Predicting the Completion of a Student's Science Degree based only on their First-year Marks

Gcobisile Matafeni

UNIVERSITY OF THE
WITWATERSRAND,
JOHANNESBURG

School of Computer Science and Applied Mathematics
Faculty of Science
HONOURS RESEARCH REPORT

Supervisor(s):
Mr. Ritesh Ajoodha
Dr. Benjamin Rosman

A research report submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in partial fulfillment of the requirements for the Honours degree in Computer Science

Declaration
University of the Witwatersrand, Johannesburg
School of Computer Science and Applied Mathematics
SENATE PLAGIARISM POLICY

I, Gcobisile Matafeni, (Student number: 709637) am a student registered for COMS4044A in the year 2017. I hereby declare the following:

- I am aware that plagiarism (the use of someone else's work without their permission and/or without acknowledging the original source) is wrong.
- I confirm that ALL the work submitted for assessment for the above course is my own unaided work except where I have explicitly indicated otherwise.
- I have followed the required conventions in referencing the thoughts and ideas of others.
- I understand that the University of the Witwatersrand may take disciplinary action against me if there is a belief that this in not my own unaided work or that I have failed to acknowledge the source of the ideas or words in my writing.

Signature:

Signed on _____ day of _____ , 2017 in Johannesburg.

Abstract

Over the past couple of decades, much work has been done in the field of predicting student success in first year computer science and in other first year courses. Our principal contribution is to provide an expert system that statistically predicts the success of a first year student given only their academic merit and subject matter. The ability to predict the completion of a student's degree based only on their first year marks can serve as a powerful tool that would help students withstand unnecessary debt and not waste many years struggling to qualify. Historically, other authors focused on using linear statistical models to predict student success in first year courses. This report presents an approach of using the naïve Bayes classifier, support vector machines and decision trees as models that can be used to predict the completion of an undergraduate science degree. The models used student marks that were collected from the Faculty of Science at the University of the Witwatersrand, Johannesburg. The models were used to test the following hypothesis that states, students who perform well in their first year of study, and who elect complementing courses from second year obtain their science degrees in record time, this is a period of 3 years. This was done by firstly training the classifiers and then testing them. The support vector machine achieved the best accuracy (87%) in predicting the completion of a science degree based only on first year marks, this was followed by the naïve Bayes model (86.36%) and the decision tree (65.62%) came last.

Acknowledgements

Regarding my research and other help, I would like to express my sincere gratitude to Ritesh Ajoodha, my Honours project supervisor and to Dr. Benjamin Rosman my co-supervisor. I would also like to give many thanks to Johannes Kokozela for helping me during my data preparation phase, the Wits BSc Honours (CS) class (2017) and the whole Stack-overflow community for support, ideas and solutions for debugging my code. Furthermore, I would also like to thank the Industrial Development Corporation (IDC) for the financial assistance provided.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

Predicting the completion of a student's degree can serve as a powerful tool that would help students withstand unnecessary debt and not waste many years struggling to qualify. A view that is generally agreed upon is that acceptance into a university programme often proves to be life changing. The prospect of obtaining a degree is sometimes a promise of higher income and improves ones standard of living [Thurow 1972]. The idea of students having the ability to know after their first year of study which courses to take in order to maximize their chances of succeeding is helpful. Historically, institutions of higher learning have been struggling to improve their throughput rates over the years. Since the dawn of the democratic dispensation, enrollment rates in higher education institutions have sky rocketed while the dropout rates have increased significantly [Scott 2013]. To mitigate drop-out rates and improve success, this research project will attempt to build a model for calculating the probability of a student completing an undergraduate degree in the Faculty of Science at the University of the Witwatersrand, Johannesburg.

The methodologies that have been used by previous authors are different from the approach that will be utilized in this project. The motivation for the project is building an expert system that will mitigate the problems that have been identified. There hasn't been a model that could be used to successfully predict the completion of a student's science degree, so this project is also motivated by this. According to [Butcher and Muth 1985] and [Campbell and McCabe 1984], some useful features to recognize the success of a student in first year computer science include using the Scholastic Aptitude Test (SAT) and the American College Testing (ACT) scores across different subjects. The approach of this project is intuitive as it must determine the value associated with taking a combination of subjects and tie it to the completion of a student's degree. This kind of methodology has not been observed in the surveyed literature. It will also have the ability to calculate the alternative streamlines that might align the student with better options, so that he can reconsider their academic standing. In [Butcher and Muth 1985], the authors used statistical analysis in order to build a linear model that described the correlations using all linear combinations of the dependent variables. [Gathers 1986] made use of a step-wise discriminant to identify all significant factors in a study group of 87 freshman computer science majors. The above papers demonstrate the methodologies that rely on using statistical models to do the classifications.

Having contextualized the problem statement for this research project we can proceed to generate a research question. This question will be followed by a research hypothesis that will be tested during the course of the project. This will be done by using the methodology that will be outlined later on. Under the topic of predicting the completion of a student's science degree we can have the following question. Is there a correlation between first year academic results and the number of years a student will take to complete their degree? This question is formulated to study the relationship between first year marks and the probability of completing a science degree. From the above question we can have the following hypothesis, students who perform well in their first year of study and who elect complementing courses from second year, they obtain their science degrees in record time (3 years). An example of complementing courses would be Physics II and Mathematics II or you can have Computational and Applied Mathematics II. This is a well balanced combination of courses with respect to its complementing subject matter.

The research hypothesis was tested by following the research methodology that is outlined in Chapter 3. The data was gathered from the University of the Witwatersrand, Johannesburg, from the department of Academic Information Systems Unit (AISU). This was done with the help of the supervisor who liaised with the relevant staff member. This was the first phase of the methodology and it was followed by data preprocessing in order to prepare the data for Weka. The models were trained and test in Weka and we used a naïve Bayes classifier, support vector machine and a decision tree. The three approaches are distinct from methods that have been used by other authors. A naïve Bayes classifier was trained and it was able to calculate the probability of a student to succeed in a set of possible degrees given their first year marks. Our principal contribution from this research project is to provide an expert system that statistically predicts the success of a first year student given their academic merit and subject matter. This contribution is unique as it is a different piece of work from the literature that is reviewed in Chapter 2.

The naïve Bayes was able to achieve an accuracy of 86.36% which is comparable to the accuracy of the support vector machine 87%. The decision tree learning approach achieved an accuracy of 65.62% which is quite low as compared to the first two methods. The results of the three techniques are a generalization of student results as they used all the 216 features in the dataset. A more holistic approach was to build a model that can predict student success for tailored degrees in the Faculty of Science. The tailored degrees fall under four categories which are; Physical Sciences, Mathematical Sciences, Biological and Life Sciences and the last one is Earth Science. These degrees have prescribed core courses for first year students and using these courses as features we can see whether there is an improvement in one of the above models. The naïve Bayes technique was used for each of the four categories and the recorded accuracy for Physical Sciences was 86.38%, Mathematical Sciences 84.11%, Biological and Life Sciences 85.97%, and Earth Sciences 86.42%. These results are all comparable to each other and to the general model, the results will be discussed in detail in Chapter 4.

