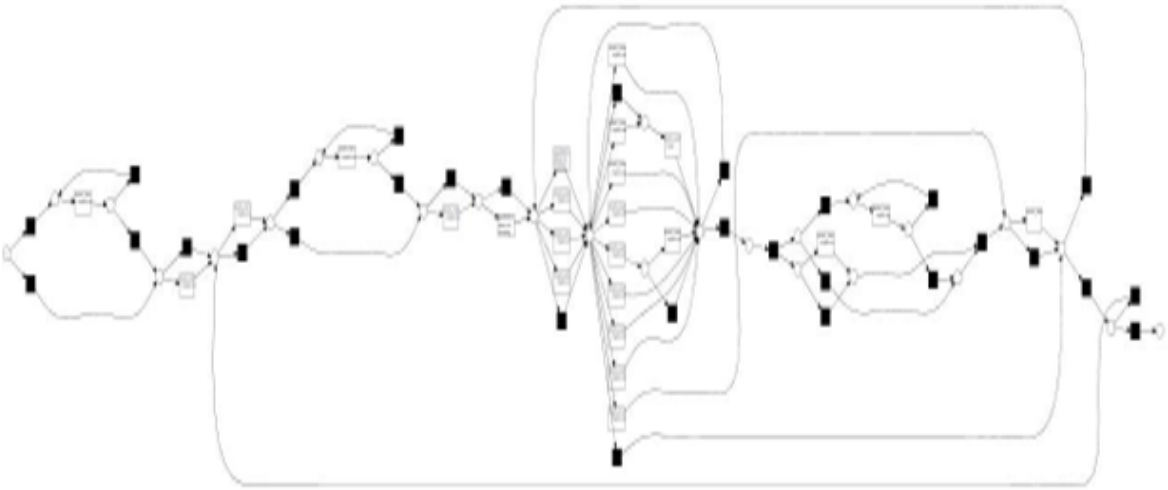


Figure 4.7: Process model of a Low risk students

4.2.2 Conformance Checking

Conformance checking was performed using model representing High risk, Low risk and Medium risk student's weekly activities. The event log of the students was replayed using the above-mentioned model to find the relationship between the event and the model and to analyse the deviation of students from the modelled behaviour. The result will display log model alignments for each case as shown in Figure 4.8. Fitness score obtained from conformance testing was combined along with the general features obtained from dataset1 and used for predictions. In most of the cases the model and the trace are not conforming with respect to each other.

Figure 4.8: Replay results

The coloured picture of the trace alignment can be used to determine fitness which is illustrated in Figure 4.9.

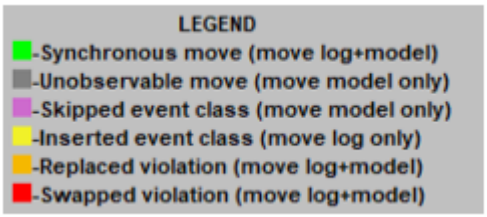


Figure 4.9: Alignment legend

If there are green alignments, then the trace is classified as fitting. The grey block indicates the invisible activities. Lastly yellow corresponds to the non-observant event i.e the activity occurs in the trace of event, but the model does not describe about the occurrence of the activity and it is classified as non-fitting.

4.2.3 Dataset2 features

The fitness score obtained from conformance checking was combined along with the general features and utilized for the prediction analysis.

As stated before if a move in the model cannot imitate the move in the log, then move in model occurs. If a move in the log cannot imitate the move in the model, then move in log occurs. Trace fitness indicates how well the event log(data) align with the model.

	participatable_id	Move-Log Fitness-medtrain	Trace Fitness-medtrain	Move-Model Fitness-medtrain
0	12031	0.48	0.48	1.00
1	11901	0.45	0.46	1.00
2	11855	0.46	0.46	0.99
3	11887	0.47	0.47	1.00
4	11950	0.49	0.49	1.00

Table 4.4: Various fitness scores obtained from conformance checking

The fitness score obtained from conformance checking was quite low the reason for such an observation is that since the student can do the activities on their own time, so each student will not follow the same path. Moreover, since there are some activities which are not mandatory, so it is obvious that some students might leave some of those activities. There was less synchronous move between the trace and the model. All these may be the causes for low fitness score.

Chapter 5

Research approach and methodology

5.1 Preparing data for mining

The overall procedure of data mining comprises: collecting the data, pre-processing the data; executing data mining algorithms and producing the outcomes. This project also followed the same data mining procedures. Though, the Xorro-Q data are not intended for data mining analysis, it was important to do pre-processing and select the data attributes sensibly. The general procedure of data mining is revealed in the Figure 5.1 below. Starting from pre-processing stage every case in the figure is explained in the subsequent sections.

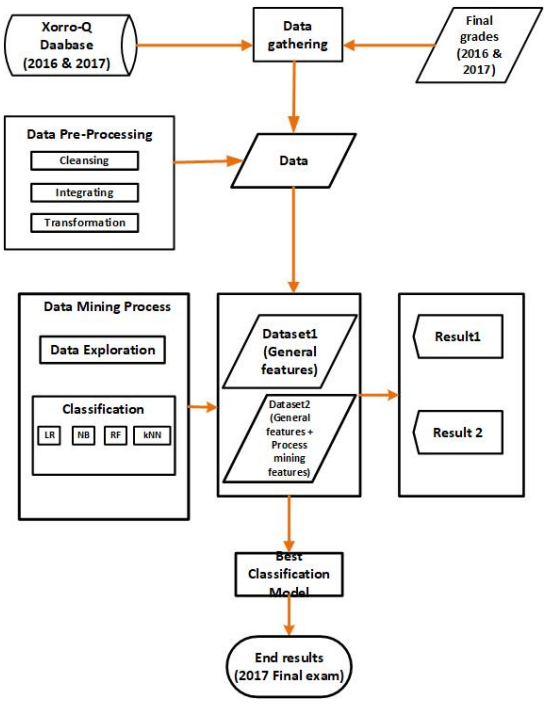


Figure 5.1: Various process mining stages

5.2 Data pre-processing

Before implementing the EDM methods the pre-processing of the data source was performed. Data in raw form were not generally the finest for investigation, and particularly not for predictive analysis. Real-world data is known to be noisy, incomplete and may contain discrepancies (Varun Kumar and Chadha 2012). The reasons to be noisy may be the data is not continuously collected, a mistake is made in data entry, duplication in data entries or violation of data constraints and much more. Consequently, it is important to change the data to a relevant form for solving the academic problem. This comprises deciding what data to gather, concentrating on the questions to be replied, ensuring that the data line up with the questions posed.

Moreover, huge volume of data from various sources with different formats are stored on the educational environments. Further huge volume of attributes and domain variables with information about each student which can delay prediction algorithm to attain an interesting conclusion in a lesser period. Hence it is significant to reduce the data appropriately for better analysis. Furthermore, if a dataset is highly imbalanced, that is, if number of occurrences from one class is considerably higher than the number of occurrences from other classes, the prediction algorithm tends to emphasis on learning from classes with higher number of occurrences. So, the data need to be pre-processed or organized and altered to obtain the finest mineable frame (Cristobal Romero and Ventura 2013).

Data preparation is essential since different predictive data mining techniques act differently relying on the pre-processing and transformational methods(Nsofor 2006). Data pre-processing is utilized to alter the raw data into a clean dataset. The attributes are reduced during this process and the number of attributes become less when compared to the number of attributes in the original dataset.

The different steps involved in data pre-processing are data cleaning, data integration, data transformation and data reductions (refer Figure 5.2). For achieving better results in machine learning data pre-processing is done because certain machine learning algorithm do not support null values. In this study, the dataset used (i.e., Xorro-Q dataset) had many missing values and therefore required data pre-processing.

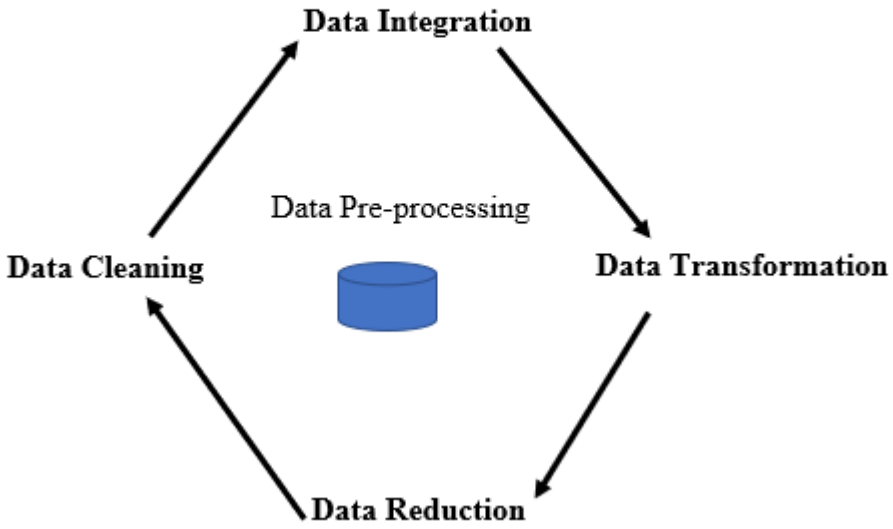


Figure 5.2: Various steps in data pre-processing

5.2.1 Data Cleaning

Once the data were separated from the database, the subsequent step was to clean them of inconsistencies, errors and noises beforehand the data were prepared to be mined. Data cleaning is the first step which is implemented in data pre-processing. It is a basic fact that incorrect or inconsistent data can prompt false conclusion and henceforth wrong assumption and decisions. Subsequently, high-quality data needs to pass an arrangement of accuracy, integrity, uniformity, consistency and uniqueness. Data cleaning tries to fill in missing values and correct irregularities in the data(Anwar and N. Ahmed 2011). The question in this research was on how to handle the missing data. These missing values can be handled by either ignoring the missing values or by filling the missing values by the mean, median or mode of the observed given values. In this project, the missing values have been filled by zero instead of omitting or filling it with the mean value; the reasoning being that some of the activities were not mandatory, so the students had not attempted these activities. Therefore, it made more sense to fill the missing values with zeroes rather than with any other value.

5.2.2 Data integration

Data integration involves incorporating or combining the data from different tables or databases into a meaningful data format(Han, Pei, and Kamber 2011). Data mining process frequently comprise regaining and analyzing several data attributes that are scatter over various tables or databases. As mentioned earlier, the data for this study originated from two sources: (1) the students' activity data collected from Xorro-Q and (2) students' assessments from spreadsheets acquired from the course facilitator. Hence, the data had to be merged into a summarization table which consisted of all required features for conducting the data mining researches.

5.2.3 Data Transformation

Data transformation was done to change the raw data into a specified format according to the need of the model. Normalization and aggregation under data transformation are additional pre-processing measures that contribute towards the success of the mining procedure undertaken. Normalization is done to convert the numerical data in to a specified range so that the scaling of the data can be performed. Aggregation is assembling the data to a form which help in ease of the handling of large amounts of data.

5.2.4 Data reduction

The attributes that do not deliver sufficient knowledge inputs applicable to the mining process must be eradicated. This could be achieved by choosing the attributes related to data mining. Many attributes have been derived for the analysis purposes like minimum/maximum slope, number of activities passed by the student, the maximum score in a particular activity, etc. However, the increase in feature space leads to difficulties for supervised learning; as a result, a high number of features can prompt a decrease in classification accuracy. Therefore, to select the best attributes for prediction purposes, correlation is used and is an important pre-processing step for feature selection and reduction. Correlation refers to the strength of association between the two variables. For this project correlation coefficient was utilized to find the correlation between the attributes with the target attribute. The correlation equation was applied to the data and there were some attributes which showed positive correlation with the targeted attributes and helped in increasing the prediction accuracy so only those attributes were selected for the analysis purposes.The attributes which are not positively correlated does not showed any improvement in prediction accuracy and so they were not used for the analysis. In this manner the number of attributes were reduced and only the attributes which are positively

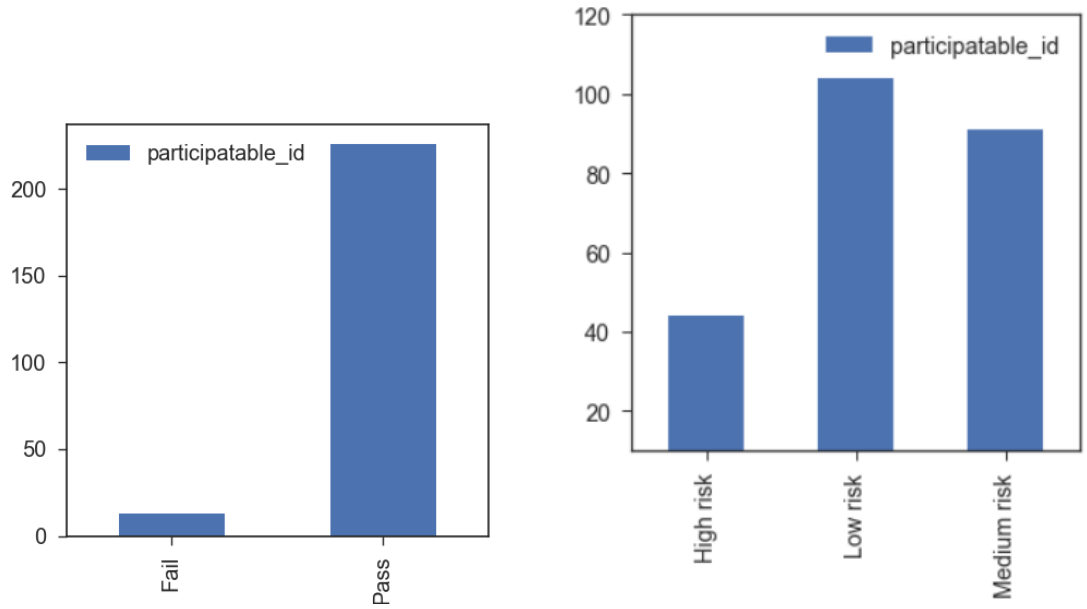
correlated with the target attribute were selected.

