

Using machine learning to support students' academic decisions

استخدام التعلم الآلي لدعم القرارات الأكاديمية للطلاب

by

AISHA QASIM GHAZAL FATEH ALLAH

**Dissertation submitted in fulfilment
of the requirements for the degree of
MSc INFORMATICS
(KNOWLEDGE AND DATA MANAGEMENT)**

at

The British University in Dubai

March 2019

DECLARATION

I warrant that the content of this research is the direct result of my own work and that any use made in it of published or unpublished copyright material falls within the limits permitted by international copyright conventions.

I understand that a copy of my research will be deposited in the University Library for permanent retention.

I hereby agree that the material mentioned above for which I am author and copyright holder may be copied and distributed by The British University in Dubai for the purposes of research, private study or education and that The British University in Dubai may recover from purchasers the costs incurred in such copying and distribution, where appropriate.

I understand that The British University in Dubai may make a digital copy available in the institutional repository.

I understand that I may apply to the University to retain the right to withhold or to restrict access to my thesis for a period which shall not normally exceed four calendar years from the congregation at which the degree is conferred, the length of the period to be specified in the application, together with the precise reasons for making that application

Signature of the student

COPYRIGHT AND INFORMATION TO USERS

The author whose copyright is declared on the title page of the work has granted to the British University in Dubai the right to lend his/her research work to users of its library and to make partial or single copies for educational and research use.

The author has also granted permission to the University to keep or make a digital copy for similar use and for the purpose of preservation of the work digitally.

Multiple copying of this work for scholarly purposes may be granted by either the author, the Registrar or the Dean only.

Copying for financial gain shall only be allowed with the author's express permission.

Any use of this work in whole or in part shall respect the moral rights of the author to be acknowledged and to reflect in good faith and without detriment the meaning of the content, and the original authorship.

ABSTRACT

Making the right decision for students in higher education is vital, as it has a great influence on their study, career, life, and eventually, the whole society. Predicting the future performance of students can inform their choice of majors, concentrations, and courses. It also helps teachers and advisors provide the necessary support to students as needed.

While many studies address the issue of predicting students' performance, they mainly predict student performance at one stage of their study only. This work proposes a framework for assisting students in their decision throughout their study journey. At enrollment, this work predicts a student's GPA in different majors using enrollment data such as high school average, placement test results, and IELTS score. After completing their first year, this work predicts student's GPA in different concentrations using grades of Year 1 courses. At any point of time after the student finishes some courses, a user-based collaborative filtering approach using K-Nearest Neighbor is used to predict a student's grade in a future course. This approach uses other students' grades to make a prediction.

This research tests and compares the performance of Decision Trees, Random Forests, Gradient-Boosted trees, and Deep Learning machine learning regression algorithms to predict student GPA. Furthermore, the strongest predictors of student's GPA are identified at each stage. Gradient Boosted Trees performed the best when predicting student's Major GPA, while Deep Learning performed the best for predicting Concentration's GPA.

نبذة مختصرة

يعد اتخاذ القرار الصائب لطلاب التعليم العالي أمرًا بالغ الأهمية ، حيث إنه له تأثير كبير على دراستهم ومهنتهم وحياتهم ، وفي النهاية على المجتمع بأسره. يمكن للتنبؤ بالأداء المستقبلي للطلاب أن يساعد في اختيار التخصصات الرئيسية والفرعية بالإضافة لاختيار المواد. كما أنه يساعد المعلمين المرشدين على توفير الدعم اللازم للطلاب حسب حاجتهم.

بينما تتناول العديد من الدراسات مسألة التنبؤ بأداء الطلاب ، إلا أنها تتنبأ بشكل رئيسي بأداء الطلاب في مرحلة واحدة من دراستهم فقط. تقترح هذه الدراسة إطارًا متكاملًا لمساعدة الطلاب في اتخاذ قرارهم طوال فترة دراستهم الجامعية. فعند التسجيل ، تتوقع هذه الدراسة معدل الطالب في التخصصات الرئيسية المختلفة باستخدام بيانات التسجيل مثل معدل الدرجات في الثانوية العامة، ونتائج اختبار تحديد المستوى ، ودرجة اختبار IELTS. بعد الانتهاء من عامهم الأول ، تتوقع هذه الدراسة معدل الطالب في التخصصات الفرعية المختلفة باستخدام درجات مواد السنة الأولى. وفي أي وقت بعد انتهاء الطالب من بعض المواد ، يتم استخدام نهج ترشيح تعاوني قائم على المستخدم (user-based Collaborative Filtering) باستخدام طريقة البحث عن أقرب الجيران (K-Nearest Neighbor) للتنبؤ بتقدير الطالب في المواد المستقبلية. يستخدم هذا المنهج درجات الطلاب الآخرين في التنبؤ.

يقوم هذا البحث باختبار ومقارنة أداء الخوارزميات المختلفة لتعلم الآلة مثل أشجار القرار (Decision Trees) ، والغابات العشوائية (Random Forests) ، وأشجار التدرج المعزز (Gradient Boosted Trees) ، والتعلم العميق (Deep Learning) للتنبؤ بالمعدل التراكمي للطلاب. علاوة على ذلك ، يتم تحديد أقوى العوامل التي تساعد في توقع الأداء الأكاديمي للطلاب في كل مرحلة. كان أداء أشجار التدرج المعزز (Gradient Boosted Trees) هو الأفضل عند توقع معدل الطالب في التخصص الرئيسي ، بينما كان أداء التعلم العميق (Deep Learning) هو الأفضل للتنبؤ بمعدل الطالب في التخصصات الفرعية.

DEDICATION

I dedicate this work to my students; may you *all* be successful; *always*!

I also dedicate this work to my family; thank you for always being there for me; no words can express how blessed I feel for having you in my life.

Aisha Ghazal

ACKNOWLEDGEMENTS

All praise is due to Allah, the Worthy of all the praises and compliments.

I want to express my deepest appreciation to Professor Sherief Abdallah who introduced me to this exciting field. When he explains, he simplifies complex topics in an inspiring way for me as an educator. He engages our minds and asks questions that encourage us to think critically. Without his guidance and feedback, this dissertation would not have been possible.

I would like to also thank Dr Khaled Shaalan for his continuous support and guidance, in addition to all my teachers throughout my life. If it were not for them, I would have been illiterate.

Special thanks go to my friend Ghazala. Having her with me through this journey was a great blessing.