The remaining parts of this research report are structured as follows, since the research problem has been introduced, Chapter 2 reviews the literature and the related work. Chapter 3 discusses the research methodology that was used to conduct the relevant experiments that were used to test the research hypothesis. Chapter 4 presents the qualitative results of the experiments and it discusses them thoroughly. Lastly, Chapter 5 gives a summary of our work and what we have done, it also states future work that can be done to extend the work of this research project.

Chapter 2
Background and Related Work

2.1 Introduction

Chapter 1 outlined some of the research that has been done in the field of predicting student success in first year computer science and introduced the topic by providing the context. In this research project we are concerned with predicting the completion of an undergraduate science degree based only on first year marks. The project will be adapted from the work in the literature survey which focused on predicting student success in first year computer science by using high school results. This chapter provides the necessary background and related work that was used during the research project.

There are 3 main themes in this chapter, first we look at the literature survey and the related work, secondly we then look at the machine learning techniques that were used. Lastly, we look at a few feature extraction and selection techniques. Section 2.2, gives some background on the work that has been done in predicting success at first year computer science. It firstly looks at how multiple factors can be used to construct a model for predicting success and then it looks at how Mathematics can be used as a single predictor. The section then looks at another single predictor which uses English for classification purposes. Section 2.3 provides the theory of a naïve Bayes classifier and details their advantages and disadvantages while Section 2.4 provides the theory of support vector machines and also looks into their advantages and disadvantages. Section 2.5 provides a brief introduction to decision tree learning and Section 2.6 delves into feature extraction and selection techniques. This will be followed by a brief summary in section 2.7 of the important points of this chapter.

2.2 Predicting success at first year computer science

In this section, we look at different approaches that were used to predict the performance of first year students who majored in computer science. We first evaluate how multiple factors were used for this purpose, we then move into how mathematics as a single subject was used for prediction purposes. Lastly, we look at how English as another single subject was used for prediction purposes.

2.2.1 Predicting success using multiple factors

There are many approaches that can be used to look at what affects student performance in first year courses. One approach is to use multiple variables to compute the probability of succeeding in a freshmen major. In [Butcher and Muth 1985], 13 variables are used which were independent to each other, these include the American College Testing (ACT) scores for mathematics, the American College Testing (ACT) scores for English, the American College Testing (ACT) scores for natural science and the student's class rank to name a few. The American College Testing (ACT) are assessments that are used to measure the college readiness of high school students in America. The authors focused their study on a sample size of 269 students. For every variable, the mean, standard deviation, minimum and the maximum were calculated. Utilizing the correlations that were developed, a linear model was used to fit the data and the results indicated that it is possible to calculate the

probability of a pass or a fail in an introductory computer science course. In [Campbell and McCabe 1984], the authors used multiple factors to predict the performance in a computer science major and some of their work which focused on ACT English and other scores was extended in [Butcher and Muth 1985]. In [Campbell and McCabe 1984], the authors consider the SAT scores, the sex of the student and their high school grades. By combining all of these they developed a linear discriminant as a function to perform classification. In a data sample of 256 students, they were able to successfully classify 175 students which is 68.4% of the data into the correct group.

Gathers [1986] studied 10 factors and used these to find their relationship to student success in a first year computer science major. The ACT scores were used similarly to what [Butcher and Muth 1985] used in their variables. According to [Gathers 1986], the placement factors that reduced the failure rate in first year were the ACT English scores and the UTM mathematics placement scores, with the former scores being the best predictor. To identify the factors that contribute to the forecasting of success a discriminate function was used. In [Gathers 1986], the author found that the other factors were not significant to be used as predictors. [Campbell and McCabe 1984] found SAT mathematics and verbal scores as best predictors but did not indicate which one was the best predictor. In [Hostetler 1983], the author builds a model that forecasts the success in a programming course, with the hope of counseling students to make informed decisions. On top of using past academic achievement as used in [Butcher and Muth 1985], [Campbell and McCabe 1984] and [Gathers 1986], the author includes certain cognitive skills and personality traits. The study focused on a sample size of 120 students that was randomly selected from a population of 600 students. The variables that were used in this paper are both independent and dependent and they are 21 in total. This is significantly more variables in comparison to the number of variables that were used in other papers; [Butcher and Muth 1985], [Campbell and McCabe 1984] and [Gathers 1986]. The multiple regression equation that was developed was able to classify 61 students out of 79 (77.2%). The approach of using multiple factors to build a model for predicting the success of students worked in the papers mentioned above but there are other approaches.

2.2.2 Predicting success using Mathematics

[Capstick et al. 1975] uses the IBM Aptitude Test for Programmer Personnel (ATTP) scores to classify the students who are doing different computing courses, one based on COBOL and the other based on FORTRAN. The study focused on 46 students who had written the ATTPs. A positive correlation between the FORTRAN course and the arithmetical component of the ATTP was found, another was found between the letter series and the COBOL course. The only factor that was used for forecasting the performance in an introductory computer course were the ATTP scores which is a different approach from using a combination of variables. In a paper that focuses on a commerce degree, the author [Tewari 2014] argues for the use of mathematics as a single variable rather than the matric aggregate to predict the performance in first year. The study focused on results that were taken for a period of 4 years. The approach used was to take the individual mathematics scores and compare them to pass rates in first year. According to [Tewari 2014], a good mark in matric mathematics is the best predictor of first year success compared to the matric aggregate scores.

2.2.3 Predicting success using English

Similar to the approaches used in [Capstick et al. 1975] and [Tewari 2014], in [Rauchas et al. 2006] the authors use a single factor to predict the success in first year computer science. The paper studies the correlation between success in English and the actual performance in first year. Analysis took place by use of qualitative data from a survey and an in-depth quantitative look at matric results. Results show that language scores from matric results are better predictors of success. In contrast, a recent publication by [Tewari 2014] argues about

an important relationship between mathematics results in matric and the performance in a commerce degree. The argument is of great importance as it bases its findings on the same matric results as in [Rauchas et al. 2006]. Also, in [Tewari 2014] the author did not consider language in their study and according to [Rauchas et al. 2006] there is a possibility that it might have been a factor. As alluded to earlier, [Campbell and McCabe 1984] states that it is inappropriate to only use a single matric subject as a predictor of success for a first year computer science major. A holistic approach is to use the matric aggregate as a predictor. It is worthy revisiting the findings of [Gathers 1986], the research found that amongst the several variables that were used in the study, the best single predictor for success were the ACT English scores. The overall best predictor for success in computer science was the combination of UTM mathematics and ACT English placement scores. The predictor that was developed in [Gathers 1986] was able to to successfully reduce the failure rate in computer science the following year from 28% to 18% as stated in [Rauchas et al. 2006].