5.3 Datasets Construction

From the list of the features (refer to chapter 4) datasets were built for testing the hypotheses. Since the dataset is highly imbalanced it was decided to use the Exam marks (target variable) in a categorical form.

5.3.1 Numerical data with categorical variable

It is very hard to predict the numerical value in data mining especially with imbalanced data. Hence, a dataset of numerical value with nominal class label was formed.The dataset provided for the study was imbalanced with only 19 students having failed out of the total 230 students (shown in figure 5.3(a)), hence it was decided to categorize the students into three groups The lower bound of '55'(<=55) was chosen to represent the students at High risk of failing the final Exam, a high bound of '75'(>=75) was chosen for students who are at Low risk in failing the Exam and a medium band between 55 to 75(>55) and (<75) was selected to characterize the majority students who filled the gap between the upper bound and lower bound (refer Figure 5.3(b)).



((a)) Pass& fail ((b)) High risk, Low risk & Medium risk

Figure 5.3: Grouping Students

The categorization of class label is more helpful if the results are associated to the pedagogical information. The outcome of the analysis can be better presented to instructors who while may not have any data mining background, but they can

relate to pedagogic aspects emerging from the data analysis. Therefore, numerical dataset with categorical class was utilized for analyses purposes.

5.4 Dataset Attributes

It is significant to know about the data and represent it in a meaningful way. In the educational environment, for instance data could assist instructors to obtain a pictorial representation of each student's knowledge pattern and help course administrators analyse the students' usage actions or identify course activities which are worthwhile(Goyal and Vohra 2012).

Data exploration helps in examination of the data to better understand the qualities of the data. It can be done through visualization techniques. Data visualization (e.g., scatter, histogram) is a useful method for the identification of patterns, relationships and missing or exceptional values. In this study correlation were done using scatter plot to find about associations and to measure whether one data attribute is significantly associated to others.

Comparison of percent score of activities for each week of the semester using boxplot are shown next.

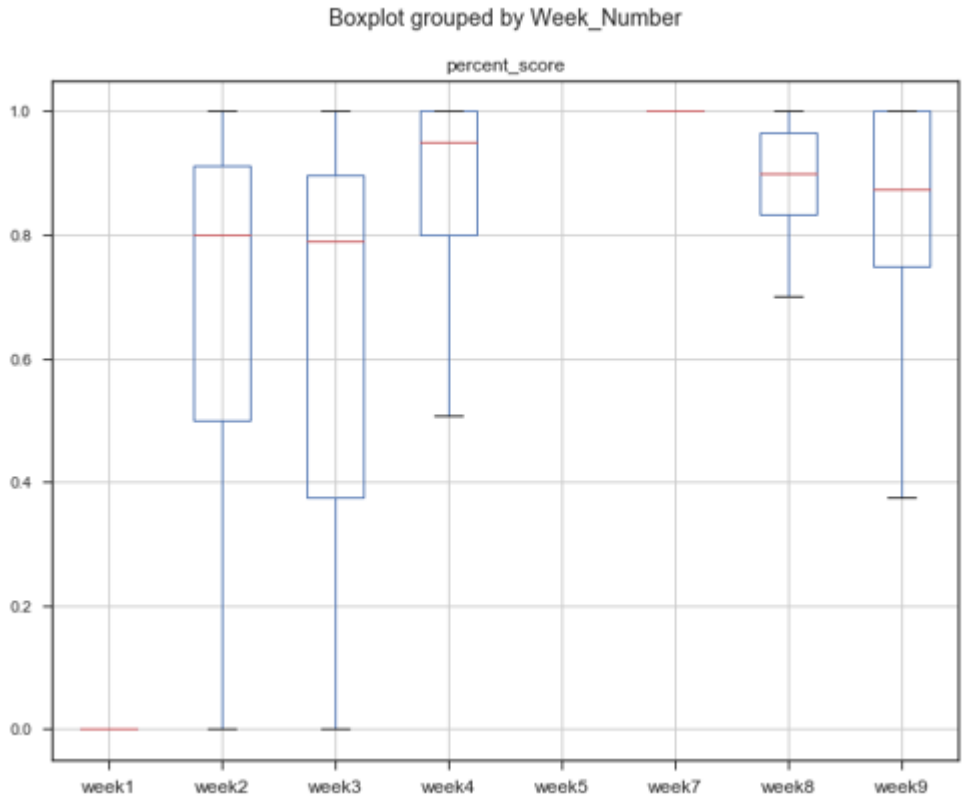


Figure 5.4: Xorro-Q activities scores by students over weeks

The box plot is a standardized method of exhibiting the spreading of data based

on five number summaries: minimum, first quartile, median, third quartile and maximum. The section inside the rectangle shows the median and "whiskers" above and below the box demonstrate the positions of minimum and maximum. The whiskers tell essentially the spread of all of the data. Boxplot was drawn to check the comparison of scores over weeks to find out any increase in scores taken by students over weeks or not. Since the week1 was taken by the Alias participants those scores are not taken in to consideration for analysis purpose. As observed from the boxplot above (Fig 5.4), there is improvement in scores over the weeks, but the improvement is not a gradual progress, rather there is a sudden increase in average scores in week 4 from week3, but after week 4 there is a decay in scores over the subsequent weeks. Week 5 and week 7 as said earlier were the weeks when the survey questions were asked to the students and student participation in those weeks does not carry any marks.

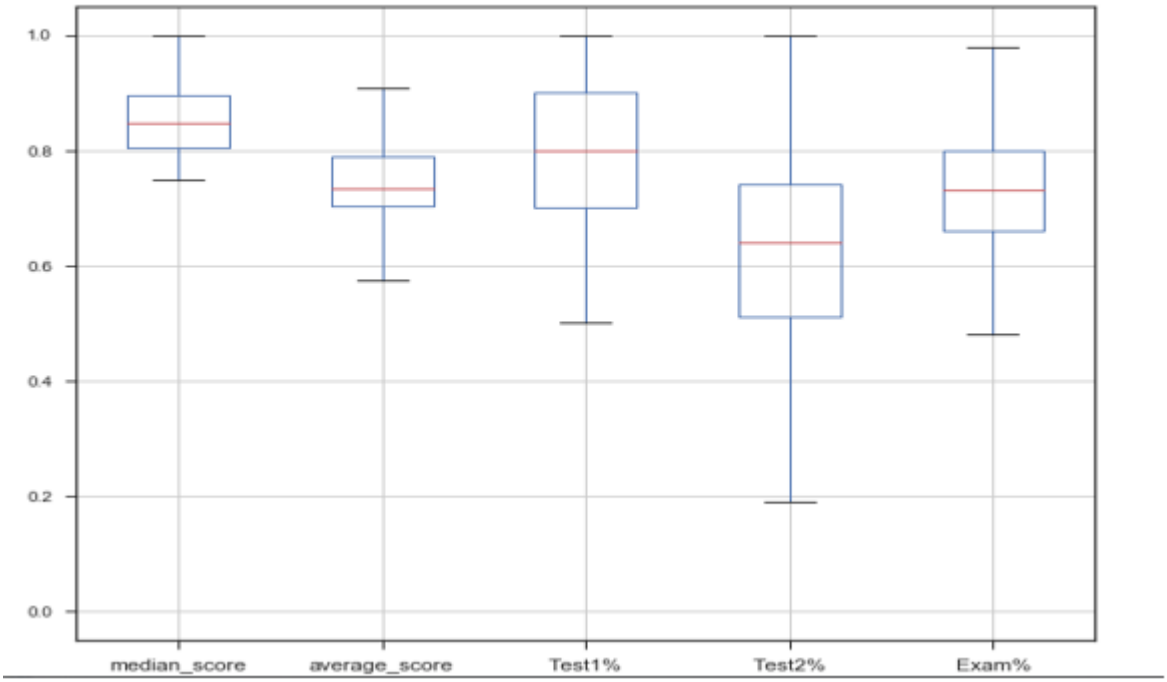


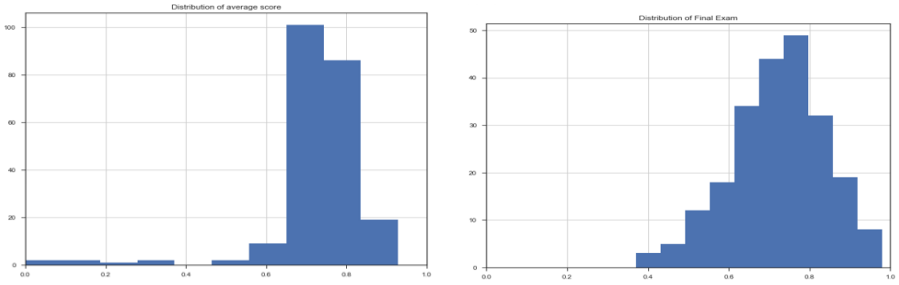
Figure 5.5: Comparison of various scores

Boxplot was drawn for all the scores to see the variance in scores. Test 2 score has significant variance which is between .2 to 1 when compared to all the other scores. Median score is generally on the higher end. Variance on Exam is between .5 and 1. The variance of average activity score was small when compared to all the other scores.

Distribution of average activities score and exam score.

In the experiment to find out the frequency distribution of activities score and the Exam score, a histogram plot is used. Histogram are the recurrence of score

existences in a continuous data set that has been divided into classes called bins. The frequency distribution displays how frequently each different value in a set of data arises.



((a)) Xorro-Q activities scores ((b)) Final exam scores

Figure 5.6: Distribution of scores

The first plot describes the frequency distribution of average Xorro-Q activities score. As can been seen in (Figure 5.6(a)), more number of students have attained scores between 70 to 80. Nearly hundred students are in that range; the reason for such high numbers are because it was a compulsory course requirement for which students had to achieve 70% score in each activity. The number of students who got more than 80% and who have less than 70% are fewer in ratio.

The second plot (Figure 5.6(b)) describes the frequency distribution of Exam score. The distribution here is normal. The decay in scores is also gradual when compared to the activities score.

A scatter plot is one of the most powerful graphical method for deciding whether there appears to be any relationship or pattern between two numerical attributes. The scatter plot is a useful technique to investigate the likelihood of correlation relationships. If one attribute implies the other, then the two attributes are said to be correlated (Han, Pei, and Kamber 2011). Scatter plot was drawn to find the relationship between the average activity score, Test1, Test2 and prior course grade with the Exam score.

Relationship between the average activity score, Test1, Test2 and prior course grade with Exam is taken into consideration because the target is to predict the students' Exam achievement. Therefore, it is significant to find the relationship of these scores with the Exam score. Hence scatter plot has been drawn individually for all the scores with the Exam scores to find whether there is a linear relationship with the Exam score or not.

If there is linear relationship, the plotted points pattern will slope from lower left to upper right side, since as the x value increases correspondingly the y value should also increase i.e., the graph pattern should resemble a 45° angle. Such a linear relationship was noticed between Test2 and the Exam (see Fig 5.7)

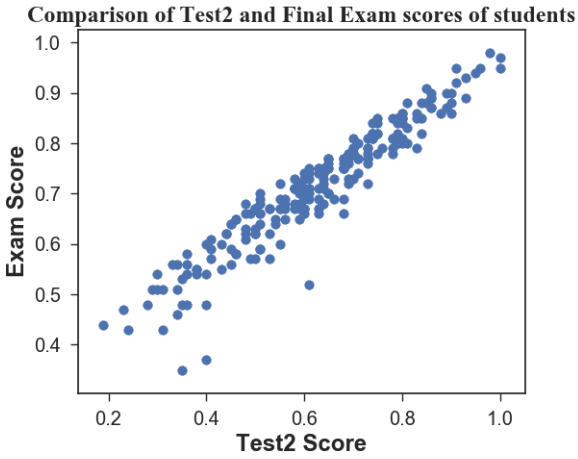


Figure 5.7: Comparison of Test2 score with final Exam score

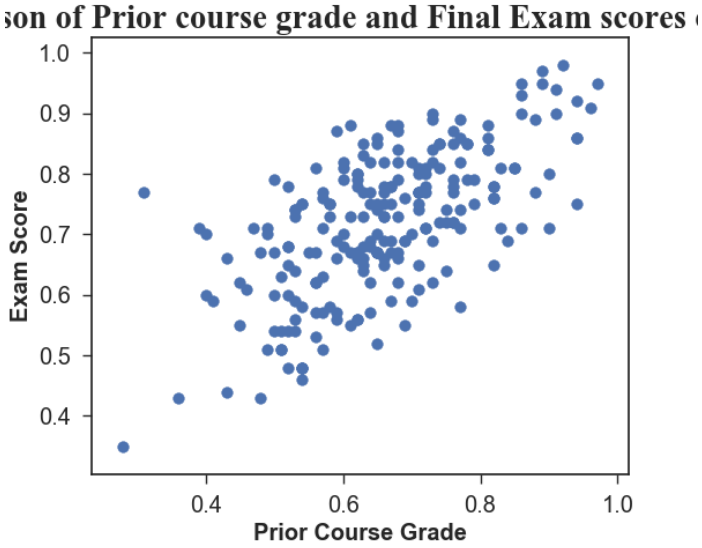


Figure 5.8: Comparison of prior course grade with Exam score.

No such correlation relationship was found between the average activity score and the Exam (Fig 5.8), or even with the prior course grade (Fig 5.9). As we delve deep into the data to produce graphical displays, we are provided with valuable insights on student learning behaviours.