Table of Contents

1.	Introduction	1
1.1.	Problem statement	2
1.2.	Research Objectives.....	3
2.	Background	6
2.1	Machine Learning.....	6
2.2	Regression Algorithms used.....	9
2.2.1	Decision Trees (DT) for Regression	9
2.2.2	Random Forests	10
2.2.3	Gradient-Boosted Trees	11
2.2.4	Deep Learning	12
2.2.5	Collaborative Filtering using K-Nearest Neighbor.....	13
2.3	Similarity measures.....	15
2.3.1	Euclidean Distance (ED)	16
2.3.2	Pearson Correlation Coefficient (PCC)	16
2.4	Evaluation Metrics: RMSE and MAE	17
3.	Literature Review.....	19
3.1	Predicting Course Grades.....	20

3.2 Predicting performance in a Major/Concentration	23
3.3 Predicting Future performance.....	24
3.4 Predicting Dropout.....	26
4. Methodology and Results	28
The Overall Process:.....	31
4.1 Predicting Major GPA (at enrollment)	33
4.1.1 Data	34
4.1.2 Preprocessing.....	36
4.1.3 Algorithms.....	37
4.1.4 Results	40
4.1.5 Predictors	42
4.2 Predicting Concentration GPA (after year 1)	46
4.2.1 Data	47
4.2.2 Preprocessing.....	48
4.2.3 Algorithms.....	49
4.2.4 Results	49
4.2.5 Predictors	53
4.3 Predicting Course Grade	59
4.3.1 Data	59
4.3.2 Preprocessing.....	60

4.3.3	Algorithm	62
4.3.4	Results	64
5.	Conclusion and Future Work	67
6.	Bibliography	71
Appendix A	77

Table of Illustrations

Figure 1: Academic standing of students.....	3
Figure 2: Random Forest.....	11
Figure 3: Gradient Boosted Trees	12
Figure 4: Deep Learning	13
Figure 5: User-item matrix for Collaborative Filtering.....	14
Figure 6: Number of students in the different majors.....	28
Figure 7: Framework to support students' decisions throughout their study journey	30
Figure 8: Overall approach for prediction tasks.....	32
Figure 9: GPA frequency distribution.....	36
Figure 10: RMSEs of the algorithms used in predicting Major GPA.....	41
Figure 11: Predictors of performance in the Business major for each machine-learning algorithm	43
Figure 12: Predictors of performance in the IT major for each machine-learning algorithm	44
Figure 13: Predictors of performance in the Engineering major for each machine-learning algorithm	45
Figure 14: RMSEs of the algorithms used in predicting Concentration GPA	50
Figure 15: Actual vs. Predicted Concentration GPA- High GPAs.....	52
Figure 16: Actual vs Predicted Concentration GPA- Low GPAs.....	53
Figure 17: Predictors of performance in the Programming concentration for each machine-learning algorithm.....	55
Figure 18: Predictors of performance in the Networking concentration for each machine-learning algorithm.....	56

Figure 19: Predictors of performance in the Security concentration for each machine-learning algorithm	57
Figure 20: Average RMSEs using enrollment data vs Year 1 data	58
Figure 21: frequency distribution of students' grade points	60
Figure 22: Average RMSEs of the main stages in this research	65
Figure A 1: Parameters of Deep Learning operator in RapidMiner	77
Figure A 2: Parameters of Decision Tree operator in RapidMiner.....	77
Figure A 3: Parameters of Random Forest operator in RapidMiner	78
Figure A 4: Parameters of Gradient Boosted Trees operator in RapidMiner	79
Figure A 5: Parameters of User k-NN operator in RapidMiner	80

List of Tables

Table 1: Examples of MAE and RMSE calculation	18
Table 2: Enrollment data features and range of values.....	35
Table 3: Summary of algorithms' performance for the Major GPA prediction	40
Table 4: Year 1 data features and the range of values	47
Table 5: Grade letters and their corresponding grade points	48
Table 6: Summary of algorithms' performance for the Concentration GPA prediction.....	50
Table 7: summary of the main performance predictors in each concentration.....	54
Table 8: Course grade prediction features and the range of values.....	60
Table 9: Un-pivoted format of (student, course, grade) data.....	61
Table 10: Grading system.....	62
Table 11: Target role assignment for user-based k-NN in RapidMiner	64
Table 12: Summary of the performance for the Course grade prediction	65

1. Introduction

Students' success continues to be a key concern to individuals, higher education institutions, policymakers, and nations. Students who do not succeed in their study, lose time and effort in their failed pursuits, and they and their families can suffer financially and emotionally. Institutions lose the scarce resources they invested as well.

Last year, three of my students were dismissed from college after reaching year 4, because they could not improve their grades within the given timeframe. This semester, a large group of students moved from one major to another, after struggling to keep good grades at their first major. It is saddening to witness students suffering the consequences of non-optimal academic choices. Students are the future of our nations, and as educators, we hope to see our students successful, in every way, and we are entrusted with the responsibility of providing our students with advice and support to their academic choices, and this how this research idea started.

There is a gap in the literature, as there is no cohesive solution that informs students' academic decision throughout their study journey. The work in this dissertation addresses this gap by utilizing the advances in machine learning to predict students' performance throughout their study years, from the time they enroll in college, till they graduate; in order to help them choose majors, concentrations, and courses.

1.1. Problem statement

From the time students decide to continue their higher education, they are asked to make decisions concerning their education, many of which can be challenging. When students join college, they choose a major. The main offered majors at the college of study are Business, Engineering, and Information technology. After finishing their first year, students choose specific concentrations in their majors. For example, in Information Technology, students can choose Security, Programming, or Networking concentration. Throughout their study, students decide which courses to take next, which general studies courses to register for, and which upper-level electives to choose.

Wrong academic decisions have a great and direct impact on students' success and future. Choosing a major or a concentration in which they cannot perform well, can result in failure, and perhaps moving to a different major and losing time, or dropping out of college altogether. If a student is on academic probation, choosing which courses to take next becomes a critical decision. If a student continues to have low grades and fails to raise his/her CGPA within a year, the student will be dismissed from the college. In the higher education institution under study, 2,892 students are currently on academic probation, which comprises 22% of the total number of students (Figure 1), and they are not alone. In the USA, around 30% of year-one students do not return for their second year, and more than \$9 billion is spent on them (Aulck et al., 2016). Furthermore, the completion rates of 4-year degrees in the US are around 50% (Sweeney et al., 2016). These alarming figures

require every possible effort to support students and the higher education institutions in this critical struggle.