2.3 The naïve Bayes classifier

The naïve Bayes classifier is a special case of a Bayesian network this is because it assumes that each attribute is conditionally independent of every other attribute. The conditional independencies are represented as set of random variables. The naïve Bayes is a classification technique that uses Bayes's theorem for a set of variables. Bayes's theorem states that:

$$P(y|x_1, ..., x_n) = \frac{P(y)P(x_1, ..., x_n|y)}{P(x_1, ..., x_n)} \tag{2.1}$$

The term $P(y|x_1, ..., x_n)$ is the posterior, $P(y)$ is the prior, $P(x_1, ..., x_n|y)$ is the likelihood term relative to the class label and $P(x_1, ..., x_n)$ is the likelihood of the data.

The naïve Bayes classifier is easy to build as it is a naïve Bayes's model and it is useful for very large data sets. The figure below illustrates a generalized naïve Bayes classifier.

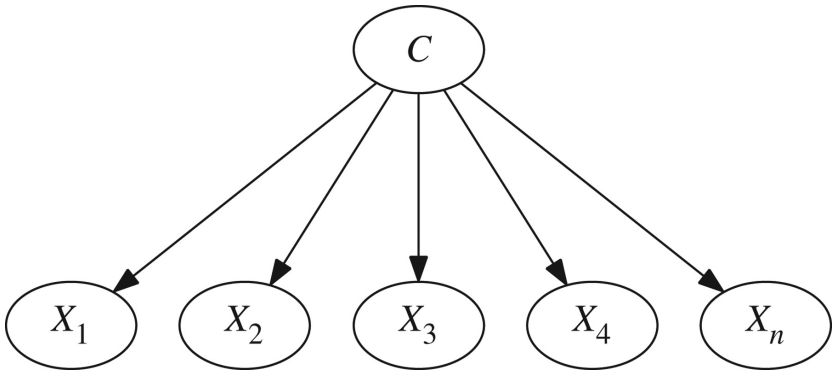

Figure 2.1: A simple naïve Bayes classifier

The values x_1, x_2 to x_n in Figure 2.1 represent the features and C is the outcome when these features have been combined through Baye's theorem. A clear disadvantage of a naïve Bayes classifier is that it declares features as mutually independent in which in some cases features can be dependent. [Pham and Ruz 2009] gives a more detailed tutorial of the naïve Bayes classifier and Bayesian networks.

2.4 Support Vector Machines

In machine learning, a support vector machine is a supervised learning algorithm that finds a hyperplane that divides two classes by the greatest margin in between them. Traditionally, it is not easy to separate data linearly, but with SVMs the data can be casted onto a higher dimensional space where the data is separable. All SVMs must find an optimal hyperplane that linearly separates the data. This must also give the maximum margin between the support vectors. SVMs can also be extended to solve tasks of regression, this is where the system is trained to give out a numerical value rather than a binary classification. Figure 2.2 illustrates a support vector machine.

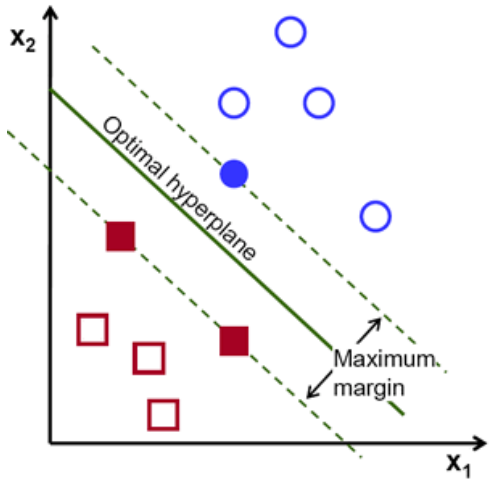

Figure 2.2: A support vector machine

Figure 2.2 shows the optimal hyperplane that gives the largest minimum distance to the training example. The maximum margin ensures that the optimal hyperplane does not pass too close to the points as this results in noise and it will not generalize correctly. We consider some of the advantages and disadvantages of support vector machines. SVMs are more effective when it comes to higher dimensional spaces. They are also memory efficient as they use a subset of the training points in the decision function. Given that the features are greater than the number of samples, the method is more likely to give poor performances. [Isa et al. 2008] gives a more detailed account of support vector machines.

2.5 Decision Trees

Decision trees fall under the realm of decision tree learning. These are the most widely used in practical methods for inductive inference [Mitchel 1997]. This is a method of approximating discrete-valued functions and it is more robust. Decision trees classify instances by sorting them down the tree from root to leaf node and this way the classification of an instance is provided for. It must be noted that each node in the tree specifies a test of some attribute of the instance.

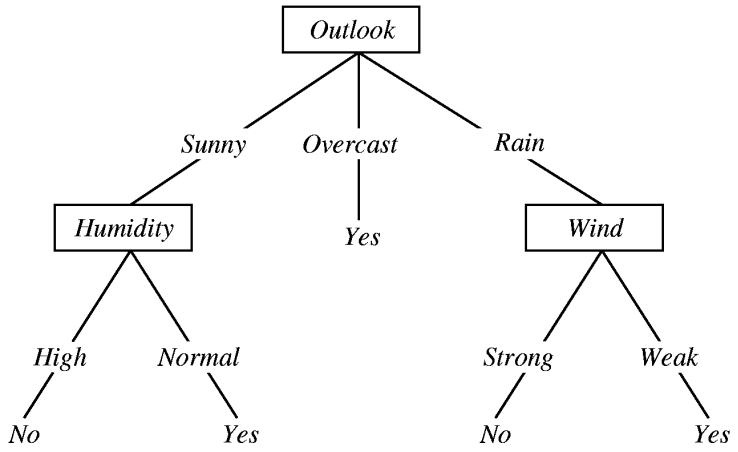

Figure 2.3: A decision tree

In Figure 2.3, we have a decision tree for a concept of deciding whether to play tennis on not depending on the given weather on the day. An example is classified by sorting it through the tree to the correct leaf node, the tree then returns the correct class label that is associated with this leaf, this can be a Yes or a No. If the given day was a Saturday, the tree would classify according to whether or not they are suitable for playing tennis.