5.5 Machine Learning Training Procedures

It is essential to justify the stability of a machine learning model. The process of validation is deciding whether the results generated are acceptable as a description of the data (Gupta, 2017). Generally, after training the model on a dataset, an error estimation of the model is done. This indicates how good the model performs on the data utilized to train it, but the possibility exists that the model is underfitting

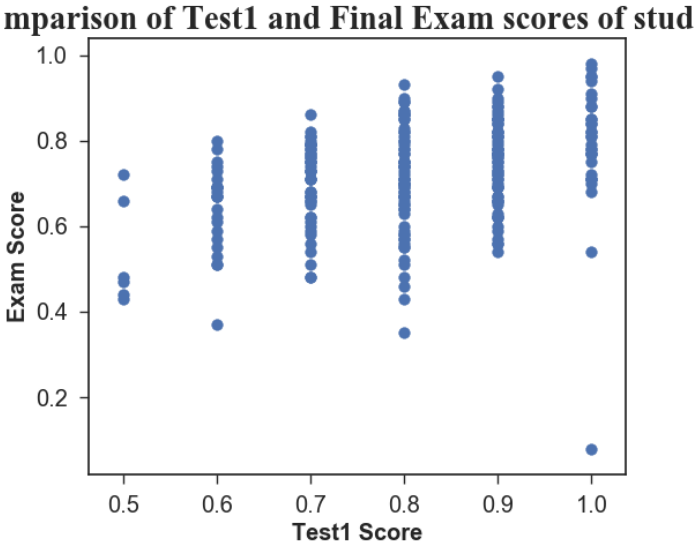


Figure 5.9: Comparison of Test1 score with Exam score

or overfitting the data. Thus, this evaluation technique does not give an indication as to how well the model will fit an as yet unseen dataset in terms of accuracy of its predictions(James et al. 2013). This can be overcome by using the model evaluation which is an integral part of model development process. While assessing machine learning models, the validation step assists in discovering the finest parameters for the model and prevents it from getting overfitted.

Specific data mining methods are tested to the dataset after it is pre-processed. The data mining methods may be based on either prediction or description. For this project concentrating on classification under predictive modelling (supervised learning), the task is of constructing a model for the target variables. The objective of the classification is to establish a model that lowers the error among the predicted and the true values of the target variable.

Two datasets are required to build a classification model, where one dataset is used for training and the other dataset is used for testing. This results in an evaluation of how the model will accomplish on forthcoming data like the training and the testing data. Different methods utilized to divide the data sample into a training dataset and a test dataset include:

- The hold out method
- Cross validation
- The leave-one-out approach.

The important goal of using model evaluation are

- To predict the performance of the model from the existing data by an algorithm.

- Secondly, the performance of various algorithm is compared and thus, the finest algorithm appropriate for the data will be selected (Yadav and Shukla 2016).

For this experiment, the holdout and cross validation methods were carried out along with ensemble methods which are discussed next.

5.5.1 K-fold cross validation

Cross validation is a technique for assessing the performance of the predictive model by splitting the dataset into a training set to train the model and a testing dataset to validate the model. In this technique, both the training and the testing dataset must traverse in consecutive rounds to make sure each data point can be validated (Refaeilzadeh, Tang, and H. Liu 2016). K-fold is the basic form of cross validation. In k-fold cross-validation, initially the data is separated into K segments of similar size or almost equal which is also called a fold. Consequently, k iterations of training and validation are done such that within each iteration a various segment of the data is retained for validation while the remaining k-1 segments are utilized for learning. In cross validation, for each iteration one or more learning algorithms use k-1 segments of data to learn one or more models and consequently the learned models make predictions about the data in the validation segment. The performance of each learning algorithm on each fold can be track by checking the accuracy. Since k samples of accuracy are available for each algorithm, averaging can be used to aggregate measures from these samples. By default, the value of K is set to 10, however, it is not a strict rule, and K can be set to any value.

For the k-fold cross validation the default 10- folds cross validation test was chosen. The 10- fold cross validation performs by separating the training data in to 10 segments of same sizes of folds. Out of 10 folds, the one-fold was used for testing and the rest remained utilized for the training. From the training data a new model was constructed, and the left-out testing data was used to evaluate the obtained model. After completing the process for the first test fold, a new subset was chosen for the testing and the rest subset was utilized for the training. This procedure was continued till entire folds have been used for testing. In Python Scikit- learn a random split training and testing sets can be calculated using the train_test_split helper function. The 10-folds cross validation method was applied to the four-chosen classification algorithm namely LR, KNN, RF and NB.

5.5.2 Hold-out methods

In hold-out validation the data is separated in to two non-overlapping parts and these parts are utilized for training and testing set and this consists of only one

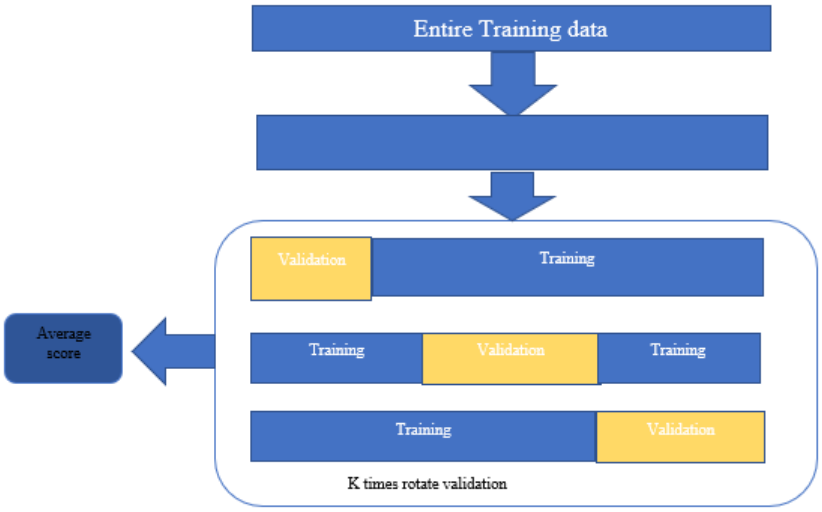


Figure 5.10: K-fold cross validation

iteration. The hold out part is the part which is used for testing and the model is learnt using the rest fragment of the data. Thus, the dataset is split into the training dataset and the test dataset. The training set is utilized to train the model and then the trained model is utilized to make predictions on unobserved data. The percentages of data being held out for testing can vary for hold-out method. Suppose if 90% of the data are being used to learn the model and 10% are left for testing then there are chances of getting over-fitting because it is not distributed appropriately and varies immensely for 10% data. The total time engaged for learning the model is comparatively smaller in hold-out method (Yadav Shukla,2016). In general cross-validation comprise shuffling the order of the instances and hence is not a suitable method for spliting the data into training and testing for the prediction problem. But with hold-out, the training test split is more suitable because the order of the occurrences is maintained (Rory Thabtah, 2017).

Initially the 2016 data was selected, and the hold-out method was carried for that. Here the entire dataset was separated into training and testing.75% of the dataset was utilized for the training process and the rest 25% was left out for the testing purpose. The model was created with the training dataset using the four-mentioned classification algorithm and the trained model was utilized to predict on the test data to test the performance of the classifiers. Then the 2017 data was used to validate the performance of the classifiers. For this the entire 2016 was utilized to train the model and the 2017 data was utilized to test the performance of the classifiers.

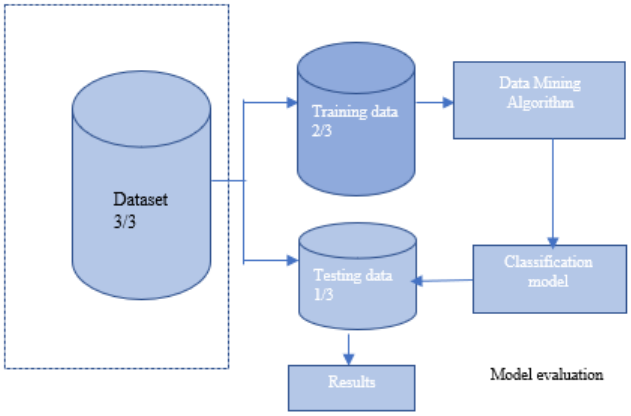


Figure 5.11: Hold-out method

Ensemble methods

Ensemble methods, or classifier combination methods, combine the predictions of several classifiers into a single learning model. Numerous models are combined to produce improved results as compared to results from a single prediction model(Demir 2010). Several classifier models are trained, and the outcomes obtained are generally united over a voting or averaging process. The basic idea of ensemble methods is to develop a linear combination of some model fitting process, rather utilizing a single fit of the method(Gentle, Härdle, and Mori 2012).

The three most popular methods for merging the predictions from different models are

1. Bagging
2. Boosting
3. Voting.

Bagging

Bagging or Bootstrap Aggregation includes taking various samples from the training dataset and training a model for each sample. Bootstrap aggregation generally implies building repetitions of the training sets of the same size as the original set with some instances occurring more than once and some not occurring at all. Each replicated training set generates a classifier model. The final output predication is averaged across the predictions of all the sub-models. Bagging utilizes bootstrap sampling to acquire the data subsets for training the base learners. Bagging utilizes voting for classification and averaging for regression for aggregating the output of the base learners. Bootstrap aggregation can be used to reduce the variance for those algorithms which have high variance(Sumana and Santhanam 2015). Bagging is extremely beneficial for huge, high dimensional dataset difficulties where discovering a good model in a single step is not possible because of the difficulty

and scale of the problem (Bühlmann and Yu 2000).

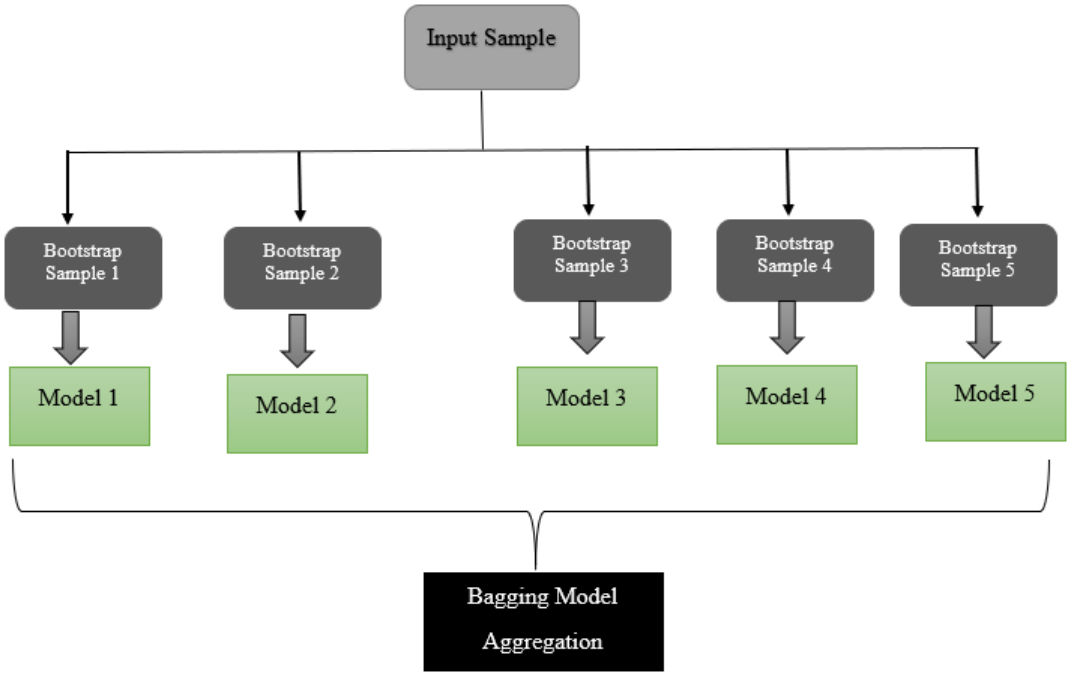


Figure 5.12: Bagging model

Bagging works as follows: Initially T bootstrap samples Lt, t=1, 2,....T are taken from the original learning sample L. Secondly, a base learner is applied to each of the bootstrap samples bt(Lt) and the classifier models f(bt(Lt)) are constructed.

$$\sum_{t=1}^{T} bT(L) = \frac{\sum_{m=1}^{M} f(bt(Lt))}{M}$$

Each sample model in the standard bagging procedure receives equal weight.

Voting

Voting is used for classification and it is one of the simplest ways to combine the predictions from multiple machine algorithms. It works by first creating two or more models from the training dataset. Each model creates a prediction for every test instance and the ultimate yield prediction is the one that obtains the maximum number of votes. (A voting classifier can then be used to combine the results from the models and average the prediction results of sub models.) Predictions for every model were created and stored in a matrix called predictions where every column holds predictions from one model. The idea behind the voting classifier is to balance out individual model weaknesses(scikit-learn 2007). This method was used for this experiment because while predicting the student performance in the initial weeks, the accuracy the classifiers was less than expected, so this method was trialled with

the aim to achieve better results, but it didn't make much difference.

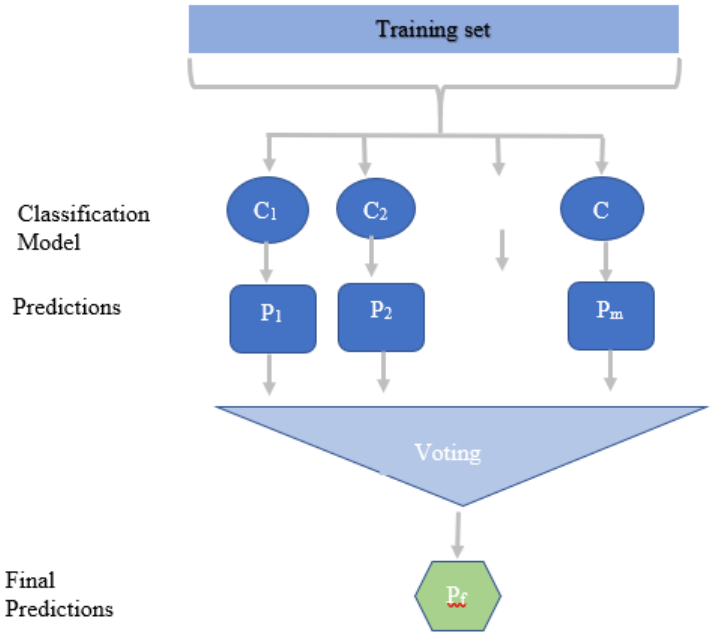


Figure 5.13: Voting

Boosting

Boosting ensemble algorithm makes a series of models that seek to rectify the mistakes of the models before them in the sequence. Once the model is made, it makes predictions which may be weighted by their demonstrated accuracy and the results are combined to make a final output prediction. To build a boosting technique a series of classifiers are produced by running the base algorithm repetitively by altering the distribution of the training set depending on the performance of the already formed classifier. Originally, the weights are allocated evenly to all the instances before learning the first classifier and the weights of the inaccurately classified instances are increased after every iteration and those of which are grouped accurately are reduced. With the goal that the following classifier focuses on incorrectly classified instances. Lastly, the weighted predictions of every classifier model are joined by voting for classification and averaging for regression to obtain a single composite classifier(Sumana and Santhanam 2015).