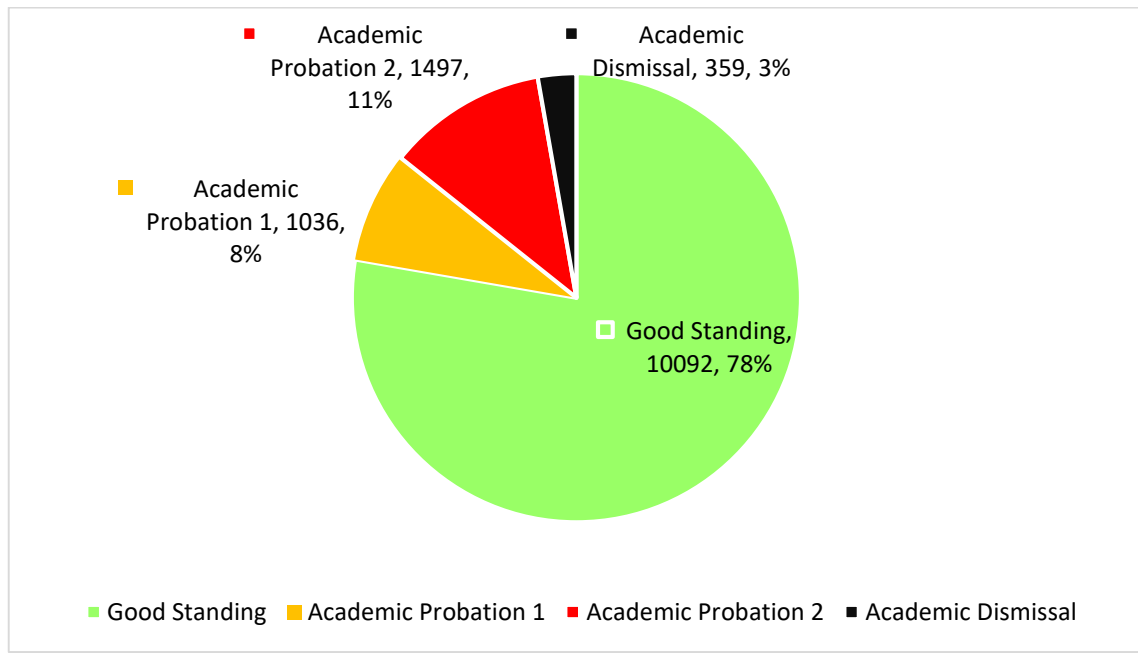


Figure 1: Academic standing of students

1.2. Research Objectives

To help alleviate those problems and to support students' academic decisions throughout their study journey, this work develops a framework that utilises historical data and machine learning algorithms to estimate how well a student will perform in different Majors, Concentrations, and Courses that they have not yet taken. Multiple machine-learning algorithms are tested and compared, to find the best performing amongst them. Furthermore, the main predictors of student performance at each stage are identified.

The performance prediction is important to students as it can be used by them and their academic advisors to make informed choices. This can also help identify the appropriate action to take and create personalised degree pathways that enable them to successfully and effectively acquire the necessary knowledge to complete their degrees in a timely fashion.

The prediction using machine learning algorithms is done at three stages:

1. At enrollment, this research uses enrollment data to predict student performance (measured in GPA) in each of the three main offered majors: Business, IT, and Engineering. This can reduce the percentage of students changing majors after finding out that the major they selected was too challenging for them.
2. After year 1, this research uses grades of the finished courses to predict student GPA in different concentrations, such as Networking, Security, or Programming. This can inform the selection of the concentration that best matches their capabilities and maximises their chances of success.
3. At any point of time after finishing some courses, this research uses the student's finished course grades in addition to other students' grades to predict future course grades. Providing prediction of student's grades in different courses can assist the student in choosing their courses.

These tools help not only students, but also academic advisors, teachers, and administrators while supporting students at different stages of their study journey.

Students on probation can avoid courses with low predicted grades and opt for courses

with the highest predicted grade. Alternatively, students, their teachers, and advisors can take preventative measures and actions if a student is predicted to perform poorly in a mandatory course. Stakeholders (mainly top management) can greatly benefit from such prediction. I met with some senior management, and they were very interested in how this research can help reduce the number of students on probation. Furthermore, they are planning to offer custom specialisations and interdisciplinary degrees to students to encourage entrepreneurship and innovation, and predicting performance would be of value for students while making their choices.

To achieve the above objectives, this research will be answering the following **research questions**:

RQ1: How effectively can student performance in a Major be predicted at enrollment?

- 1.1. What are the best performing machine-learning algorithms?
- 1.2. What are the main predictors of student's GPA in the different majors?

RQ2: How effectively can student performance in a Concentration be predicted after year one?

- 2.1 What are the best performing machine-learning algorithms?
- 2.2 What are the main predictors of a student's GPA in the different concentrations?

RQ3: How effectively can student performance in a course be predicted?

The remaining sections of the report are organised as follows: Section 2 has background information, and section 3 has the literature review. Section 4 covers the overall methodology and process , in which sections 4.1, 4.2, and 4.3 describe in detail the three main research areas: Predicting Major GPA, Predicting Concentration GPA, and Predicting Course Grade. In each of those sections, I describe the data; the required preprocessing, the configuration and performance of algorithms used for the prediction, the top predictors (when applicable), followed by discussions of the results at each stage. Finally, section 5 has a conclusion and future work.

2. Background

This section provides background information about machine learning in general, and regression algorithms used in this study in particular. The algorithms used are Decision Trees, Random Forests, Gradient Boosted Trees, Deep Learning, and K-Nearest Neighbor. It also describes two popular similarity measures used in those algorithms, Euclidean Distance, and Pearson Correlation Coefficient. Lastly, popular regression evaluation metrics are described, mainly Mean Absolute Error and Root Mean Squared Error.

2.1 Machine Learning

Large amounts of data are stored in databases; with potential importance; however, they are not fully discovered. “Data mining is the extraction of implicit, previously unknown, and potentially useful information from data” (Witten et al.,2016). They are algorithms that

go through databases automatically, looking for patterns, which can be generalised into structures to make predictions on future, unseen data. Machine learning is the technical basis of data mining. It takes the raw data (i.e. the examples) and infers the structure that lies behind it. The found structure can be used for prediction, explanation, and understanding the data.

Data mining prediction approaches describe what will happen in the future based on what happened in the past. Many techniques describe the structure that is used to classify the data. This structural description helps to understand the data in addition to predictions. This is one of machine learning's major advantages over classical statistical modelling. As stated by Lison (2015), "Virtually all learning problems can be formulated as (complex) mappings between inputs and outputs". It tries to find a function f that produces output o for a set of possible inputs i .

There are two classical machine-learning methods depending on the kind of available data:

- **Supervised machine learning:** supervised machine learning is predictive, and is used when we have examples of data that have both the inputs and the correct outputs (i,o) (Knox and Overdrive.com, 2018). Supervised machine learning methods include:
 - **Classification:** classification algorithms predict discrete class labels. They learn from the given data input - usually multiple attributes- and then use this learning to classify new observation. For example, based on student's records, it could predict the classification of the student to be "At risk" or "Not at risk". The algorithms can also produce a multi-values classification. Some classification

algorithms include Decision Trees (DT), K-Nearest Neighbors (KNN), Artificial Neural Networks (ANN), Naïve Bayes (NB), and Support Vector Machines (SVM).

- **Regression:** regression predicts a continuous number. It also learns from the data input given to it and then uses this learning to predict the output for new observations. For example, based on students' records, it could predict final student mark (range 0-100) or GPA (range 0-4). Some popular algorithms that can be used for regression are Decision Tree or an ensemble of it, such as Gradient Boosted Trees and Random Forests, Artificial Neural Networks (including deep learning), and Linear Regression.
- **Unsupervised machine learning:** unsupervised machine-learning is mainly descriptive, and is used when we only have the inputs *i*. Main approaches include:
 - **Clustering:** Clustering finds groups of similar objects. For example, it could identify segments of customers. This could be used for targeted marketing, recommender systems, etc. Algorithms used for clustering include K-Means, Hierarchical clustering, Density-based clustering
 - **Association mining:** Association mining discovers interesting relations between variables. For example, it could reveal that certain products are usually bought together. Apriori algorithm is a popular algorithm used for association mining.