2.6 Feature Extraction Techniques

In machine learning, typically each problem has data that comprises of features and these describe each entry and they are commonly referred to as attributes. In some instances the data can be in 3 dimensions which is a simpler to work with and in difficult cases the data might be in a larger vector space which necessitates feature extraction. Feature extraction is the process of deriving values from a larger set of attributes in order to construct a more meaningful set of features that is less redundant and leads to better interpretation. Given data that is too large to process and is redundant, it is imperative to apply feature selection or extraction in order to build more accurate models. Weka has a built in tool set that allows for seamless feature extraction and this ranges from Principal Component Analysis, Information Gain, Correlation etc. In this section we focus of three attribute evaluators that have their corresponding search methods.

In Weka terminology, the attribute evaluator is a tool that takes each feature in your dataset and it evaluates it in the context of the output which is the class/target [Remco R. Bouckaert 2013]. On the other hand, the search method is a tool that is used to navigate different permutations of the attributes in order to end up with a short list of chosen features [Remco R. Bouckaert 2013].

2.6.1 Correlation Attribute Evaluator

A popular attribute selector is the Correlation Attribute Evaluator which must go hand in hand with the Ranker Search Method. Formally, this is known as the Pearson's correlation coefficient and it determines how the features correlate to the output which is the class. After these features are correlated to the output, the Ranker Search Method ranks the features according to how much they affect the output.

2.6.2 Information Gain Attribute Evaluator

This technique works by calculating the information gain, which is formally known as the entropy for each attribute. The values of the entropy vary from 0 which corresponds to no information gain and 1 which is for maximum information gain. Attributes that have maximum information gain or those that contribute more will be selected.

2.6.3 Wrapper Subset Evaluator

The wrapper subset evaluator is also known as the learner based feature selection technique. This technique works by using a generic learning algorithm and it does an evaluation of its performance on the provided dataset with different subsets of selected attributes. The subset that results in the best performance is taken as the selected subset. One can choose to use a decision tree as an attribute evaluator and from there they have the freedom to choose between a GreedyStepwise function or a Ranker function as a Search Method.

2.7 Conclusion

This chapter focused on attempting to provide the background of the research topic and the related work. It starts off by reviewing the literature that is related to the research topic. The literature focused more on predicting student success in first year computer science. The first approach that was used by authors like [Butcher and Muth 1985], [Campbell and McCabe 1984] and [Gathers 1986], studied how a different combination of factors like high school grades and aptitude tests were used to construct a model for predicting student success. The second approach that was reviewed focused on using single factors to build models for prediction purposes. Authors like [Capstick et al. 1975] and [Tewari 2014] used high school mathematics scores to predict the probability of performing well in first year. Their assumption is that it has a positive correlation to student success and their results back up their claim. In [Rauchas et al. 2006], the authors used another single predictor, in this case it being high school English scores and they found a correlation to student success. The chapter ends off by giving a brief introduction of the naïve Bayes classifier, support vector machines, decision trees and Feature Extraction Techniques that will feature in the research methodology of the project.

Chapter 3
Research Methodology

3.1 Introduction

Predicting the success of a first year computer science student is affected by many variables as demonstrated in [Butcher and Muth 1985]. The background to the problem has been presented in the previous chapter and it also lays out the different approaches that have been used by other authors who focused on first year performance by using high school results. It must be noted that the previous authors who are surveyed in the literature had their focus on predicting success in first year computer science and other first year courses. The project will be taking this work further by trying to predict the completion of a science degree based only on first year marks. The naïve Bayes classifier, support vector machines and decision trees can be used to build classification models for this particular problem area as presented in Chapter 2. These classification models will use the first year marks as training data and they will also be used to test the hypothesis.

In this chapter we describe the research hypothesis, the objectives of the research will be formally stated as well. In section 3.2, the research hypothesis and the research question are defined and presented. Section 3.3 details the methodology that is going to be followed to the test the hypothesis and answer the research question. This will be followed by a brief summary in section 3.4 of the important points of this chapter.

3.2 Research Hypothesis

There are a lot of algorithms that can be used to solve classification problems in machine learning. As discussed in Chapter 2, naïve Bayes, support vector machines and decision trees can be used in this instance. The research project will be focusing on finding a solution to the question of: **Is there a correlation between first year academic results and the number of years a student will take to complete their degree?** From this question we can give the following hypothesis: **Students who perform well in their first year of study and who elect complementing courses from second year obtain their science degrees in record time (3 years).**

3.3 Methodology

In order to test and verify our hypothesis, several experiments were conducted. In this section we go into detail of how the experiments were conducted. The methodology will contain 4 major phases of the research project.

3.3.1 Phase 1: Data Collection

The data was gathered from the University of the Witwatersrand, Johannesburg, from the department of Academic Information Systems Unit (AISU). This was done with the help of the supervisor who liaised with the relevant staff member. The data contained a set of student marks for a period of 7 years, from 2010 to 2017. Every discipline of study from the Faculty of Science undergraduate degrees was included. In a period of seven years a couple of degrees should have been completed, this is a reasonable data set to use for classification. This data set contained a good mixture of students and some were students who have enrolled in the current

academic year. For each student who has a complete set of academic records from first year to final year, these results were used to build a model. This model was later used with only first year marks to calculate the probability of success for students who haven't completed their degrees.

3.3.2 Phase 2: Data Preparation

In this phase the necessary exploratory data analysis and data preprocessing was performed on the given dataset. The data was received in a spreadsheet format and this had to be converted into a format that Weka can work with. Weka can work with comma separated values (csv) or attribute-relation format files (arff), the spreadsheet was converted into a csv file. The spreadsheet was read into an iPython notebook which uses Python and a package called pandas to work with stored data, this platform makes it easy to perform exploratory data analysis. The student numbers were anonymized in order to protect the students identity. The dataset contained students who did their year of study 1,year of study 2 and year of study 3 and in our case since we were working with year of study 1 so we dropped the other records (YOS2 and YOS3). After dropping the unnecessary rows, the rows that were left contained the subject and its mark for each student, say student x, the first 6 rows would be student x with the 6 subjects they did in first year with the corresponding marks and Progression Outcome Type. Since each student had more than one row identifying them we performed a transpose on the given table so that each record is one unique student with the columns having the subject matters and marks. The last column was the class label which was either a Yes or a No for the Progression Outcome Type. There was 216 features in total after the data was prepared correctly. The data was then saved onto a Weka readable format in order to run the supervised learning algorithms. The data preparation phase of the methodology is important so it was handled with care as this might have had an adverse effect the outcome of the research. The data was not required to be split as Weka did this automatically.