Model Evaluation

After designing the classification model, the subsequent step is to find how effective the model is based on metrics and datasets. The performance of the classification model can be measured by confusion matrix. The confusion matrix comprises in-

formation regarding the actual and predicted classifications (Márquez-Vera et al. 2013). The evaluation is performed by associating the predicted values with the actual values. The four possible prediction outcomes are the true positive (TP) and true negatives (TN) occurs when an object of a class 'yes' is predicted correctly as 'yes', and if the object of a class 'no' is wrongly predicted as 'yes' then it is called false positive (FP) and false negative (FN).With these four possible predictions, various performance metrics can be used to gauge the quality of the classification results. They are

- Classification accuracy is the number of accurate predictions divided by the total number of predictions.
- Precision is the fraction of significant instances among the retrieved instances, i.e., a measure of classifier's exactness (Powers 2011).

$$Precision = \frac{T_p}{T_p + F_p}$$

- Recall is the fraction of significant instances retrieved over the total of retrieved instances, i.e. a measure of the classifier's correctness(Powers 2011).

$$Recall = \frac{T_p}{T_p + F_n}$$

In multi-class predictions, the confusion matrix can be extended to a matrix with rows (actual class) and columns (predicted class) where each element shows the number of test Examples. A confusion matrix is important in analysing the results because it helps to find out where the classifier went wrong and got confused in predicting the label. Table 6 shows a sample of a confusion matrix for a three-class problem.

		Predicted		
		A	B	C
Actual	A	A (predicted as A)	(A predicted as B)	(A predicted as C)
	B	B (B predicted as A)	(B predicted as B)	(B predicted as C)
	C	C (C predicted as A)	(C predicted as B)	(C predicted as C)

Table 5.1: Confusion matrix for a multiple classes

To predict classifier performance, the accuracy determined above alone might be misleading if the dataset is highly imbalanced because a model can predict the value of the majority class for all predictions and attain a high classification accuracy(Jason 2016). Thus, correctly predicting the positive outcomes is not sufficient,

and a predictive model should comprise a blend of both successful positive predictions and successful negative predictions. In such circumstances, it is better to use F-measures which consider both the precision and recall and is the harmonic means of precision and recall(Pojon 2017). As the dataset has a majority of pass students compared to fail students, F-measures is taken into consideration.

$$Fmeasures = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Chapter 6

Results

6.1 Results

The aim of this research is to predict students' academic performance using Xorro-Q. The prediction model was created using python language, which has built-in functions suitable for this approach, and which generates the required outcome for estimating and refining the results of predictions. Python has emerged as an excellent tool for scientific computing tasks, that includes, the analysis and visualization of large datasets that are generally utilized for machine learning applications. Additionally, Python has libraries for data loading, visualization, and statistics. One of the benefit of Python is the capability to interrelate directly with the code, by means of tools like Jupyter Notebook, which is a collaborating environment for running code in the browser and also a remarkable tool for exploratory data analysis(Müller and Guido 2016).

6.1.1 Research questions 1: Is it possible to predict students' final course grades and outcomes based on data gathered through a synchronous and asynchronous in-class participation technology (Xorro-Q)? If so, which data mining algorithms provide most accurate predictions, and how early can we reliably predict a student's final course outcome?

To answer this research question, prediction was done on four chosen machine learning algorithms namely: LR, Knn, RF and NB using the k fold cross validation and hold out method. Hold-out method delivers better accuracy when compared to k fold cross validation, so it was decided to perform the experiment using former, where the dataset is divided into training and testing sets. After defining the testing and training sets, the next task is to construct the model individually using the training datasets with the above-mentioned algorithms. Developing a model using python is an easy process, which mostly contains defining the dependent and independent variable. Then the test data was applied to the created model. Output of this process

is a confusion matrix which holds the predicted values and the actual values. The same procedure applicable to all the classifiers, are shown below. First kNN classifier was tested and for this experiment scaling has been done before applying kNN. The reason for scaling is that the predictor variable may have significantly different ranges and that needs to be mitigated,so that certain features will not dominate the algorithm. Moreover, all the features must be unit-independent, which is not dependent on the scale of the measurements involved. The figure below shows the confusion matrix and the classification report.

KNeighbors Classifier (): This is the classifier function for kNN in sklean and the fundamental function for executing the algorithms. The significant parameters are:

- n_neighbors: It carries the value of K, required to pass, and should be a whole number. Different values were chosen and tested with kNN. It was found that when kNN=5, it gives better accuracy. Hence, this value was selected for carrying out the experiment

- metric: The distance metric to use for the tree. The distance metric used in this research was minkowsk with p=2 which is equivalent to the standard Euclidean distance

- Weights: It accepts a string value. The Weight function utilized in prediction can caries values like 'uniform' or 'distance' or any user defined function. Uniform weights were chosen for this research.

  - 'uniform' weight used if every point in the neighbourhood are weighted equally. Standard value for weights taken as 'uniform'

  - 'distance' weight meant for assigning nearer neighbours- more weight and distant neighbours-fewer weight, i.e., weight focuses by the converse of their distance.

  - user defined function: The user defined function is utilized while creating custom weight values, acquires distance values and yields an array of weights.

- algorithm: This indicates algorithm which ought to be utilized to compute the nearest neighbours, can use value such as 'auto', 'ball_tree', 'kd_tree', brute' and an optional parameter. 'auto' algorithm was used for this research.

  - 'ball_tree', 'kd_tree' are used to execute ball tree algorithm. These are special kind of data structures for space partitioning.

  - 'brute' is utilized to execute brute-force search algorithm.

– ‘auto’ is utilized to assign control to the system. By using ‘auto’, it automatically chooses the best algorithm as indicated by values of training data.fit ()

- data.fit (): A fit method is utilized to fit the model. It is passed with two parameters:X and Y. For training data fitting on kNN algorithm, this wants to call.
  - X: It comprises of training data with features.
  - Y: It comprises of training data with labels.predict(): It predicts class labels for the data assigned as its parameters.

If an array of features data is passed as parameters, then an array of labels is specified as output. The output labels of test data can be compared with actuals and accuracy can be calculated from confusion matrix. The figure 6.1 shows the confusion matrix and the classification report for kNN classifier.

```
('Accuracy', 0.6875)
Confusion Matrix
[[ 4  1  2]
 [ 1 41 15]
 [ 8  8 32]]
             precision    recall  f1-score   support

  High risk       0.31      0.57      0.40         7
   Low risk       0.82      0.72      0.77        57
Medium risk       0.65      0.67      0.66        48

avg / total       0.72      0.69      0.70       112
```

Figure 6.1: Confusion matrix of a kNN classifier

The second model was created using Naïve Bayes. GaussianNB (): This function implements the Gaussian Naive Bayes algorithm for classification in sklearn. The probability of the features is expected to be Gaussian:

NB implementations typically don’t have parameters for fine tuning. The procedure of model building and evaluation is same as with the previous model where the model was created using training data and tested using the test data. The accuracy and the f measures obtained was 85% which can be seen in the figure 6.2.

```
('Accuracy', 0.8571428571428571)
Confusion Matrix
[[10  0  6]
 [ 0 45  2]
 [ 3  5 41]]
             precision    recall  f1-score   support

  High risk       0.77      0.62      0.69        16
   Low risk       0.90      0.96      0.93        47
Medium risk       0.84      0.84      0.84        49

avg / total       0.85      0.86      0.85       112
```

Figure 6.2: Confusion matrix of a NB classifier

Next the model was built and evaluated using Logistic regression. LR (): This function used was sklearn. Some of the key parameters used as follows.

- Penalty: Used to specify the norm used in the penalization. L2 norm was used for this research due to its simplicity and stability
- multi_class: One versus all (ovr) option was chosen due to its computational simplicity and explainability
- Solver: liblinear solver was used because one vs all framework was used, the data size was not large.

```
('Accuracy', 0.7410714285714286)
Confusion Matrix
[[ 0  0  0]
 [ 0 41  7]
 [13  9 42]]
             precision    recall  f1-score   support

  High risk       0.00      0.00      0.00         0
   Low risk       0.82      0.85      0.84        48
Medium risk       0.86      0.66      0.74        64

avg / total       0.84      0.74      0.78       112
```

Figure 6.3: Confusion matrix of a LR classifier

The final model which was tried was Random Forest. RF Classifier(): This function under ensemble techniques of sklearn was used. Default settings were used.

```
('Accuracy', 0.8660714285714286)
Confusion Matrix
[[ 9  0  4]
 [ 0 43  0]
 [ 4  7 45]]
             precision    recall  f1-score   support

  High risk       0.69      0.69      0.69        13
   Low risk       0.86      1.00      0.92        43
Medium risk       0.92      0.80      0.86        56

avg / total       0.87      0.87      0.86       112
```

Figure 6.4: Confusion matrix of a RF classifier

From the classification experiment the best accuracy was achieved with RF model with 84%. From the outcome, the accuracy of the model was promising.The Real-time and the voluntary activities was done only for two weeks so the accuracy might probably be increased if these activities was done for the rest of the weeks too.

To find the early prediction of student's outcome, the data's starting from week 2 was used for prediction analysis. Further, while considering the week 3 data all the data from week 2 was also included and so on until week 10 where the Test2 was held. There was a slight improvement in accuracy over the weeks but while considering week 10 there was a notable difference in accuracy when compared to the other weeks. This is because the Test2 (which happened in week 10) was also added to the week 10 dataset, and as has been mentioned earlier, the Test 2 score is linearly related with the Exam score. RF outperforms all the other classifiers with 84% accuracy(with out adding process mining features) which is shown in the figure 6.5.

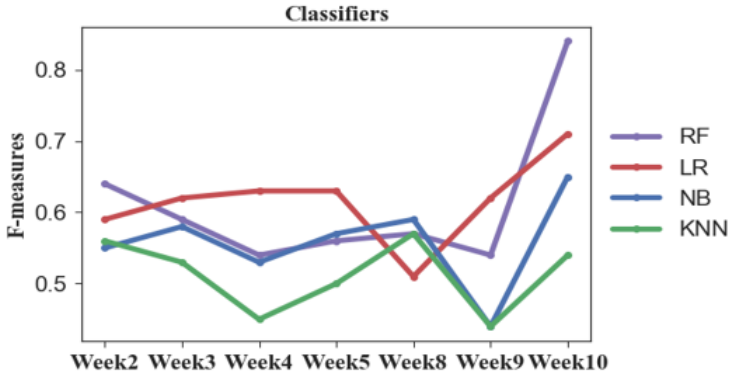


Figure 6.5: Comparative results of a various classifier

• 6.1.2 Research question2: Is it possible to improve the predictive accuracy of machine learning algorithms which use features extracted from in-class participation technology (Xorro-Q), by combining with features extracted from process mining? If in-class participation technologies (such as Xorro-Q) improve machine learning outcomes by generating valuable features, is there evidence in the captured datasets that indicate that there are benefits for students to using these technologies? Did students who performed poorly in the prior course, perform better in the course which used an in-class participation technology (Xorro-Q)? And, for those who performed better, was the extent to which they performed better, related in any way to their activity on Xorro-Q?

After predicting the students' performance using general features obtained from Xorro-Q, it was decided to implement process mining features to find out whether there is an increase in accuracy by integrating the process mining features. To answer this question, hold-out method was carried out. Firstly, in order to make the event logs the whole dataset was divided into three categories as High risk, Low risk and Medium risk students. Then the entire High risk data was split equally in to 50% for training and 50% for testing. Next the divided training data is taken and 75% from training data is used for training the model and the event log was created for them. The same procedure was followed for the remaining two category students. Then all the testing data (High risk, Low risk and Medium risk) are clubbed and event log was generated for them and also the event log was created for training data (High risk, Low risk and Medium risk) obtained from 50% of the entire data. This procedure was done so as not to leave any data unused. Using inductive miner technique process model was created for High risk, Low risk and Medium risk students. Then all the testing and training event logs was replayed on these models to see the deviation of students from the modelled behavior. The fitness score obtained from this model was used along with the general features and used for the predictions. When the fitness score was used there was an improvement in accuracy, but it was not a significant improvement. Table 6.1 shows results of classifiers using general features and process mining features.

The line graph shows the F-measures of various classifiers with and without process mining features along with general features.