- **Dimensionality reduction:** Dimensionality reduction aims at finding a low-dimensional representation of the data while retaining as much information as possible. It has interesting applications in compression and feature elicitation.

This research attempts to predict a continuous number, (GPA), and the GPA is known for the given sample (students GPAs), hence, regression is used. In the next section, I briefly explain the main algorithms used to predict the student's GPA or grade.

2.2 Regression Algorithms used

I decided to use regression algorithms instead of discretizing the GPA and using classification -even though classification is more commonly used in literature- because it provides more information than classification, as also stated by Strehle et al. (2015). The performance of different machine learning regression algorithms depends on the structure and size of the data. Hence, the choice of an algorithm often remains unclear until we test our algorithms on the data. This study uses and compares multiple regression algorithms, namely: Decision Trees, Random Forests, Gradient-Boosted Trees, Deep Learning, and K-Nearest Neighbor. The following sections discuss each one briefly.

2.2.1 Decision Trees (DT) for Regression

Decision Trees used in data mining are of 2 types, Classification Trees, and Regression Trees. Classification Trees predict a discrete value (class), while regression trees predict a continuous value

Decision tree algorithm builds a model that predicts the value of an attribute (label) based on several input variables. While training the tree, a labelled data is used, and the algorithm decides which attributes are best to split on. The algorithm splits the attributes which result in the purest child nodes. Regression trees try to minimise error at each leaf and use the variance to choose the best split. The split with minimum variance is used as the criteria to split the population using the variance where $variance = \sum$

Building the tree is usually inexpensive and predicting the values is relatively fast, and it can handle both numerical and categorical data. A major advantage of trees is that they produce a set of rules that can be easily understood by humans. However, Decision trees can grow to be complex and not generalise well from the training data, which is known as overfitting (Lior et al., 2014)

2.2.2 Random Forests

Random forests are simply an ensemble of decision trees. Multiple trees are generated instead of just one, and each tree gives a prediction or classification (Figure 2). In classification, the forest chooses the classification that had most of the trees votes. For regression, the forest takes the average of the predicted values of all the trees (Jones and Linder, 2015). Random forests are very good at learning complex relationships and can achieve high performance, and results are easy to interpret. However, they can be prone to overfitting as well, and they are resource and time intensive.

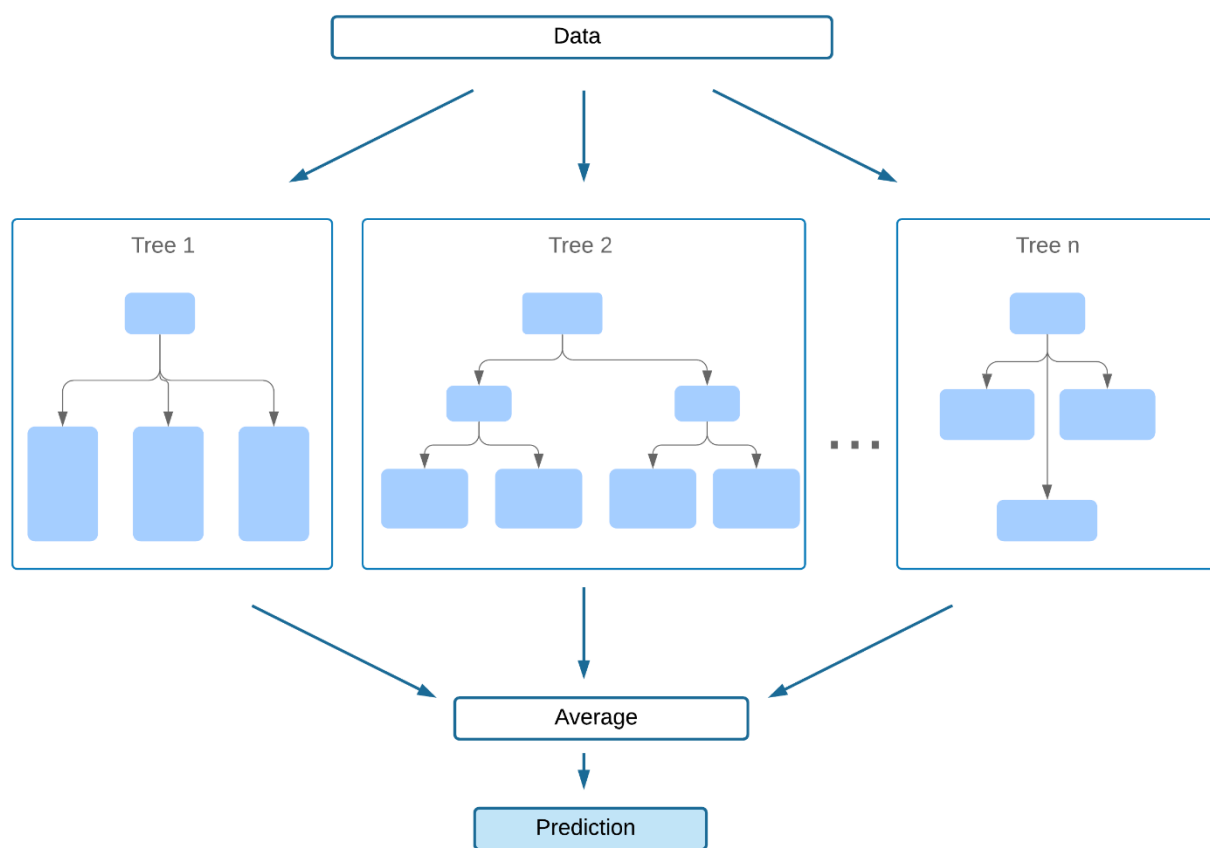


Figure 2: Random Forest

2.2.3 Gradient-Boosted Trees

Gradient Boosted Trees are also ensembles of decision trees, however, unlike the random forest where multiple trees are created “randomly”, the Gradient boosted algorithm generates the trees one after another (Figure 3). This is done by starting with a tree, then creating another tree that attempts to correct the errors from the first one, and so on. Single trees can have weak predicting accuracy, but when used together they can create a

stronger predictor, hence the term “boosting”. This approach has shown considerable success in a wide range of practical applications (Natekin and Knoll, 2013).