3.3.3 Phase 3: Training

After the implementation of the classifiers the focus shifted to evolving the models in order to improve their accuracy. This was done by performing feature extraction and selection techniques. The splitting of the data was necessitated by the fact that some models tend to overfit. A reasonable split is 66% for training and 33% for testing. The naïve Bayes should have learned the probabilities of each variable at the end of training. The same will apply to the Support Vector Machine that has the required hyperplanes and it must also learn the correct weights, have the required optimal plane and correctly chosen support vectors. The decision tree will construct a tree from top down and this is done traditionally using the J48 algorithm in Weka. After performing the feature extraction techniques we chose one supervised learning algorithm in order to see if the accuracies were comparable to the generalized models. Training was done on the Correlation, Information Gain and Wrapper Evaluation attributes. The tailored degrees have their specified subjects so for each category the training was performed as well.

Figure 3.1: Phases of the research methodology

3.3.4 Phase 4 : Testing

After learning the parameters the parameters of the classifiers, these will be used to either confirm or reject the hypothesis through testing. The testing will happen automatically as the 10 fold cross validation parameter setting enables for each model to build multiple models with different data splits for training and testing. This approach allows us to have a more robust model that is not biased and it caters for all variation in the dataset. Every naïve Bayes has an error associated with it, the error that will be derived after performing testing through cross validation of the model will give us a good indication of how well our classifier functions. The same applies for the support vector machine and the decision tree and detailed error results will be given for each in the results section in Chapter 4.

3.4 Conclusion

The focus of this chapter was on the methodology of the proposed research. The research question and hypothesis were formally stated, as were the essential steps that have to be taken in order to accept or reject the hypothesis. The steps that will be taken in order to verify or reject the hypothesis are also outlined. The following chapter will provide a detailed plan of how the experiment will be conducted.

Chapter 4
Experiments

4.1 Results

The previous chapter we discussed how the hypothesis of the research project would be tested and in this chapter we look at the performance of the machine learning techniques that were mentioned in chapter 2. Firstly, we do simple plots of the data distribution in order to have a better picture of our data and then we evaluate the performance of the 3 main techniques. This evaluation occurs on the full dataset that has 216 features and the results will be subsequently recorded. We then perform feature extraction using the techniques that were also mentioned in chapter 2, we use one chosen algorithm in order to observe if there's any improvement on the model after feature extraction. The last batch of experiments involve training models for tailored degrees in the Faculty of Science this is done due to the fact that the initial models we have generalize students in the Faculty. The tailored degrees fall under four categories which are; Physical Sciences, Mathematical Sciences, Biological and Life Sciences and the last one is Earth Sciences.

4.1.1 Graphs

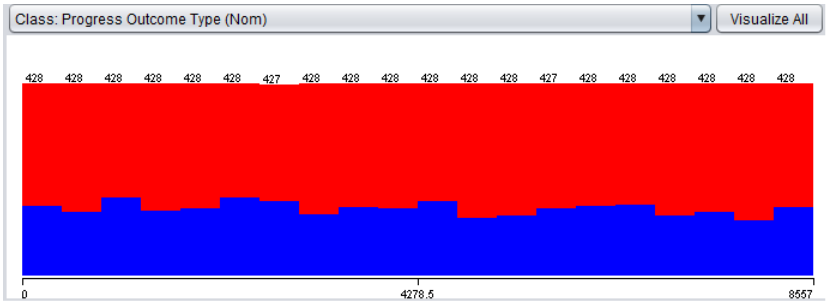

Figure 4.1: Stacked plot of Progress Outcome Type

The stacked plot in Figure 4.1 represents the distribution of the Progression Outcome Type, in Weka this is referred to as the class label. The color blue represents students that have successfully completed their degrees in record time (3 years) and red represents the class of students that have not completed their degrees. The total number of unique students is 8557 for a period of 7 years.

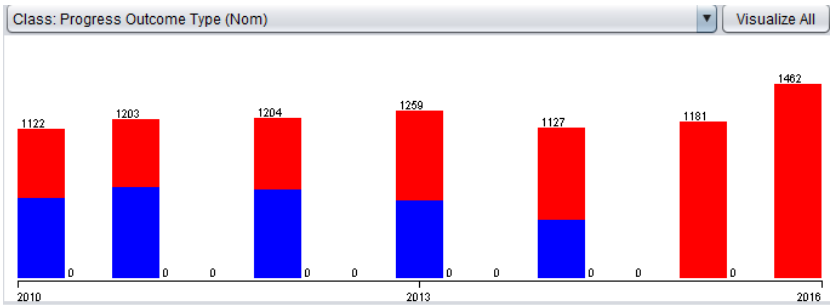

Figure 4.2: Stacked bar plot of years

Figure 4.2 represents a stacked bar plot that compares the Progress Outcome Type for each year from 2010 until 2016. Blue represents students that have completed their degrees in record time and the color red represents students that have not completed their degrees. The last two years, 2015 and 2016 have students that have not completed their degrees which is a strange phenomenon because we would expect that in every 3 year cycle there would be students that complete their degrees. Students that enrolled in 2014 should have completed their degrees in 2016, and the same applies to students that enrolled in 2013, these students should have completed their degrees in 2015. This strange observation in Figure 4.2 can be accounted to the second phase of the research methodology, we might have dropped students who completed their degrees in 2015 and 2016 by mistake. This might have been caused by the format in which we received the dataset.

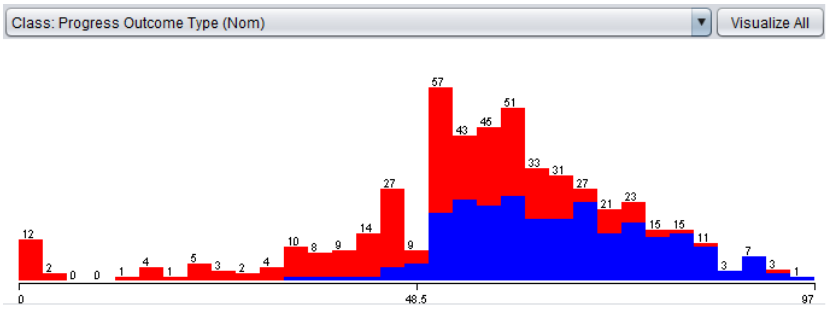

Figure 4.3: Computer Science I mark distribution

In Figure 4.3, we have a stacked plot of the distribution of marks in Computer Science I. The plot resembles a normal distribution (Gaussian) which is what we expected for a set of marks for any given course. There are outliers in the Computer Science I marks and these are located at the far left in the plot.

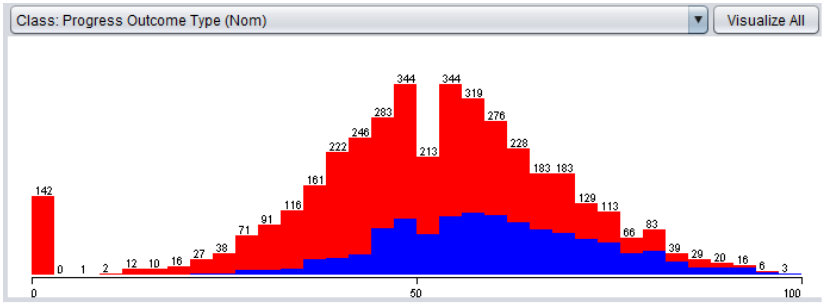

Figure 4.4: Mathematics I mark distribution

Figure 4.4 illustrates a stacked plot of the distribution of marks in Mathematics I major. The plot resembles a normal distribution (Gaussian) more than the Figure 4.3, this is because we have more data points than before. There are also outliers at the far left of the stacked plot.