Weeks	LR	LRP	KNN	KNNP	NB	NBP	RF	RFP
Week2	.59	.60	.56	.54	.55	.49	.64	.59
Week3	.62	.68	.53	.48	.58	.55	.59	.65
Week4	.63	.55	.45	.57	.53	.56	.54	.60
Week5	.63	.57	.50	.57	.57	.56	.56	.60
Week8	.51	.63	..57	.54	.59	.57	.57	.66
Week9	.62	.56	.44	.56	.44	.54	.54	.60
Week10	.71	.62	.54	.61	.65	.84	.84	.87

Table 6.1: F-measures of classification algorithm with standard features and process mining features

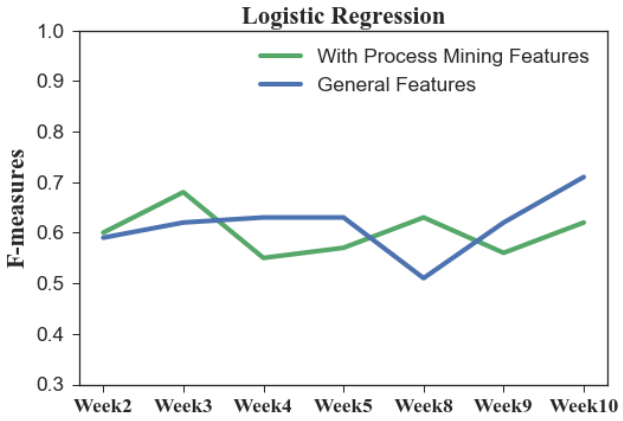


Figure 6.6: Comparative results of LR classifier

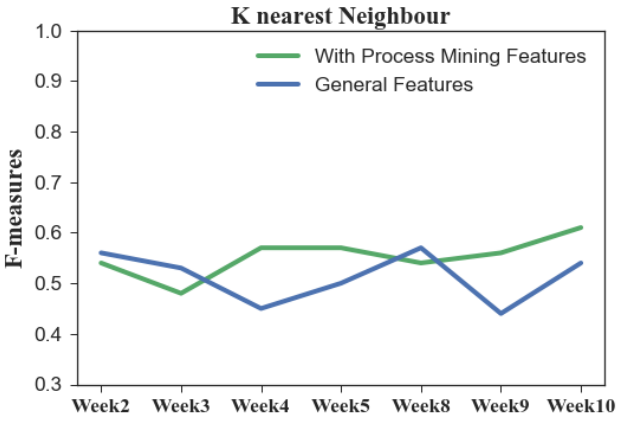


Figure 6.7: Comparative results of Knn classifier

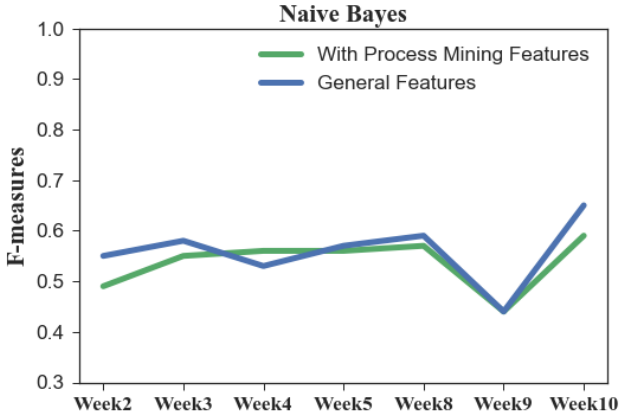


Figure 6.8: Comparative results of NB classifier

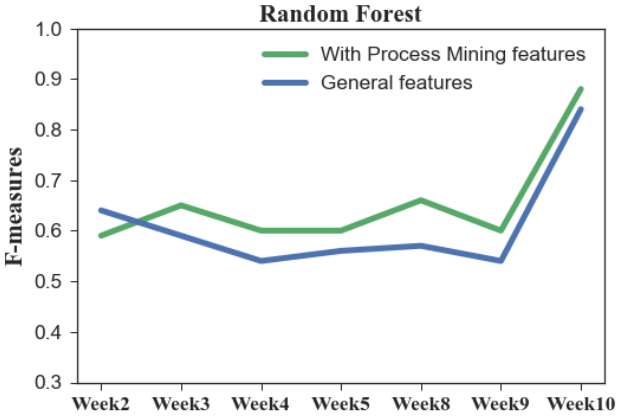


Figure 6.9: Comparative results of RF classifier

Next, to make comparisons of multiple classifiers and measure the significance of above findings Friedman test (Demšar 2006) was conducted. Friedman test is a non-parametric test used to determine whether the performance of the classifiers is consistent or not. It is also known by non-parametric randomized black analysis of variance(Settouti, Bechar, and Chikh 2016). When the p-value is small (<.05), null hypothesis is rejected. The goal of this test is to check if there is any significant difference in performance of classifiers for the given sets of data. Null hypothesis is "There is no difference in performance of classifiers". After conducting Friedman test on both datasets, the p-values obtained are p=.07 and p=.08; since the p-value is greater than .05 null hypothesis is accepted, or there is no difference in performance of multiple classifiers. The test also determines the rank of the algorithm from the best performing one to the poorest one. The best algorithm gets the highest mean rank,i.e, getting the rank of 1, the second best rank 2.The test compares the average rank of the algorithms(Demšar 2006). Findings show that for dataset1 the LR outperforms all classifiers, while RF gets the highest mean rank when applied to dataset 2 (refer table 6.2).

Weeks	LRP	KNNP	NBP	RFP
Week2	.60	.54	.49	.59
Week3	.68	.48	.55	.65
Week4	.55	.57	.56	.60
Week5	.57	.57	.56	.60
Week8	.63	.54	..57	.66
Week9	.56	.56	.44	.60
Week10	.62	.61	.59	.88
Rank(Mean)	2.14	3.00	3.5	1.28

Table 6.2: Results of F-measures and rank(mean) on datasets of process mining features

Since the RF gives better F measures when compared to all the other classifiers it was decided to test the 2017 data with the Random Forest classifier. So, all the 2016 data was utilized to build the model and all the 2017 data was tested on this model to predict the final Exam results. The same procedure was followed but the entire 2016 data was used to divide the students in to High risk, Low risk and Medium risk students. The process model was created for these students and the 2017 data was applied on these models and the fitness score obtained from this was utilized for the prediction analysis.

The final results were checked along with the predicted results and the F-measures obtained was 74% as shown in figure 6.10. This is because the number of activities done by the students for the year 2017 was more when compared to 2016 the reason being the last year activities (2016) was further break down in to smaller activities to make it easy for the students to complete it.

```
('Accuracy', 0.70817120622568097)
Confusion Matrix
[[  5   0   2]
 [  5 147  54]
 [  6   8  30]]
             precision    recall  f1-score   support

  High risk       0.31      0.71      0.43         7
   Low risk       0.95      0.71      0.81       206
Medium risk       0.35      0.68      0.46        44

avg / total       0.83      0.71      0.74       257
```

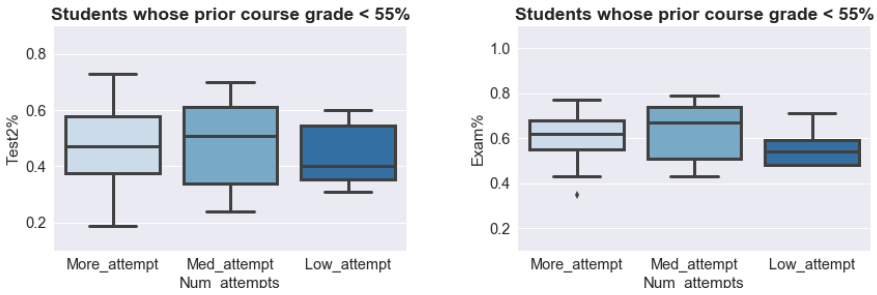
Figure 6.10: Confusion matrix of RF classifier

Out of 257 students from the year 2017, 16 students got below 55%, and out of the 16 students the model correctly predicted for 11 students. The Table 8 below illustrates the number of students on every class.

	participant_id
course_grade	
High risk	16
Low risk	155
Medium risk	86

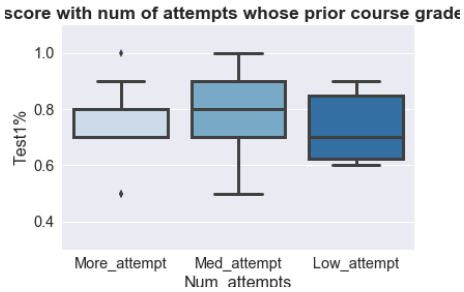
Table 6.3: Number of students on every category

In order to check if the students who performed poorly in prior course grade had performed better in the course or not, the number of attempts for a student to reach 70% in each Xorro-Q activity was summed across all the activities and percentage of those attempts was taken. The Industry partner (who provided us with these datasets) was interested in finding the performance between the top students and the poor students, hence those two categories were taken in to consideration. The students were classified based on High risk (<=55%), Low risk (>=75%) against the prior course grade and correlation of these students was found against Exam, Test 2 and Test 1 score. For Test1 only the activities which was done before Test1 i.e., till week 4 were taken in to account. The students are divided into two groups because some students had repeated the course hence their prior course grade was not there. The students were divided as: with prior course grade score and without prior course grade score. Analysis was performed for these two groups. Initially scatter plot was drawn but didn't show up the expected results clearly so it was decided to use box plot which is displayed in Fig 6.11. The number of attempts made by the students to do an activity was categorized into three groups for easy understanding. They



((a)) Test2

((b)) Final Exam



((c)) Test1

Figure 6.11: Box plot comparison of Xorro-Q activities with num of attempts for the year 2016

are more number of attempts (>1.8), Medium number of attempts(1.8>x >1.3) and less number of attempts (<1.3).

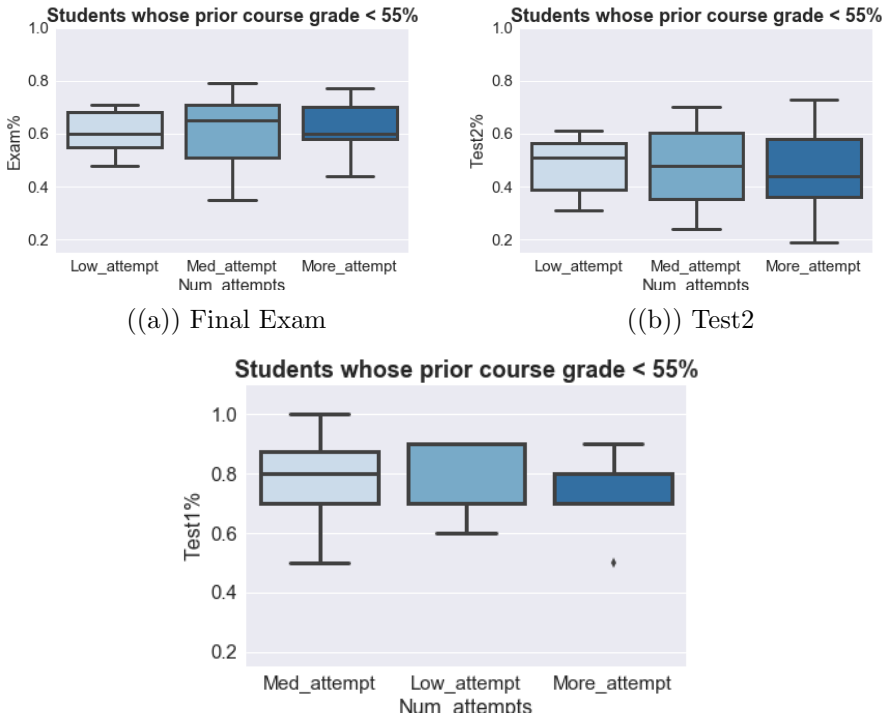
Students whose prior course grade <55%

While looking in the box plots above the students who acquired less than 55% in the prior course grade had made more number of attempts in Xorro-Q and had yielded good score in all the three assessments (i.e., Test1, Test2 and Exam) when compared to the students who made less number of attempts.

Difficult activities

Then three difficult activities were considered. Difficult activities are those activities where the students made more number of attempts as compared to other activities. Again, the Industry partner was interested in finding whether these helped students to achieve good scores or not. The same procedure was followed to see the performance among the High risk and the Low risk students. For the difficult activities, the students had tried more attempts when compared with the overall activities, so different ranges were used for grouping. These are: more number of attempts (>3), medium number of attempts (3>X >2) and low number of attempts (<2)

The box plot shows that doing difficult activities does not bring improvement for the top students, but these activities helped the poor students to achieve better results. The students who performed many number of attempts in doing the difficult Xorro-Q activities had higher max score and the median score than the students



((a)) Final Exam

((b)) Test2

((c)) Test1

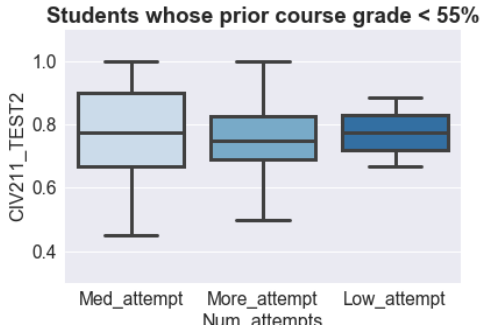
Figure 6.12: Box plot for difficult activities of num of attempts for the year 2016

who made lower number of attempts in doing the Xorro-Q activities, but then their minimum score for Test2 and the Exam was less when compared to the students with lower number of attempts.

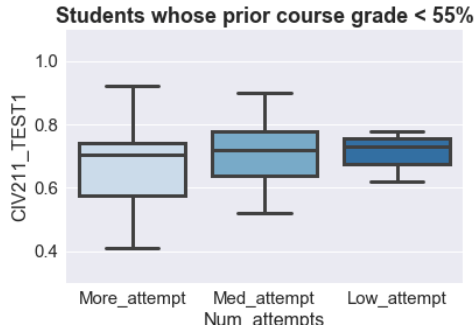
2017-year data

The same analysis (i.e., considering all the activities and filtering out the difficult activities) was performed with the 2017 data but here the students are divided into three groups. There were some new students whose prior course grade missing along with the other two groups, hence these students were also separated. The analysis was performed for students with prior course grade score since few students were there for the other two categories.

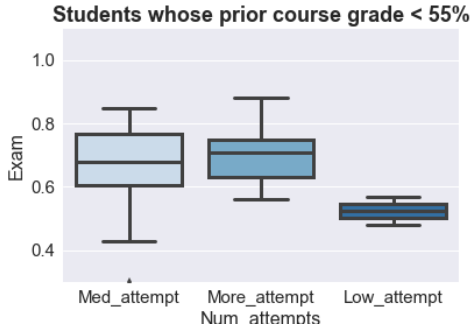
For the students below 55%, the median score improved with the number of attempts in the Exam. The students who had done many number of attempts in Xorro-Q got good score in Exam than the students with lower number of attempts; however, the number of attempts had no effect on Test 2 score. Similar kind of observation was seen even with the Test 1 score, but it was found that difficult activities really helped students to achieve good score in the Exam.