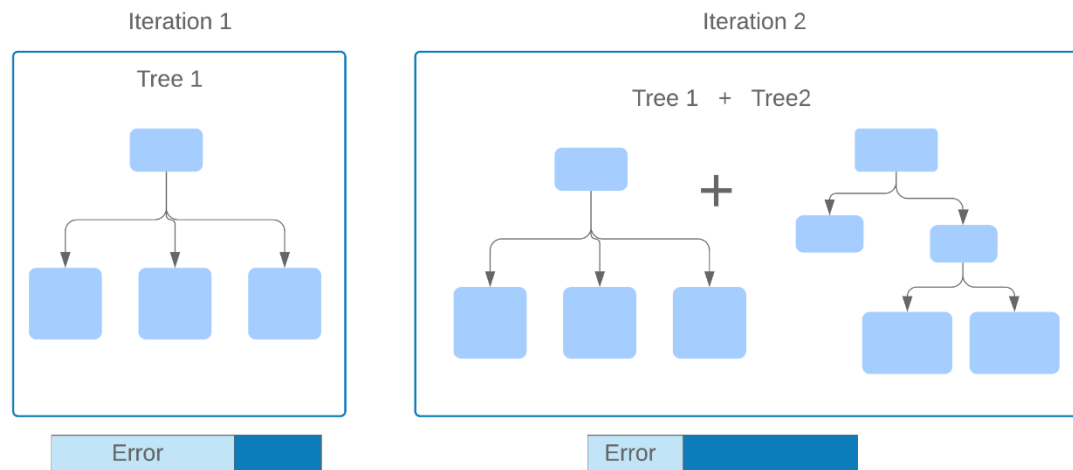


Figure 3: Gradient Boosted Trees

2.2.4 Deep Learning

Neural Network consists of interconnected nodes, called neurons, and weighted links inspired by information processing in the human brain (Figure 4). Neural Networks are trained using labelled data. The algorithm tries to find the correct function to transform the input into the output. It can have multiple layers where the output of one layer is passed to the next layer. Deep learning is a neural network with multiple layers between the input and output layers (hence the name “deep”). Naser et al. (2015)

One of the main advantages of neural networks is the ability to model complex non-linear relationships. When the algorithm is given a lot of data, performance usually increases.

However, the results are like a black box and not easy to interpret. It can also be computationally intensive to train and needs a lot of data to get good performance.

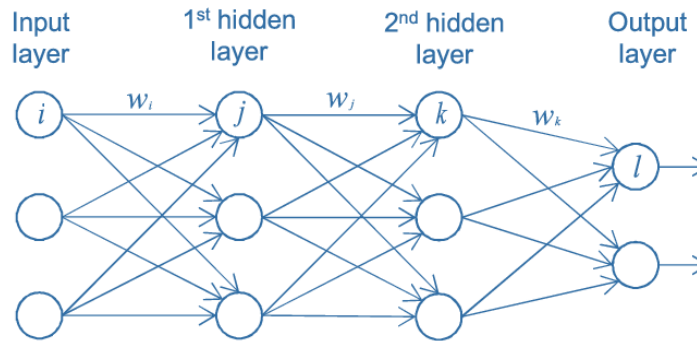


Figure 4: Deep Learning

2.2.5 Collaborative Filtering using K-Nearest Neighbor

Collaborative filtering (CF) is one of the most popular recommender system techniques (Iqbal et al. , 2017). It makes predictions of a user's interests based on the preference or interest of other similar users. It relies on large available data of interests or ratings of users, stored in a user-item matrix similar to the one in Figure 5 (Isinkaye et al. , 2015). The system finds similar users (user-to-user similarity) or similar items (item-to-item similarity) and predicts the rating of the new item. A rating is not necessarily explicit, it could be implicit such as clicking an item or liking it.

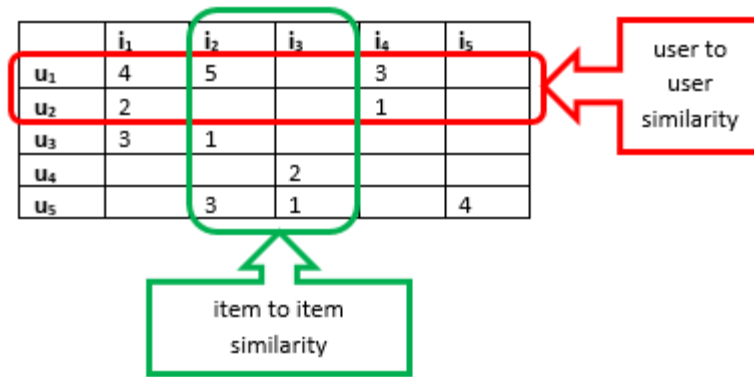


Figure 5: User-item matrix for Collaborative Filtering

In this study, the collaborative filtering algorithm is used to predict student's course grade. It takes the grade of a student instead of his/her rating, and predict a future grade in a course, by finding similar students in the data set, and aggregating the grades of those students to make a prediction.

A big advantage of collaborative filtering is domain-independence (Liu et al., 2014). There is no need to build an item profile in order to make recommendations. Hence, it can be applied to any domain.

However, there are several weaknesses in CF, as listed below Iqbal et al. (2017):

- Cold-start: Collaborative filtering needs sufficient explicit feedback/rating to make predictions. If a user has not rated anything, his/her preferences reference will be unknown. Moreover, a prediction cannot be done for new items that have not yet been rated.
- Data sparsity: the user-item matrix is usually sparse since not many users give rating or feedback to items. This leads to weak predictions and recommendations.

K-Nearest Neighbor is one of the most common algorithms to collaborative filtering (According to Amatriain et al., 2015). K-NN is simple and powerful. However, when it needs to be re-run every time we need to find neighbors, and is considered relatively expensive and slow to predict new instances. Nevertheless, this has an advantage as it can adapt to rapid changes in the data, such as the user rating matrix.

When a trying to predict a grade for a new student, student's grades in different courses are fed to the k-NN algorithm, and the algorithm finds the k nearest students (i.e. most similar or neighbours) in the training set. It assumes the student and his/her neighbours have the same grades and will predict the grade as a weighted average of those grades.

Similarity measures are usually used to calculate the similarities among users or items. The most popular distance measure in this algorithm is Euclidian distance and Pearson correlation Coefficient, both of which will be discussed in the coming section.

2.3 Similarity measures

Similarity measures are an important concept in data mining. They measure the distance between items, and they are used by many machine learning algorithms. The following section gives an overview of two main similarity measures, Euclidean Distance, and Pearson Correlation Coefficient.