Given that there is 216 possible courses that are features of the data, the same phenomenon of normally distributed marks can be generalized to them but we plot only 2 diagrams for purposes of illustration.

4.1.2 Predicting the completion of a general Bachelor of Science's degree based on first year marks

Table 4.1: Table of algorithm accuracies

Algorithm	**Correctly Classified**	**Incorrectly Classified**
Naïve Bayes	7391 [$\mathbf{86.3636}\%$]	1167 [$\mathbf{13.6364}\%$]
Support Vector Machine	7446 [$\mathbf{87.0063}\%$]	1112 [$\mathbf{12.9937}\%$]
Decision Tree	5616 [$\mathbf{65.6228}\%$]	2942 [$\mathbf{34.3772}\%$]

Table 4.1 represents the results of the 3 different machine learning algorithms that were used to test the hypothesis. The support vector machine outperformed the naïve Bayes and the decision tree. The results of the decision tree are the worst from the first two algorithms as an accuracy of 65.62% cannot be deemed as reasonable given that the support vector machine achieved an accuracy of 87%.

Table 4.2: Confusion Matrix of the naïve Bayes

a	b	
2622	363	a = True
804	4769	b = False

Table 4.3: Confusion Matrix of the support vector machine

a	b	
2380	605	a = True
507	5066	b = False

Table 4.4: Confusion Matrix of the decision tree

a	b	
2979	6	a = True
2936	2637	b = False

The Tables 4.2, 4.3 and 4.4 represent the Confusion Matrices for the respective algorithms that are in Table 4.1. These tables describe the performance of each algorithm and this is another way of representing the accuracy of each algorithm. It is clear that the Confusion Matrix of the support vector machine is the one that classifies the students better to the other two algorithms and these algorithms used a total of 216 courses as their features.

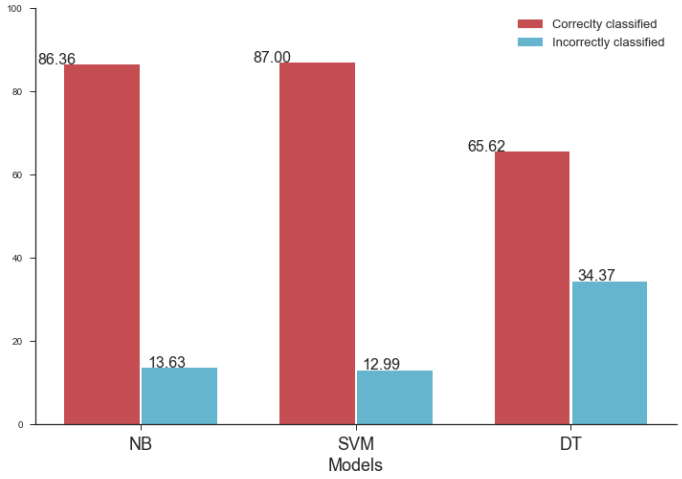

Figure 4.5: Stacked bar plot of years

In Figure 4.5 we have a bar plot that compares the accuracy of the three algorithms. The graphical representation gives a better picture than Table 4.1 and the Confusion Matrix tables. Here we are able to visually analyze the accuracy of each algorithm. The grouped bar plots are for the naïve Bayes which is labeled as NB, the support vector machine which is labeled as the SVM and the decision tree which is labeled as DT. The legend of the plot shows that the red color represents the category of the correctly classified instances and the cyan represents the incorrectly classified instances.

4.1.2.1 Results with Feature Extraction using naïve Bayes

Table 4.5: Table of the different feature extraction techniques

Feature Extraction Technique	**Correctly Classified**	**Incorrectly Classified**
Correlation	7393 [$\mathbf{86.387}\%$]	1165 [$\mathbf{13.613}\%$]
Information Gain	7387 [$\mathbf{86.3169}\%$]	1171 [$\mathbf{13.6831}\%$]
Wrapper Evaluator	6968 [$\mathbf{81.4209}\%$]	1590 [$\mathbf{18.5791}\%$]

Table 4.5 represents the results of the naïve Bayes classifier that was used on the 3 feature extraction techniques. Correlation applied the Ranker Search Method and obtained the following features; Encrypted Student Number, Calender Instance Year, APPM1006, BIOL1000, COMS1000, ECON1009, MATH1034, PHYS1000,

ACCN1000, ECON1000, BIOL1006, ECON1008, GEOL100, CHEM1012 and Progress Outcome Type.

Information Gain applied the Ranker Search Method and obtained the following features; Calender Instance Year, Encrypted Student Number, CHEM1012, MATH1034, BIOL1000, MATH1036, APPM1006, PHYS1000, ACCN1000, PHYS1001, GEOL1000, BIOL1006, ECON1008, ECON1009, COMS1000 and Progress Outcome Type.

The Wrapper Evaluation feature extraction technique applied the GreedyStepwise Method and obtained the following features; Calender Instance Year, APPM1006, ACCN1000, BIOL1000, CHEM1012, ECON1018, MATH1010, MATH1034, STAT1002 and Progress Outcome Type.

The features were used separately to determine whether the naïve Bayes classifier would perform better after using feature extraction techniques. The correlation technique out performed the other two feature techniques, it achieved an accuracy of 86.387% and the Wrapper Evaluator technique had the worst performance with an accuracy of 81.42%

Table 4.6: Confusion Matrix of the naïve Bayes (Correlation)

a	b	
2696	289	a = True
876	4697	b = False

Table 4.7: Confusion Matrix of the naïve Bayes (Information Gain)

a	b	
2640	345	a = True
826	4747	b = False

Table 4.8: Confusion Matrix of the naïve Bayes (Wrapper Evaluator)

a	b	
2437	548	a = True
1042	4531	b = False

The Tables 4.6, 4.7 and 4.8 represent the Confusion Matrices for the respective algorithms that are in table 4.5. These tables describe the performance of the naïve Bayes when applied on the features that have been extracted for each technique. The Correlation Confusion Matrix demonstrates that it classifies the students better compared to the other feature extraction techniques and its accuracy of 86.387% is comparable to the initial accuracy of the naïve Bayes for the general case which is 86.36%.