((a)) Test2

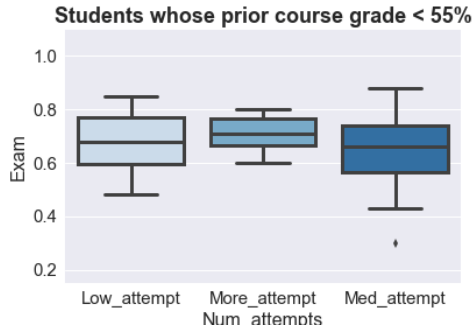


((b)) Test1

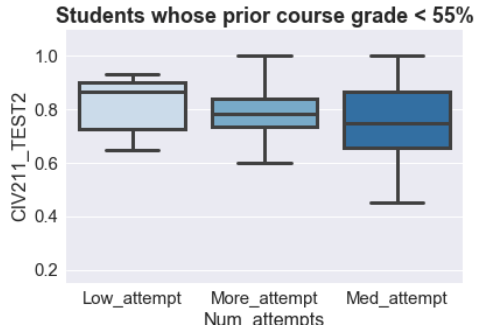


((c)) Final Exam

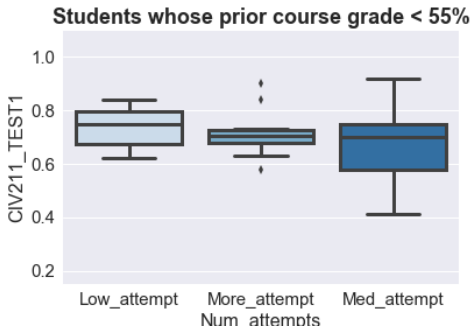
Figure 6.13: Box plot comparison of Xorro-Q activities with num of attempts for the year 2017



((a)) Final Exam



((b)) Test2



((c)) Test1

Figure 6.14: Box plot for difficult activities with num of attempts for the year 2017

While considering the difficult activities for the year 2017(refer figure 6.14), it did not make much difference with the top students like in the year 2016 but for the poor students this helped. All the students who made many number of attempts in doing the difficult activities got a minimum score of nearly 60% (even though its max score has no effect in Exam and Test 1). Moreover, the max score for the Test2 was quite high for the students who made more number of attempts while doing the difficult activities.

Then the number of answers attempted by the students are selected to test the hypothesis.

For doing so,the total answers which is count of all answers across all the activities were taken into consideration because it's a simple metric indicating the effort committed. The reasons for choosing answer counts for a student are speculated as

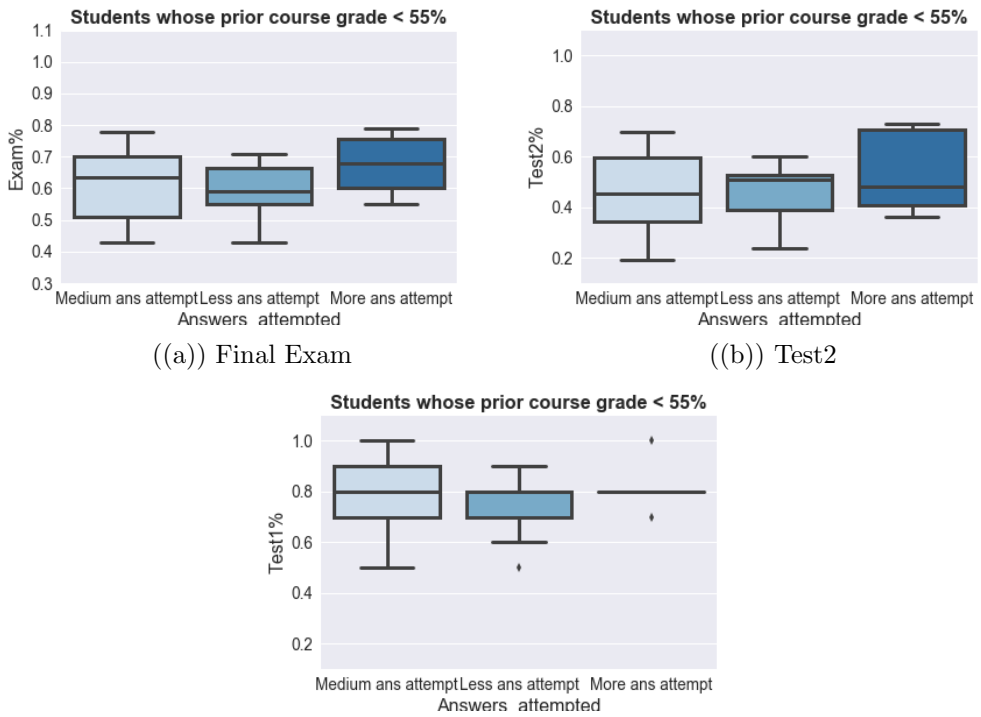
- A student who easily achieves results (e.g., one who got high grades in the prior course) is motivated to get better results through extra attempts to drive his/ her scores to 100% in Xorro-Q.

- The student who is struggling needs many attempts to achieve the required minimum outcome.

Then using box plot, correlation was found between total answers and Test 1 score, Test2 score and the final Exam score.

The analysis reveals that students who got less than 55% in prior course grade and who had taken more attempts to do the Xorro-Q activities had got more scores in Test2 and Exam; even their minimum and maximum scores were more when compared to the other student categories.But the students did not show any improvement on Test 1. Even the top students who scored more than 75% in prior course grade (refer appendix) got more scores in the Exam and Test 2. However, since here we are concentrating on students who have not performed too well and are struggling to meet the 55% score, the boxplot for these students is shown.

The 2017 data box plot analysis also shows that the students with less than 55% in the prior course grade had attempted more questions from the Xorro-Q activities and had achieved more scores as compared with the students with less number of attempts.



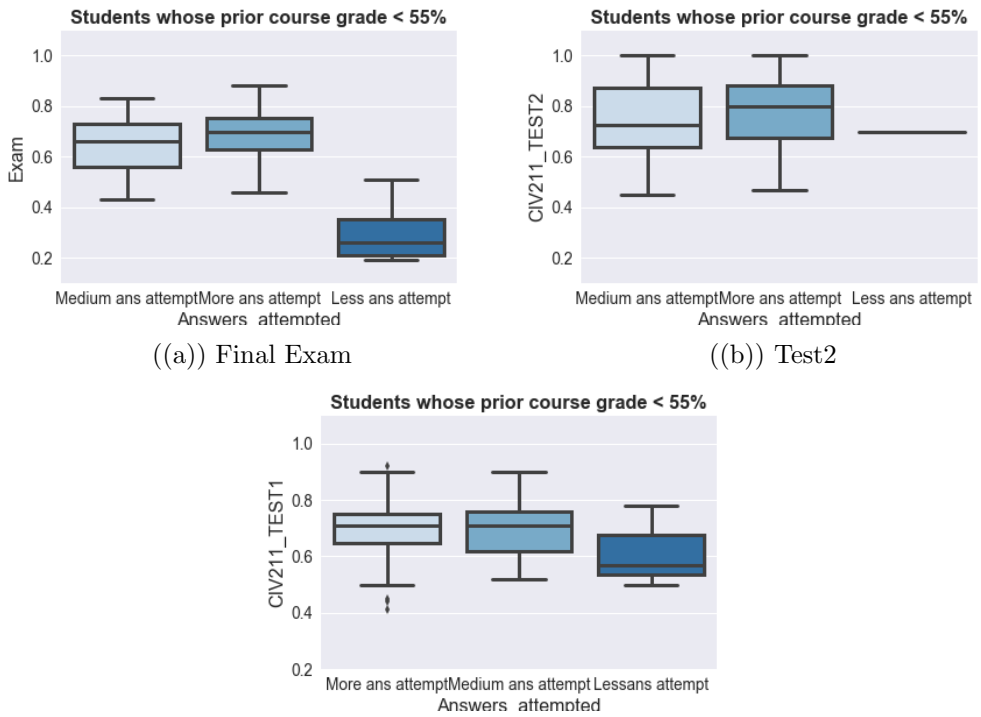
((a)) Final Exam

((b)) Test2

((c)) Test1

Figure 6.15: Box plot comparison of answers attempted for the year 2016

The 2017 data analysis shows the maximum score for the Exam and the Test1 is more than 80%, their median score is close to 75%. while the students who has not taken much activities and has answered only very less number of Xorro-Q activities has scored very less in the exam and the Test 1 and the max score for the students with less number of question attempts falls to 55%.Those students did not show any improvement even in Test 2. This indicates that attempting more questions in the Xorro-Q activities has helped the students to achieve more marks in their final Exam. Moreover, having done more attempts in specific activities is significant to achieving better results in Exams and is irrespective of number of correct and wrong answers to the questions in the activities.



((a)) Final Exam

((b)) Test2

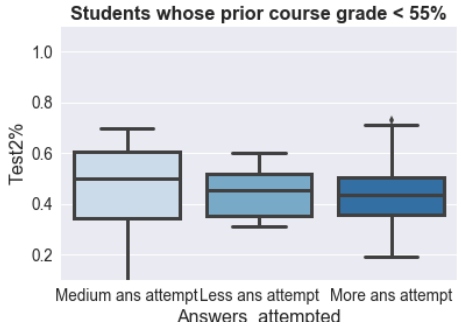
((c)) Test1

Figure 6.16: Box plot comparison of answers attempted for the year 2017

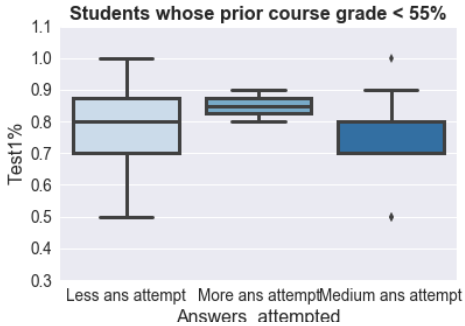
Difficult activities

Subsequently the difficult activities for the year 2016 and the year 2017 alone were filtered from the over all activities, and the same analysis was performed.Since the number of activities as already mentioned were further reduced in the year 2017 while comparing to the year 2016 so there were more number of difficult activities for the year 2017 when compared to the year 2016. The analysis revealed that there is not much difference between the student group which had more number of attempts from the group with less attempts for the top performing students (refer appendix) but there was some improvement in score for the poor performing students.

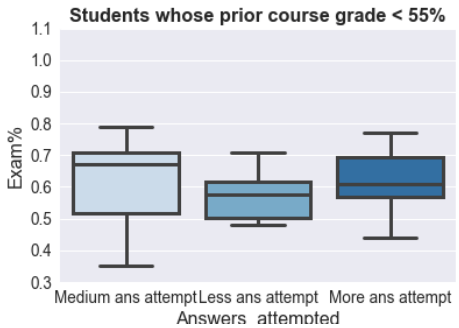
In the year 2016 the maximum mark scored by the students for Test2 and Exam was little higher for the more questions answered student group compared to the less questions answered group (see figure 6.17). But this was not the same case with the 2017 data. Here the students with less than 55% in the prior course grade and who made more attempts had scored good marks in all the three assessments (i.e, Test1, Test2 and Exam) and their median and max score was much better than the group with lower attempts (see Figure 6.18).



((a)) Test2

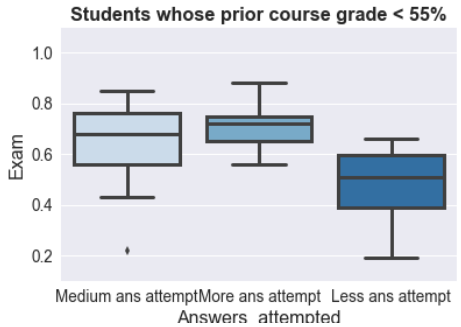


((b)) Test1

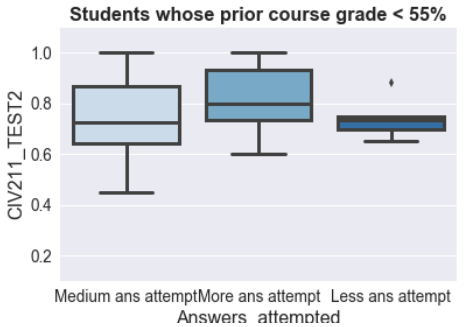


((c)) Final Exam

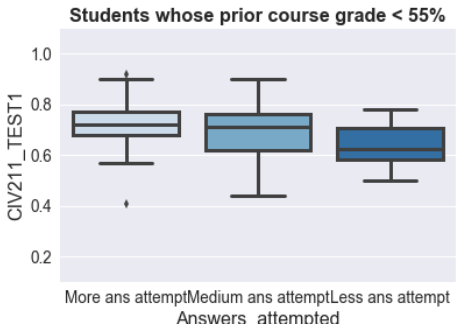
Figure 6.17: Box plot comparison for difficult activities answers attempted for the year 2016



((a)) Final Exam



((b)) Test2



((c)) Test1

Figure 6.18: Box plot comparison of difficult activities answers attempted for the year 2017

Chapter 7

Discussions and Future Scope

7.1 Discussions

The results have demonstrated that prediction of student's academic performance can be done with Xorro-Q features with an accuracy of 86%, where the RF outperformed all the other classifiers. In addition, process mining features were also integrated, but the improvement was not very significant, and RF proved to be the best performer on this dataset. While Examining the early prediction on students' performances, over a 12-week semester course, the accuracy showed to be increased gradually over the weeks, and the better accuracy was obtained in the 10th week.

To check whether the performance of the classifiers is identical, Friedman test was conducted, where the best algorithm would get the highest mean rank. Findings showed that there was no difference in performance of the classifiers. Alternatively, while determining the rank, Logistic regression got the highest mean rank (when applied to dataset 1), while on the other hand, RF achieved the highest mean rank (when applied to dataset 2).