2.3.1 Euclidean Distance (ED)

Euclidean distance is the distance between points. Thus the similarity: $similarity(a, b)$ between two items a and b , with a sample size n is measured as: $similarity(a, b) = 1 - distance(a, b)$, and the Euclidean distance between the two items is measured as:

$$distance(a, b) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}$$

2.3.2 Pearson Correlation Coefficient (PCC)

Pearson correlation coefficient measures the linear correlation between two vectors and is one of the most commonly used similarity measures. (Isinkaye et al., 2015). The values range between -1 and +1, where +1 is a total positive linear correlation, -1 is a total negative correlation, and 0 is no correlation. In a sample size n , the similarity between item a and item b , is measured as:

$$similarity(a, b) = r = \frac{\sum_{i=1}^n (a_i - \acute{a})(b_i - \acute{b})}{\sqrt{\sum_{i=1}^n (a_i - \acute{a})^2} \times \sqrt{\sum_{i=1}^n (b_i - \acute{b})^2}}$$

Where \acute{a} and \acute{b} are the means of vectors a and b

$$\acute{a} = \frac{\sum_{i=1}^n a_i}{n} ,$$

$$\acute{b} = \frac{\sum_{i=1}^n b_i}{n}$$

2.4 Evaluation Metrics: RMSE and MAE

When the output of the algorithm is a continuous numeric value, evaluations of algorithms accuracy in predictions commonly performed using two regression metrics: Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) (Chai and Draxler, 2014)

MAE is the average of the absolute difference between the actual value and the predicted one, and is calculated as:

$$MAE = \frac{1}{N} \sum_{i=1}^N |p_i - a_i|$$

Where p_i is the predicted value and a_i is the actual value of the same instance. Lower values of MAE and RMSE implies higher prediction accuracy.

Root Mean Squared Error (RMSE) measures the standard deviation of the prediction errors (also called residuals), and is calculated as $RMSE = \sqrt{\frac{1}{N} \sum (p_i - a_i)^2}$

Some of the predictions might be more than the original value (positive), and some are less (negative). Simply adding the differences will result in cancelling each other out, that is why the formula squares the difference. RMSE also penalizes the larger prediction errors more than the small ones (Chai and Draxler, 2014), and it is very important to avoid predicting student's scores to be far from the actual value, hence, this metric is used in this study to compare algorithms because it penalizes algorithms that produce larger prediction errors.

Table 1 shows an example of the difference between how MAE and RMSE are calculated. An advantage of RMSE is that the error uses the same units as the predicted value, which makes it easier to understand the results.

Actual Values	Predicted Values	MAE	RMSE
2,4,6,8	4,6,8,10 (no large errors)	2.0	2.0 (RMSE is similar to MAE)
2,4,6,8	4,6,8,12 (one large error)	2.5	2.56 (RMSE is higher than MAE)

Table 1: Examples of MAE and RMSE calculation

3. Literature Review

Data mining is the process of discovering useful patterns and trends in large data sets.

Educational Data Mining (EDM) is the application of data mining techniques in the field of education, to address important educational questions, and this area of research is growing rapidly (Shahiri and Husain, 2015). Del Rio and Insuasti (2016) surveyed papers that used data mining in predicting academic performance in traditional environments at higher-education institutions. Many of the papers reviewed addressed the issue of predicting academic performance to support student decisions. Majority of the papers reviewed predict course grades, while a few recommended majors or specializations. Also, several studies tried to predict future performance, while others attempted to predict dropouts. To the best of my knowledge, there are no papers that support student academic decisions throughout their journey in higher education, from choosing a major to a concentration, to a course.

The reviewed papers are grouped into four main categories:

1. Papers that predict a course grade,
2. Papers that predict performance in a major or a concentration
3. Papers that predict future performance (such as future GPA)
4. Papers that predict dropout.

The following sections review some of the papers in each category.

3.1 Predicting Course Grades

A large number of studies aimed at predicting course grades and many of them used those grades to recommend courses to students. Elbadrawy and Karypis (2016) in their study attempted to predict Grades of students in courses, and recommend top-n courses to students. The study used multiple sources of data such as course features, student features and academic level. The dataset was comprised of 1,700,000 grades spanning 13 years. The research used collaborative filtering and matrix factorization, in addition to popularity ranking methods. It reported that small sample sizes affect grade predictions accuracy negatively. It used a 0-4 GPA scale and the RMSEs they achieved varied, but the lowest RMSE was 0.65.

Another research that also predicted course grade was done by Iqbal et al. (2017) and used students' pre-university data such as high school grades, entry test scores, course credits, and course grades of 24 courses, for 225 undergraduate students. It also examined Collaborative filtering, Matrix factorization, in addition to Restricted Boltzmann Machines (RBM) techniques. The evaluation metrics used were Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). The research concluded that Restricted Boltzmann Machines performed better than other techniques for predicting the students' grades in a particular course. The number of students is relatively small, and the study centralized the data (i.e. subtracted the average GPA of the course from the predicted values), so the RMSE they reported (which is as low as 0.3) is relative.

Ng and Linn (2017) attempted at predicting student rating of the course. The study used data from multiple sources, including course information, professor information, and students preference. Course topics were extracted using machine learning algorithms from a corpus of course descriptions. The authors also performed sentiment analysis on professors' and courses' ratings from RateMyProfessor.com website in an aim to provide a general approach that could be applied in any higher education institution. The research also asked students for their preference of the course type (for example, the quality of the course, how easy it is, etc.). Matrix factorization was used to predict the student rating of the course and make recommendations accordingly. In similar research, Chang, Lin, and Chen (2016) recommended courses to students after predicting student grade and rating of course based on multiple sources of data such as students information (including grades) and professor ratings. They also investigated combining multiple methods including collaborative filtering, clustering, and Artificial immune network.

Polyzou and Karypis (2016) predicted future course grade based on previous courses taken by the student. Their dataset had 76,748 grades of 2,949 students. This study examined both Matrix Factorization and Linear Regression. It used a 0-4 scale for GPA, and their RMSEs ranged between 0.60 and 0.75. The research showed that the accuracy of grade prediction could be improved by focusing on course-specific data, but the degree of improvement depended on the department. Dwivedi and Roshni (2017) also examined collaborative filtering approach to predict students' future grades, but it was item-based, using historical students' grades. The research used Mahout Machine Learning and Hadoop for the recommendation.

Upendran et al. (2016) and Sorour et al. (2015) used different data mining approaches, mainly unsupervised machine learning. Upendran et al. (2016) examined the use of association rule to predict the performance of the student in courses and make recommendations accordingly. The rules with the highest support and confidence were used in the recommendation model. Data used included high school grades (Math, physics, chemistry, biology, English). Sorour et al. (2015) used clustering (namely k-means), in addition to text mining (namely latent semantic analysis (LSM)) to analyse and predict student performance in a course. The text it mined contained free-style comments by students at the end of lessons. The prediction accuracy reached up to 78.5%.

Yang et al. (2018) on the other hand combined Multiple Linear Regression (MLR) and Principal Component Analysis (PCA) to improve the accuracy of predictions. The dataset included data about student's online activity (such as video-viewing), homework, exercises, and quizzes, to predict their final grade in a course.

While the previously reviewed papers provide important work in the area of predicting student performance, they only support students at the course selection stage. Major selection and concentration selection are not addressed in any of the mentioned studies. This work looks at a more comprehensive approach that supports students throughout their journey, and not just at one stage.