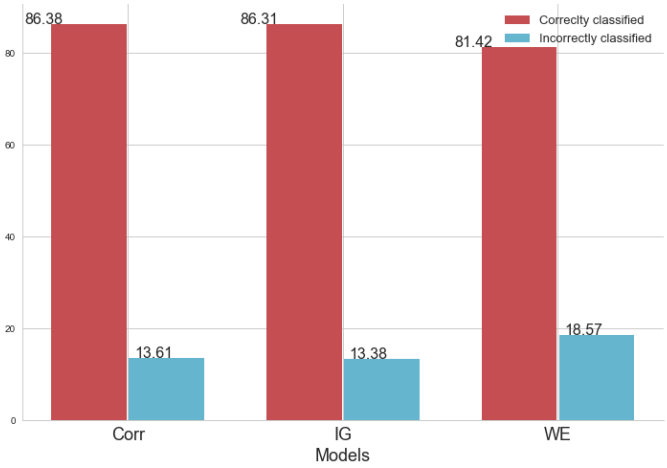

Figure 4.6: Bar plot

In Figure 4.6 we have a bar plot that compares the accuracy of the three feature extraction techniques. The graphical representation gives a better picture than Table 4.5 and the Confusion Matrix tables. Here we are able to visually analyze the accuracy of each feature extraction technique. The grouped bar plots are for the Correlation technique which is labeled as Corr, the Information Gain technique which is labeled as the IG and the Wrapper Evaluator technique which is labeled as WE. The legend of the plot shows that the red color represents the category of the correctly classified instances and the cyan represents the incorrectly classified instances. The feature extraction techniques were used to demonstrate as whether we could test the hypothesis better by utilizing other methods on one of the supervised learning algorithms. It has been shown that these techniques performed comparably to the generalized case.

4.1.3 Predicting the completion of a field specific Bachelor of Science degree based on first year marks

Table 4.9: Table of the field specific degrees

Degree	Correctly Classified	Incorrectly Classified
Physical Sciences	7392 [$\mathbf{86.3753}\%$]	1166 [$\mathbf{13.6247}\%$]
Mathematical Sciences	7198 [$\mathbf{84.1084}\%$]	1360 [$\mathbf{15.8916}\%$]
Biological and Life Sciences	7357 [$\mathbf{85.9663}\%$]	1201 [$\mathbf{14.0337}\%$]
Earth Sciences	7396 [$\mathbf{86.4221}\%$]	1162 [$\mathbf{13.5779}\%$]

Table 4.9 represents the results of the field specific degrees in the Faculty of Science. The naïve Bayes classifier was used for each of the 4 fields. The correctly classified instances for each of the fields have comparable accuracies and these are also comparable to the generalized case which uses the 216 features. Given that some students when they start their first year's of study they pursue field specific degrees, the model was also tested for field specific degrees in order to see whether we could test our hypothesis for these cases.

Table 4.10: Confusion Matrix of the naïve Bayes (Physical Sciences)

a	b	
2753	232	a = True
934	4639	b = False

Table 4.11: Confusion Matrix of the naïve Bayes (Mathematical Sciences)

a	b	
2771	214	a = True
1146	4427	b = False

Table 4.12: Confusion Matrix of the naïve Bayes (Biological and Life Sciences)

a	b	
2784	201	a = True
1000	4573	b = False

Table 4.13: Confusion Matrix of the naïve Bayes (Earth Sciences)

a	b	
2746	239	a = True
923	4650	b = False

The Tables 4.10, 4.11, 4.12 and 4.13 represent the Confusion Matrices for the respective field specific degrees that are in Table 4.9. These tables describe the performance of the naïve Bayes when applied on the 4 field specific degrees. The naïve Bayes performs better on the Earth Sciences field degree. The courses contained under the Physical Sciences stream are; MATH1034, MATH1036, PHYS1000, CHEM1012, ECON1002, PHYS1015.

The courses contained under the Mathematical Sciences stream are; MATH1034, MATH1036, PHYS1000, CHEM1012, STAT1003, STAT1002, APPM1006, APPM1021, COMS1000.

The courses contained under the Physical Sciences stream are; BIOL1000, CHEM1012, APPM1006, PSYC1001, PHIL1001.

The courses contained under the Physical Sciences stream are; GEOL1000, MATH1034, MATH1036, CHEM1012, GEOG1000, ARCL1000. It must be noted that some courses that are included in some of the fields are courses that were chosen at the discretion of a given student.

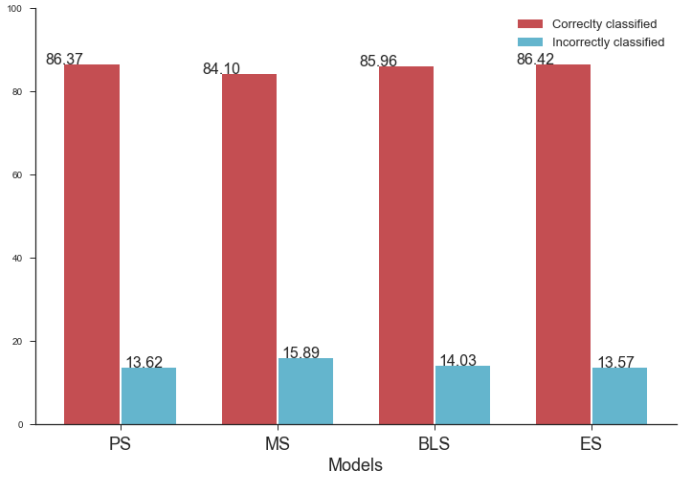

Figure 4.7: Bar plot

In Figure 4.7 we have a bar plot that compares the accuracy of the the naïve Bayes when used on each of the four field specific degrees in the Faculty of Science. The graphical representation gives a better picture than Table 4.9 and the Confusion Matrix tables. Here we are able to visually analyze the accuracy of each field specific degree when the naïve Bayes classifier is applied. The grouped bar plots are for the Physical Sciences field which is labeled as PS, the Mathematical Sciences field which is labeled as the MS, Biological and Life Sciences field which is labeled as the BLS and the Earth Sciences field which is labeled as ES. The legend of the plot shows that the red color represents the category of the correctly classified instances and the cyan represents the incorrectly classified instances. The degree specific fields were used to demonstrate as whether we could test the hypothesis better by focusing on courses that were tailored for certain degrees. The naïve Bayes classifier had a comparable performance to the generalized case and the feature extraction case.

4.2 Discussion

The previous subsections focused on the qualitative results of the research project. We discuss the obtained results in the above subsections, we preform comparisons of them and we elaborate on the limitations of the models.