Later, the 2017 data was tested using the 2016 model, where the entire 2016 data was used to create the model for testing the 2017 dataset to predict the final Exam results. The prediction accuracy obtained was 74% with the RF classifier. To ascertain that the activities were not repeated by the students as in 2016, they were segregated into smaller segments in 2017, which multiplied the count and led to a substantial increase in the activities performed by the students in 2017

7.2 Study limitations and future work

The fundamental motivation behind the utilization of data mining in education is to discover patterns and obtain knowledge about student's performance that might be utilized by the educators to assist students in learning. However, the present

research also has certain limitation.

Predicting students' academic performance using Xorro-Q is a challenging task. The reason being, the available data is extremely imbalanced and choosing the finest features is cumbersome. The present study used only the Xorro-Q data gathered in one semester and for one course in particular; hence the analysis was restricted the outcomes would be more promising if the data mining was carried for data that had been gathered for more than one course. The data as such was not designed which will be appropriate for mining. Subsequently, the data remained complicated, with many missing values. Also, this study did not have access to the demographic information of the students, as this is considered as classified information. However, numerous predictions tasks in the literature are potentially using these, which contributed to better accuracy. Albeit, this aspect was overlooked in this research work to avoid unnecessary discrepancies that may involve some ethics approvals. Therefore, the accuracy would have been compromised in relative to the results published in other literature. This limitation can be overcome by considering the above said demographic information in future work.

In future, this research work can be enhanced by carrying out the mixture of various machine learning methods. This research has implemented all the methods under classification. Other methods such as clustering, neural networks can also be utilized for improved representations of the output. Having said this, this investigation has shared new visions on strengthening the value of the role of analytics for predictive modelling by integrating process mining features in the training set of data.

Bibliography

Aalst, Wil MP van der, Alfredo Bolt, and Sebastiaan J van Zelst (2017). "RapidProM: Mine your processes and not just your data". In: *arXiv preprint arXiv:1703.03740*.

Abaidullah, Anwar Muhammad, Naseer Ahmed, and Edriss Ali (2015). "Identifying Hidden Patterns in Students" Feedback through Cluster Analysis". In: *International Journal of Computer Theory and Engineering* 7.1, p. 16.

Ahmed, Abeer Badr El Din and Ibrahim Sayed Elaraby (2014). "Data Mining: A prediction for Student's Performance Using Classification Method". In: *World Journal of Computer Application and Technology* 2.2, pp. 43–47.

Anwar, MA and Naseer Ahmed (2011). "Knowledge Mining in Supervised and Unsupervised Assessment Data of Students' Performance". In: *2011 2nd International Conference on Networking and Information Technology IPCSIT vol.* Vol. 17.

Aslan, Ayse (2017). "Combining Process Mining and Queueing Theory for the ICT Ticket Resolution Process at LUMC". MA thesis. University of Twente.

Badr, Ghada et al. (2016). "Predicting students' performance in university courses: a case study and tool in KSU mathematics department". In: *Procedia Computer Science* 82, pp. 80–89.

Baker, RSJD et al. (2010). "Data mining for education". In: *International encyclopedia of education* 7.3, pp. 112–118.

Baker, Ryan Shaun, Albert T Corbett, and Kenneth R Koedinger (2004). "Detecting student misuse of intelligent tutoring systems". In: *International conference on intelligent tutoring systems*. Springer, pp. 531–540.

Beck, Joseph E and Jack Mostow (2008). "How who should practice: Using learning decomposition to evaluate the efficacy of different types of practice for different

types of students". In: *International conference on intelligent tutoring systems*. Springer, pp. 353–362.

Bhavsar, Hetal and Amit Ganatra (2012). "A comparative study of training algorithms for supervised machine learning". In: *International Journal of Soft Computing and Engineering (IJSCE)* 2.4, pp. 2231–2307.

Bishop, Christopher M (2006). "Machine learning and pattern recognition". In: *Information Science and Statistics. Springer, Heidelberg*.

Black, Erik W, Kara Dawson, and Jason Priem (2008). "Data for free: Using LMS activity logs to measure community in online courses". In: *The Internet and Higher Education* 11.2, pp. 65–70.

Bogarın, Alejandro, Rebeca Cerezo, and Cristóbal Romero (2018). "A survey on educational process mining". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8.1.

Breiman, Leo (2001). "Random forests". In: *Machine learning* 45.1, pp. 5–32.

Bühlmann, Peter Lukas and Bin Yu (2000). "Explaining bagging". In: *Research report/Seminar für Statistik, Eidgenössische Technische Hochschule Zürich*. Vol. 92. Seminar für Statistik, Eidgenössische Technische Hochschule (ETH).

Cairns, Awatef HICHEUR et al. (2015). "Process mining in the education domain". In: *International Journal on Advances in Intelligent Systems* 8.1.

Cairns, Awatef Hicheur et al. (2015). "Analyzing and improving educational process models using process mining techniques". In: *IMMM 2015 Fifth International Conference on Advances in Information Mining Management*, pp. 17–22.

Castro, Félix, A Nebot, and Francisco Mugica (2007). "Extraction of logical rules to describe students' learning behavior". In: *Proceedings of the sixth conference on IASTED International Conference Web-Based Education*. Vol. 2, pp. 164–169.

Chalaris, Manolis et al. (2014). "Improving quality of educational processes providing new knowledge using data mining techniques". In: *Procedia-Social and Behavioral Sciences* 147, pp. 390–397.

Criminisi, Antonio, Jamie Shotton, Ender Konukoglu, et al. (2012). "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning". In: *Foundations and Trends® in Computer Graphics and Vision* 7.2–3, pp. 81–227.

Dayton, C Mitchell (1992). "Logistic regression analysis". In: *Stat*, pp. 474–574.

Delavari, Naeimeh, Somnuk Phon-Amnuaisuk, and M Reza Beikzadeh (2008). "Data mining application in higher learning institutions." In: *Informatics in Education* 7.1, pp. 31–54.

Demir, Necati (2010). *Ensemble Methods: Elegant Techniques to Produce Improved Machine Learning Results*. URL: `https://www.toptal.com/machine-learning/ensemble-methods-machine-learning`.

Demšar, Janez (2006). "Statistical comparisons of classifiers over multiple data sets". In: *Journal of Machine learning research* 7.Jan, pp. 1–30.

Feng, Mingyu et al. (2008). "Can we predict which groups of questions students will learn from?." In: *EDM*, pp. 218–225.

Gentle, James E, Wolfgang Karl Härdle, and Yuichi Mori (2012). *Handbook of computational statistics: concepts and methods*. Springer Science & Business Media.

Ghawi, Raji (2016). "Process Discovery using Inductive Miner and Decomposition". In: *arXiv preprint arXiv:1610.07989*.

Giudici, Paolo (2005). *Applied data mining: Statistical methods for business and industry*. John Wiley & Sons.

Goyal, Monika and Rajan Vohra (2012). "Applications of data mining in higher education". In: *International journal of computer science* 9.2, p. 113.

Grigorova, K, E Malysheva, and S Bobrovskiy (2017). "Application of Data Mining and Process Mining approaches for improving e-Learning Processes". In: *(-2017)*, pp. 1960–1966.

Günther, Christian W (2009). "Process mining in flexible environments". In:

Han, Jiawei, Jian Pei, and Micheline Kamber (2011). *Data mining: concepts and techniques*. Elsevier.

Hanna, Margo (2004). "Data mining in the e-learning domain". In: *Campus-wide information systems* 21.1, pp. 29–34.

Hoffait, Anne-Sophie and Michael Schyns (2017). "Early detection of university students with potential difficulties". In: *Decision Support Systems* 101, pp. 1–11.

Hornix, Peter TG (2007). "Performance analysis of business processes through process mining". In: *Master's Thesis, Eindhoven University of Technology*.

Jacob, John et al. (2015). "Educational Data Mining techniques and their applications". In: *Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on*. IEEE, pp. 1344–1348.

James, Gareth et al. (2013). *An introduction to statistical learning*. Vol. 112. Springer.

Janecek, Andreas (2009). "Efficient feature reduction and classification methods". PhD thesis. uniwien.

Jason, Brownlee (2016). *How to estimate the performance of machine learning algorithms in Weka*. URL: `https://machinelearningmastery.com/estimate-performance-machine-learning-algorithms-weka/)`..

Kumar, Varun and Anupama Chadha (2012). "Mining association rules in student's assessment data". In: *International Journal of Computer Science Issues* 9.5, pp. 211–216.

Leemans, Sander JJ, Dirk Fahland, and Wil MP van der Aalst (2016). "Scalable process discovery and conformance checking". In: *Software & Systems Modeling*, pp. 1–33.

Luan, Jing (2002). "Data mining and its applications in higher education". In: *New directions for institutional research* 2002.113, pp. 17–36.

Manhães, Laci Mary Barbosa, Sérgio Manuel Serra da Cruz, and Geraldo Zimbrão (2014). "WAVE: an architecture for predicting dropout in undergraduate courses

using EDM". In: *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. ACM, pp. 243–247.

Márquez-Vera, Carlos et al. (2013). "Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data". In: *Applied intelligence* 38.3, pp. 315–330.

Mueen, Ahmed, Bassam Zafar, and Umar Manzoor (2016). "Modeling and Predicting Students' Academic Performance Using Data Mining Techniques". In: *International Journal of Modern Education and Computer Science* 8.11, p. 36.

Müller, Andreas C and Sarah Guido (2016). *Introduction to machine learning with Python: a guide for data scientists*. " O'Reilly Media, Inc.".

Munoz-Gama, Jorge et al. (2014). "Conformance checking and diagnosis in process mining". PhD thesis. Springer.

Nikam, Sagar S (2015). "A comparative study of classification techniques in data mining algorithms". In: *Oriental Journal of Computer Science and Technology* 8.1, pp. 13–19.

Nilsson, Nils J (1996). *Introduction to machine learning: An early draft of a proposed textbook*.

Nsofor, Godswill Chukwugozie (2006). "Comparative analysis of predictive data-mining techniques". In:

Nyce, Charles and API CPCU (2007). "Predictive analytics white paper". In: *American Institute for CPCU. Insurance Institute of America*, pp. 9–10.

Osmanbegović, Edin and Mirza Suljić (2012). "Data mining approach for predicting student performance". In: *Economic Review* 10.1, pp. 3–12.

Pandey, Umesh Kumar and Saurabh Pal (2011). "Data Mining: A prediction of performer or underperformer using classification". In: *arXiv preprint arXiv:1104.4163*.

Pechenizkiy, Mykola et al. (2009). "Process Mining Online Assessment Data." In: *International Working Group on Educational Data Mining*.

Peng, Chao-Ying Joanne, Kuk Lida Lee, and Gary M Ingersoll (2002). "An introduction to logistic regression analysis and reporting". In: *The journal of educational research* 96.1, pp. 3–14.

Pojon, Murat (2017). "Using Machine Learning to Predict Student Performance". In:

Polamuri, Saimadhu (2017). *How the random forest algorithm works in machine learning*.

Polumetla, Aditya (2006). "Machine learning methods for the detection of RWIS sensor malfunctions". PhD thesis. Citeseer.

Powers, David Martin (2011). "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation". In:

Priya, K Shanmuga and AV Senthil Kumar (2013). "Improving the student's performance using educational data mining". In: *International Journal of Advanced Networking and Applications* 4.4, p. 1806.

Priyadarshini, Nandita (2017). "A review: Data Mining Techniques in Education Academia". In:

Rajibussalim, MInfoTech (2014). "Data Mining for Studying the Impact of Reflection on Learning". PhD thesis. University of Sydney.

Refaeilzadeh, Payam, Lei Tang, and Huan Liu (2016). "Cross-validation". In: *Encyclopedia of database systems*, pp. 1–7.

Reimann, Peter, Lina Markauskaite, and Maria Bannert (2014). "e-Research and learning theory: What do sequence and process mining methods contribute?" In: *British Journal of Educational Technology* 45.3, pp. 528–540.

Ribeiro, JTS (2013). "Multidimensional process discovery". In: *Eindhoven University of Technology, Eindhoven*.

Romero, Cristobal and Sebastian Ventura (2013). "Data mining in education". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 3.1, pp. 12–27.

Rudnitckaia, Julia (n.d.). "Process Mining. Data science in action". In: *University of Technology, Faculty of Information Technology*, pp. 1–11.

Sarkar, Manish and Tze-Yun Leong (2000). "Application of K-nearest neighbors algorithm on breast cancer diagnosis problem." In: *Proceedings of the AMIA Symposium*. American Medical Informatics Association, p. 759.

Saxena, Rahul (2016). *Knn Classifier, Introduction to K-Nearest Neighbor Algorithm*.

scikit-learn (2007). *Ensemble methods*. URL: `http://scikit-learn.org/stable/modules/ensemble.html#voting-classifier`.

Settouti, Nesma, Mohammed El Amine Bechar, and Mohammed Amine Chikh (2016). "Statistical comparisons of the top 10 algorithms in data mining for classification task". In: *International Journal of Interactive Multimedia and Artificial Intelligence* 4.1, pp. 46–51.

Shaw, Reena (2017). *Top 10 Machine learning Algorithms for Beginners*. URL: `https://www.kdnuggets.com/2017/10/top-10-machine-learning-algorithms-beginners.html`.

Shih, Benjamin, Kenneth R Koedinger, and Richard Scheines (2011). "A response time model for bottom-out hints as worked examples". In: *Handbook of educational data mining*, pp. 201–212.

Southavilay, Vilaythong, Kalina Yacef, and Rafael A Callvo (2010). "Process mining to support students' collaborative writing". In: *Educational Data Mining 2010*.

Sumana, BV and T Santhanam (2015). "Optimizing the Prediction of Bagging and Boosting". In: *Indian Journal of Science and Technology* 8.35.

Şuşnea, Elena (2009). "Using data mining techniques in higher education". In: *The 4th international conference on virtual learning*, p. 373.

Trcka, Nikola and Mykola Pechenizkiy (2009). "From local patterns to global models: Towards domain driven educational process mining". In: *Intelligent Systems Design and Applications, 2009. ISDA'09. Ninth International Conference on*. IEEE, pp. 1114–1119.