3.2 Predicting performance in a Major/Concentration

Not many studies were found to predict performance in a major or specialisations. Bautista et al. (2016) used classification to recommend specialisation for engineering students finishing their general engineering courses. The study used multiple algorithms and found the decision tree to be the best classifier with an accuracy of 80.06%. The study found that students grades of Algebra, Calculus, and physics, in addition to student's gender are the main predictors of success in the engineering specialisations, such as Civil Engineering, Computer Engineering, Mechatronics Engineering, and Manufacturing Engineering, etc. on the other hand, Kusumaningrum et al. (2017) used association rule to recommended majors to students based on their academic history, profile data, and interests.

Mostafa et al. (2014) recommended majors to students transferring from one major to another, using a case-based reasoning system (CBR). The study based the recommendation on the similarity between the previous courses taken by the students and the concepts in the different majors and recommends the major that is nearest to student's learned concepts. Surveys were also given to advisors to evaluate the system, but no results were published.

These studies also focus only on recommending either a major or specialisation and do not support students' academic decisions throughout their study journey.

3.3 Predicting Future performance

Several studies investigated future performance prediction. Naser et al. (2015) used Artificial Neural Network (ANN) to predict senior student performance in the faculty of Engineering. The authors used numerous variables for input such as high school score, Math I, Math II, Electrical Circuit I, and Electronics I scores, number of completed credits, CGPA, high school type, and gender, among others. The data consisted of 150 students only, and they stated that their ANN model was able to correctly predict the performance of more than 80% of prospective students. However, the study only focused on one algorithm only and did not explore other possible algorithms.

Asif et al. (2017) used data mining to predict and understand the performance of students. Firstly, they used classification to predict student graduation performance using socio-economic data for 210 students. Data used included pre-university grades in addition to the first two years. Secondly, the research identified courses that predict good or poor performance. Using decision trees, four courses were found to be the strongest indicators. Lastly, they investigated how students' academic performance progresses over the years of study. The clustering techniques they used revealed that students tend to have the same performance (low, intermediate, or high marks) in all courses, and throughout the years. Tekin (2014) also attempted to predict a student's GPA at graduation. The study investigated the use of Naïve Bayes', Support Vector Machine (SVM) and Extreme Learning Machine (ELM) classifiers. In one scenario, the researcher used students' grades in their first two years to predict their GPA at graduation. In the second scenario, grades for the

first three years were used. Their data consisted of the courses taken by 127 students only. Their best-reported accuracy was achieved using SVM and reached up to 93.06% for the first scenario, and 97.98% for the second scenario. Such high accuracy needs further investigation as the model could be overfitting.

Goga, Kuyoro, and Gogan, (2015) addressed predicting student's first-year performance by designing a framework using machine learning. The framework uses background data to make predictions, and utilizes Decision Trees, Neural Networks and Association rules methods. Furthermore, data is fed into a recommender system to suggest the course of action. The study, however, did not provide a detailed evaluation of the work.

Patil, Ganesan, and Kanavalli (2017) developed Feed-Forward Neural Networks and Recurrent Neural Networks to predict students GPA based on the courses they have taken previously. The research RMSE as the evaluation metric to compare the two methods. In a similar study, Al-Barrak and Al-Razgan (2016) used students' grades in previous mandatory courses to predict their future GPA. The dataset comprised of 236 students records and used Decision Tree only for GPA class prediction (A+, A, B+,..., F). The study identified the most important courses for predicting performance in each semester as well.

Elbadrawy et al. (2016) aimed at predicting next-term GPA as well as student's performance on in-class assessments (for example, homework). The research used regression-based methods and Matrix Factorization techniques. It reported an RMSE of 0.7381 for next term GPA prediction (GPA is between 0 and 4). The study also concluded that both Personalized Linear Multi-regression and their advanced Matrix factorization

techniques could predict next-term grades with lower error rates than traditional methods. The data set included admissions records and grades in courses that were already taken by all the students; in addition to course information and instructors' information.

All the studies in this section addressed predicting student's future performance while they are in their current year, but do not predict their performance in particular courses, majors, or specializations.

3.4 Predicting Dropout

Various studies addressed the concern of student's dropout. Aulck et al. (2016) attempted at predicting students retention using student demographic data, pre-college entry information, and first-year transcript records of 32,500 students at the University of Washington. Regularised logistic provided them with the best predictions, and the study reported math, chemistry, psychology, and English courses to be the strongest predictors of attrition, in addition to birth-year and enrollment-year.

Both Manhaes et al. (2014), Sara et al. (2015) used the classification techniques to predict dropout. Manhaes et al. (2014) used student's grades in each semester to predict dropout. The study examined multiple classification algorithms including Naïve Bayes (NB), Support Vector Machine (SVM), and Multilayer Perceptron (MLP). Naïve Bayes achieved the best accuracy of 80%. On the other hand, Sara et al. (2015) found but Random Forest classifier to achieve the best accuracy of 93.5% and an area under the curve (AUC) of 0.965. The research used a large dataset consisting of 36,299 students.

Wolff et al. (2013) predicted students at risk of failing an online course by analysing their clicks. The research used multiple sources of data such as demographic data, assessments, and virtual learning environment. They had three modules: module A consisted of 4,397 students and 1,570,402 clicks, and Module B consisted of 1,292 students and 2,750,432 clicks. The researchers used classification to predict the final result of a student (Pass or fail). It found the level of activity of students and the number of clicks around the exam times to be predictors of student performance. Better accuracy was reported as a result of combining assessments, demographic, and clicks data.

The studies reviewed in the area of predicting dropout focused mainly on whether a student is at risk of dropping out or not, and highlighted the predictors of dropping out. While this serves as a very useful warning system, it does not offer insight into future course performance, which could greatly help students who are at risk of dropping out choose courses that could take them out of the risk zone.

The work found in the literature review which focused on one stage of student's academic journey or another inspired the work of this research. The aim of this work is to provide a cohesive and comprehensive framework to support students' choices throughout their academic journey and offer them the largest amount of possible support as they choose courses, majors, and concentration, in the hope of maximizing their chances of better performance. In the coming sections, I go throughout the overall methodology and framework, followed by a detailed description and findings of each stage of support.

4. Methodology and Results

The college under study has three main majors: Business, Information Technology (IT), and Engineering. There is a total of 13,750 students in the different majors as per Figure 6, where 4,047 students are studying Business, 3,492 in IT, and 6,211 in Engineering.