The theory of naïve Bayes, as discussed in detail in Chapter 2 and the models that were built for the general case and the specific cases, these obtained reasonable accuracy which we would assume can predict the success of students using only their first year marks. It must be noted that naïve Bayes is premised from the fact that attributes are assumed to be conditionally independent from each other as they utilize Baye's Theorem. The results indicate a good predication rate for the general case as we have an accuracy of 86.3636% for the general case. The support vector machine machine achieved an accuracy of 87% on the general case that has a total of 216 features and the algorithm took some considerable time to be run on compared to the naïve Bayes. The decision tree had the worst performance and it achieved an accuracy of 65.6228%. The decision tree learning algorithm uses the Information Gain technique that is similar to the approach of the Information Gain feature extraction technique but this algorithm performed dismally as compared to that feature extraction technique. Given that the total number of features is 216, the algorithm might not have been able to compute accurately

the Information Gain for each.

There was three feature extraction or selection techniques that were utilized and these were Correlation, Information Gain and Learner Based Feature (Wrapper Evaluator) Selection techniques. These achieved accuracies of 86.387%, 86.3169% and 81.4209% respectively and these are comparable to the general results of the naïve Bayes classifier accuracy in the general case. The model of the naïve Bayes classifier was further extended to the tailored degree cases and these achieved accuracies of 86.3753% , 84.1084%, 85.9663% and 86.4221%, these results are comparable to the general case of the naïve Bayes model. Each algorithm in the results section made use of 10 fold cross validation, cross-validation is described as a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. This model validation technique is used when the goal for statistical analysis is prediction. This technique works by partitioning the original dataset into two samples, the training set and the testing set (validation set). This technique allows for models to be tested by making use of the full training set by means of repeated sampling.

It can be argued that the models obtained in this section represented results that are reasonable as 10 fold cross validation was used, other discrepancies can be accounted to how the algorithms are optimized, the sample size and class label distribution in our data. The algorithms described in this section were able to test the hypothesis successfully and the support vector machine achieved the best accuracy among the others.

4.3 Conclusion

This chapter discusses some of the major results we have found throughout this research. In section 4.1.1 we have the graphs that illustrate our dataset after the data preparation phase was implemented. Section 4.1.2 has the results of the general Bachelor of Science degree and 3 machine learning techniques were applied on the 216 features. Section 4.1.2.1 has results for the feature extraction techniques that compare whether these is an improved performance of the naïve Bayes after the feature extraction techniques have been applied. Section 4.1.3 has the results of the field specific degrees and how the naïve Bayes performs for these degrees.

Chapter 5
Conclusion

Our primary objective of this research was to build a model that can successfully predict the completion of a student's Science degree based only on their first year marks. To achieve this, we first conducted a research on different machine learning techniques, analyzed their advantages and limitations of each. According to the literature survey, we selected the naïve Bayes classifier, support vector machine and the decision tree. These have reasonable performances in theory that is why they were selected to be used in this research. Data preprocessing was used in order to package the data into the correct format and the feature extraction techniques were used in order to minimize the effect of over-fitting. Following a certain methodology, we conducted experiments on the selected models and on their variations. The methodology we performed can be easily extended to build other models that can be used for prediction purposes.

To evaluate the predictive performance of the models, this research applied a combination of features to the given models and their accuracies were determined. The computation time of each model is comparable except to the support vector machine that took some time do build. The experiment results show that our hypothesis is true and the support vector machine is considered as the most efficient model that produces results of reasonable accuracy. This was followed by the naïve Bayes and the decision tree had the worst performance. The naïve Bayes classifier was chosen purely on discretion when it was used for the feature extraction techniques and the field specific degrees. The support vector machine could have been chosen as well for this purpose as it had a comparable accuracy. The results of the feature extraction techniques and the field specific degrees were comparable to the general case.

It must be noted that this particular research was focused on the qualitative data that was received from the Registrar's office in the Faculty of Science in the University of Witwatersrand, Johannesburg. A lot of factors were not taken into account when building the necessary models, i.e. the social circumstances of a particular student, whether a student is on financial aid and personal preferences and this affects the future work of the research. It can also be argued that the gender of the student might also affect the performance of a particular student, so this can improve future models.

It would be interesting to apply these models on new and bigger dataset as this would give us a better picture of how these perform. It would be also interesting to apply these models on data that has different structure.

Further work could be done to build other machine learning models that have better or supreme accuracies as the ones used in this project. Deep neural networks or Bayesian networks could be used and these algorithms cater for more features and they are able to model them precisely. These models do not make a lot of assumptions about our data and they can also be used to achieve the research objectives.

Bibliography

Butcher, D. F. and Muth, W. (1985). Predicting performance in an introductory computer science course. In *Communications of the ACM Volume 28 Issue 3*, pages 263–268.

Campbell, P. F. and McCabe, G. (1984). Predicting the success of freshmen in a computer science major. In *Communications of the ACM Volume 27 Issue 11*, pages 1108–1113.

Capstick, C. K., Gordon, J. D., and Salvadori, A. (1975). Predicting performance by university students in introductory computing courses. In *ACM SIGCSE Bulletin Volume 7 Issue 3*, pages 21–29.

Gathers, E. (1986). Screening freshmen computer science majors. In *ACM SIGCSE Bulletin Volume 18 Issue 3*, pages 44–48.

Ghahramani, Z. (2001). An Introduction to Hidden Markov Models and Bayesian Networks. In *International Journal of Pattern Recognition and Artificial Intelligence*, pages (15):9–42.

Hostetler, T. R. (1983). Predicting student success in an introductory programming course. In *ACM SIGCSE Bulletin*, pages 40–43.

Isa, D., Lee, L. H., Kallimani, V., and Rajkumar, R. (2008). Text document preprocessing with the bayes formula for classification using the support vector machine. *IEEE Transactions on Knowledge and Data engineering*, 20(9):1264–1272.

Mitchel, T. M. (1997). Decision Tree Learning. In *Machine Learning*, pages (3):52–57.

Pham, D. T. and Ruz, G. A. (2009). Unsupervised training of bayesian networks for data clustering. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 465(2109):2927–2948.

Powelli, D. (2002). Introduction to Support Vector Machines.

Rauchas, S., Rosman, B., Konidaris, G., and Sanders, I. (2006). Language performance at high school and success in first year computer science. In *ACM SIGCSE Bulletin Volume 38 Issue 1*, pages 398–402.

Remco R. Bouckaert, Eibe Frank, M. H. (2013). In *Weka Manual for version 3.6.9*.

Scott, I. (2013). A proposal for undergraduate curriculum reform in South Africa: The case for a flexible curriculum structure. In *Council on Higher Education*.

Tewari, D. D. (2014). Is matric math a good predictor of student's performance in the first year of university degree? A case study of faculty of management studies, University of Kwazulu-Natal, South Africa. In *International Journal of Science Education*, pages 233–237.

Thurow, L. C. (1972). Education and economic equality. In *The Public Interest 28*, page 66.