Vahdat, M (2017). "Learning analytics and educational data mining for inquiry-based learning". In:

Van der Aalst, Wil MP (2013a). "Decomposing Petri nets for process mining: A generic approach". In: *Distributed and Parallel Databases* 31.4, pp. 471–507.

— (2013b). "Process mining in the large: a tutorial". In: *European Business Intelligence Summer School*. Springer, pp. 33–76.

— (2016). *Process mining: data science in action*. Springer.

Van Dongen, Boudewijn F et al. (2005). "The ProM framework: A new era in process mining tool support". In: *International Conference on Application and Theory of Petri Nets*. Springer, pp. 444–454.

Van, DA (2011). *Process Mining Discovery, Conformance and Enhancement of Business Processes*.

VanderPlas, Jake (2016). *Python data science handbook: Essential tools for working with data*. " O'Reilly Media, Inc.".

Wen, Lijie, Wil MP van der Aalst, et al. (2007). "Mining process models with non-free-choice constructs". In: *Data Mining and Knowledge Discovery* 15.2, pp. 145–180.

Wen, Lijie, Jianmin Wang, et al. (2010). "Mining process models with prime invisible tasks". In: *Data & Knowledge Engineering* 69.10, pp. 999–1021.

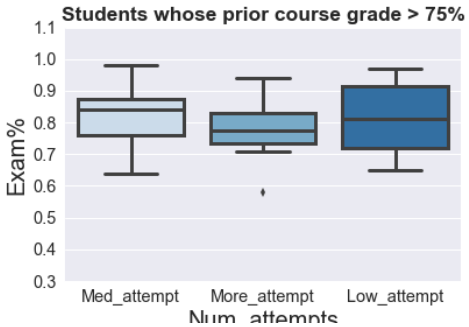
Wu, Xindong et al. (2008). "Top 10 algorithms in data mining". In: *Knowledge and information systems* 14.1, pp. 1–37.

Yadav, Sanjay and Sanyam Shukla (2016). "Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification". In: *Advanced Computing (IACC), 2016 IEEE 6th International Conference on*. IEEE, pp. 78–83.

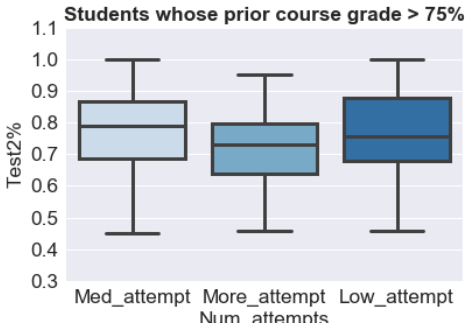
Zhang, Ying et al. (2008). "Use Data Mining to Improve Student Retention in Higher Education– A case Study". In:

Chapter 8

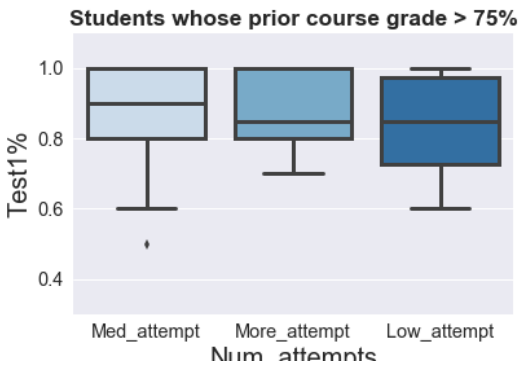
Appendix



((a)) Final Exam

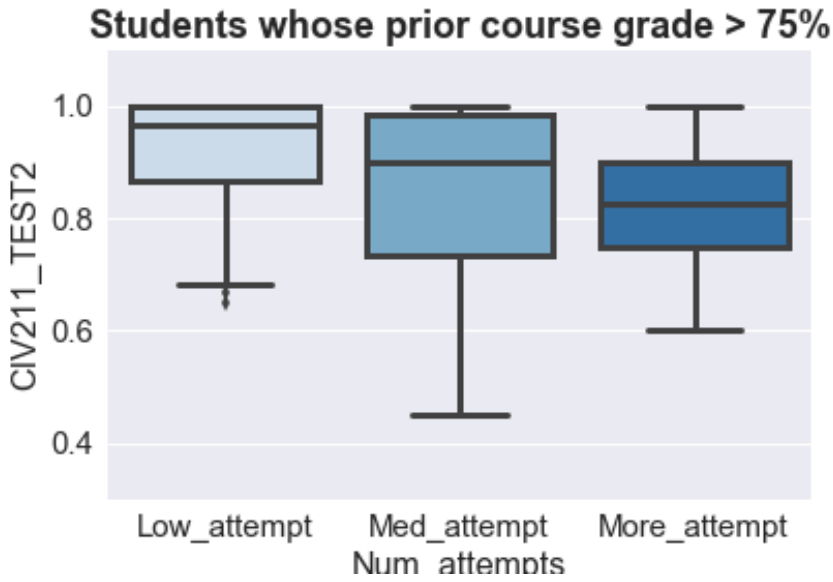


((b)) Test2

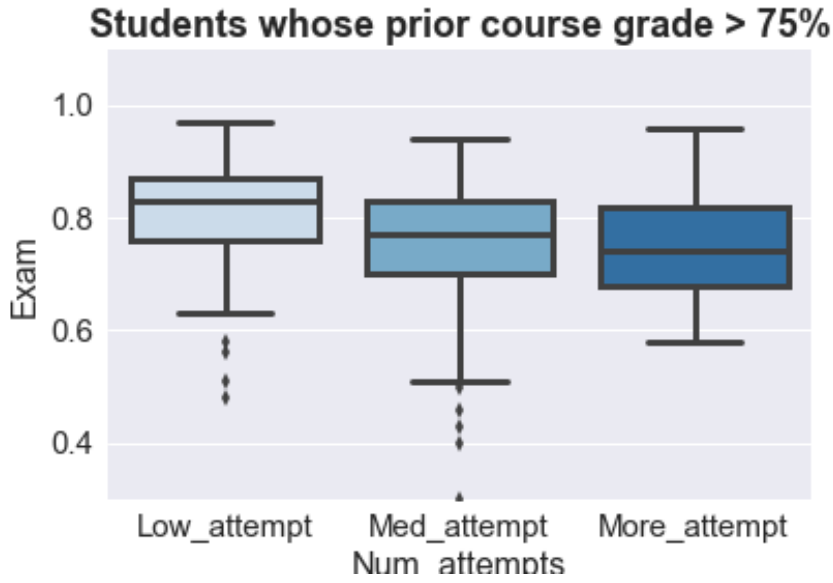


((c)) Test1

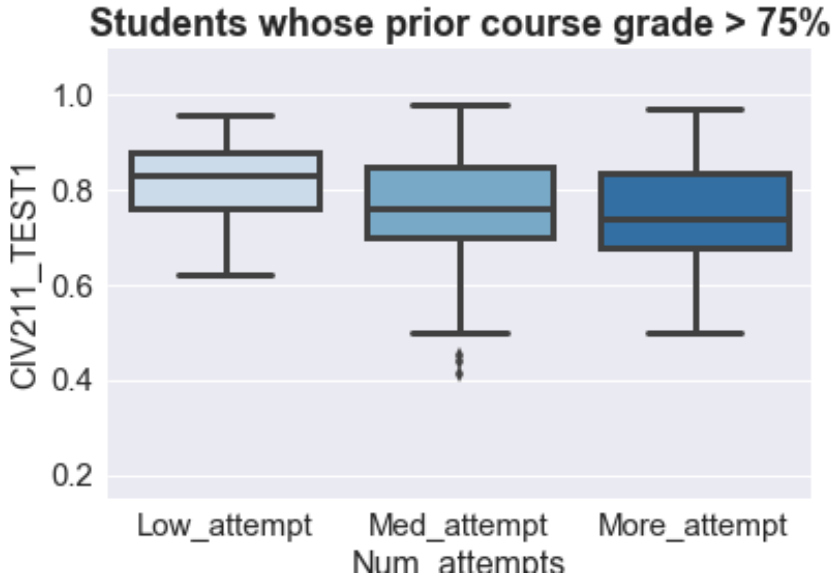
Figure 8.1: Box plot comparison of Xorro-Q activities with num of attempts



((a)) Test2

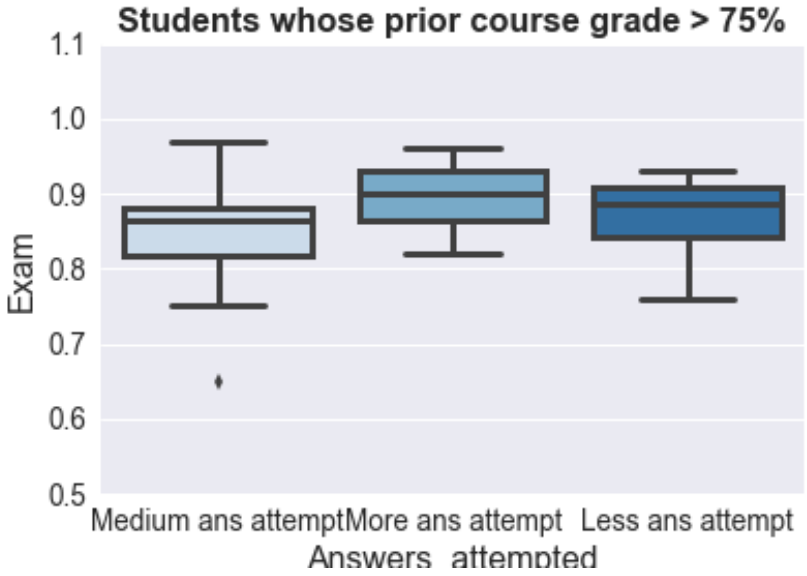


((b)) Final Exam

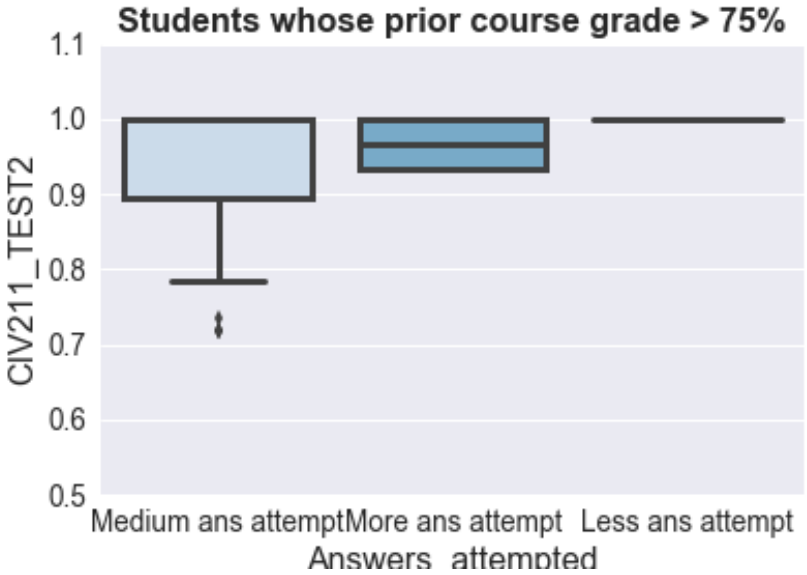


((c)) Test1

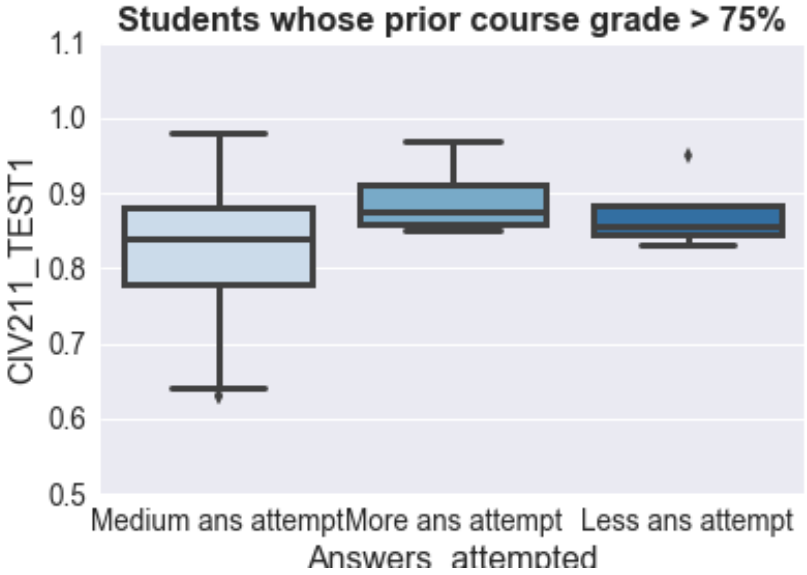
Figure 8.2: Box plot comparison of difficult activities with num of attempts



((a)) Final Exam



((b)) Test2



((c)) Test1

Figure 8.3: Box plot comparison of Xorro-Q activities with answers attempted

rticipatable_id	percentage_minscore	percentage_maxscore	percentage_meanscore	average_attempts	percentage
814	0.566532	0.869711	0.734940	1.880000	0.333333
815	0.525687	0.645628	0.586109	1.413793	0.700000
816	0.736287	0.856075	0.797471	1.480000	0.857143
817	0.776377	0.917283	0.855150	2.090909	0.444444
818	0.579157	0.839727	0.706647	2.257143	0.750000

Figure 8.4: attributes

_attempts	percentage_answers_correct	Prior Course Grade	Test1%	Test2%	Average	Exam%
	0.333333	0.49	0.8	0.61	0.705	0.70
	0.700000	0.55	0.8	0.50	0.650	0.67
	0.857143	0.94	0.9	0.91	0.905	0.92
	0.444444	0.68	0.8	0.74	0.770	0.82
	0.750000	0.62	0.7	0.53	0.615	0.67

Figure 8.5: attributes