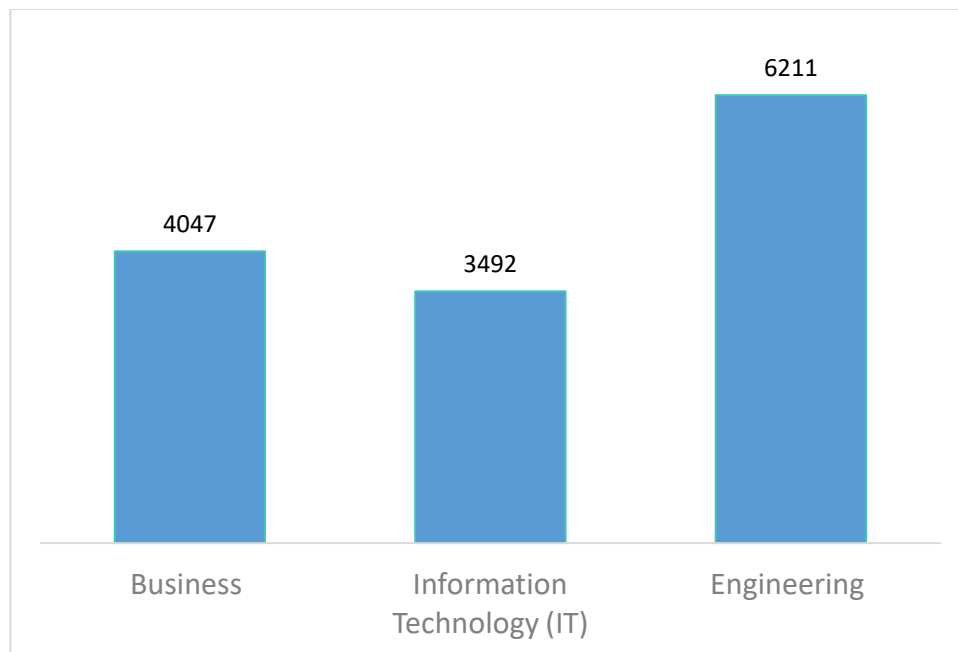


Figure 6: Number of students in the different majors

To supports students in choosing Majors, Concentrations, and Courses, this study develops a framework to predict their GPA at each stage. Figure 7 shows the suggested framework and a summary of the main tasks performed in this research. At each stage: data to be used, the prediction task and the algorithms used are shown.

The main sections of the framework are designed around the following stages of the student's journey:

1. At enrollment: based on their enrollment data, the framework **predicts their GPA in different majors** to help them choose a major most suitable for their capabilities.
2. At the end of Year 1: based on their grades in Year 1 courses, the framework **predicts their GPA in different concentrations** to help them in their choice.
3. At any time after year 1 or after finishing some courses: based on their grades in previously finished courses, the framework **predicts their grade in any future course** to help them choose courses.

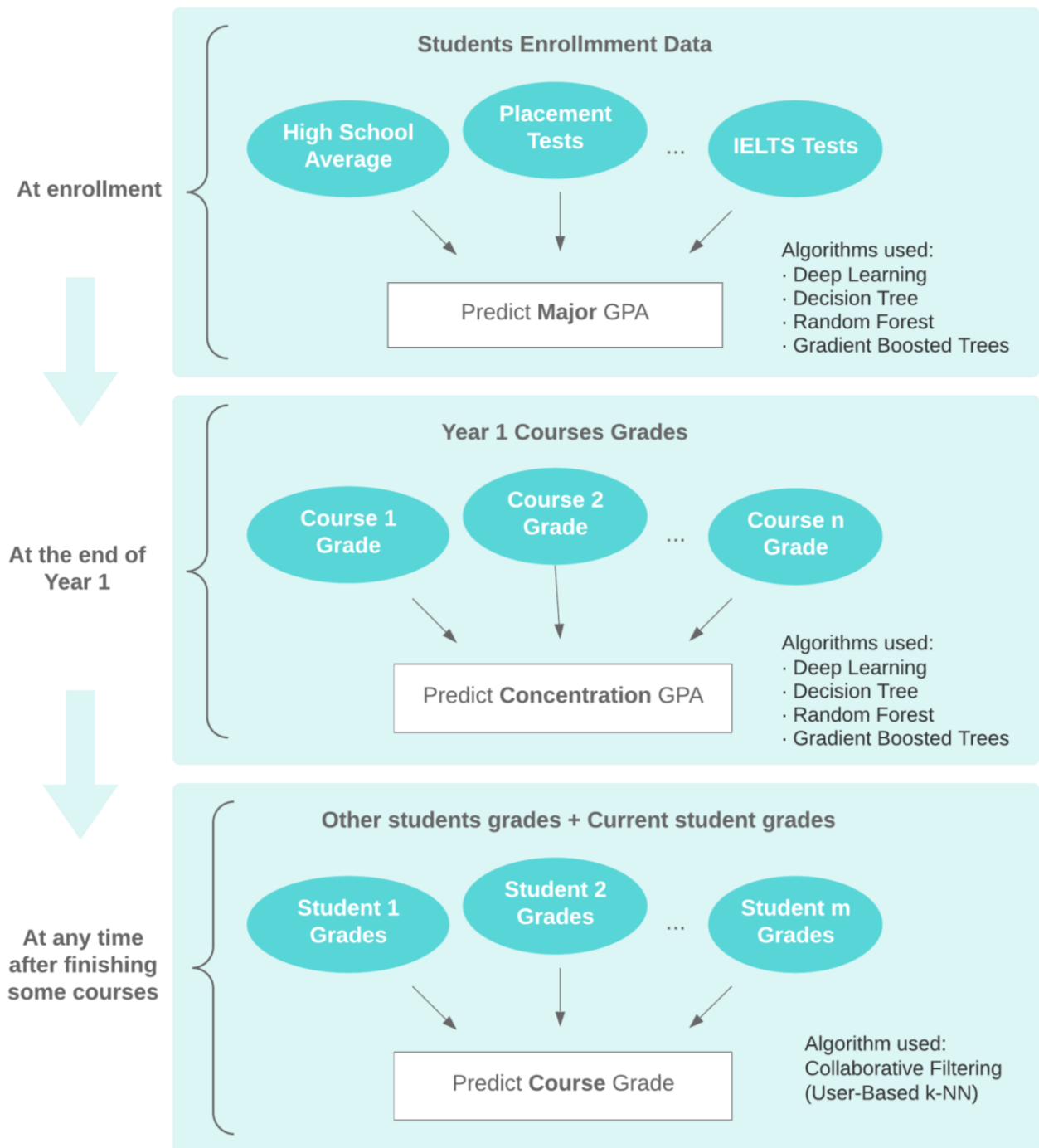


Figure 7: Framework to support students' decisions throughout their study journey

The Overall Process:

RapidMiner was used in this study to implement the framework and predict students' performance at the different stages. Below is the general approach used for all the prediction tasks (Major GPA, Concentration GPA, and Course Grade prediction) –also shown in Figure 8:

1. Preprocess data in Excel (basic preprocessing)
2. Retrieve data with a numerical label for regression.
3. Preprocess data as per the requirement of the algorithm and the task at hand
4. Assign “GPA” as the label for regression training and testing
5. Split Data into training data to build the model (70%), and testing data (30%) to test the performance of the model.
6. Pass the training data to a machine-learning regression algorithm to build a model.

The following algorithms were used and compared:

- Deep Learning
 - Decision Tree
 - Random Forest
 - Gradient Boosted Trees
 - User-based K-Nearest Neighbors
7. Apply the trained model to the testing data
 8. Find Performance of regression using cross-validation: Compare 'label' and 'prediction' to estimate performance.

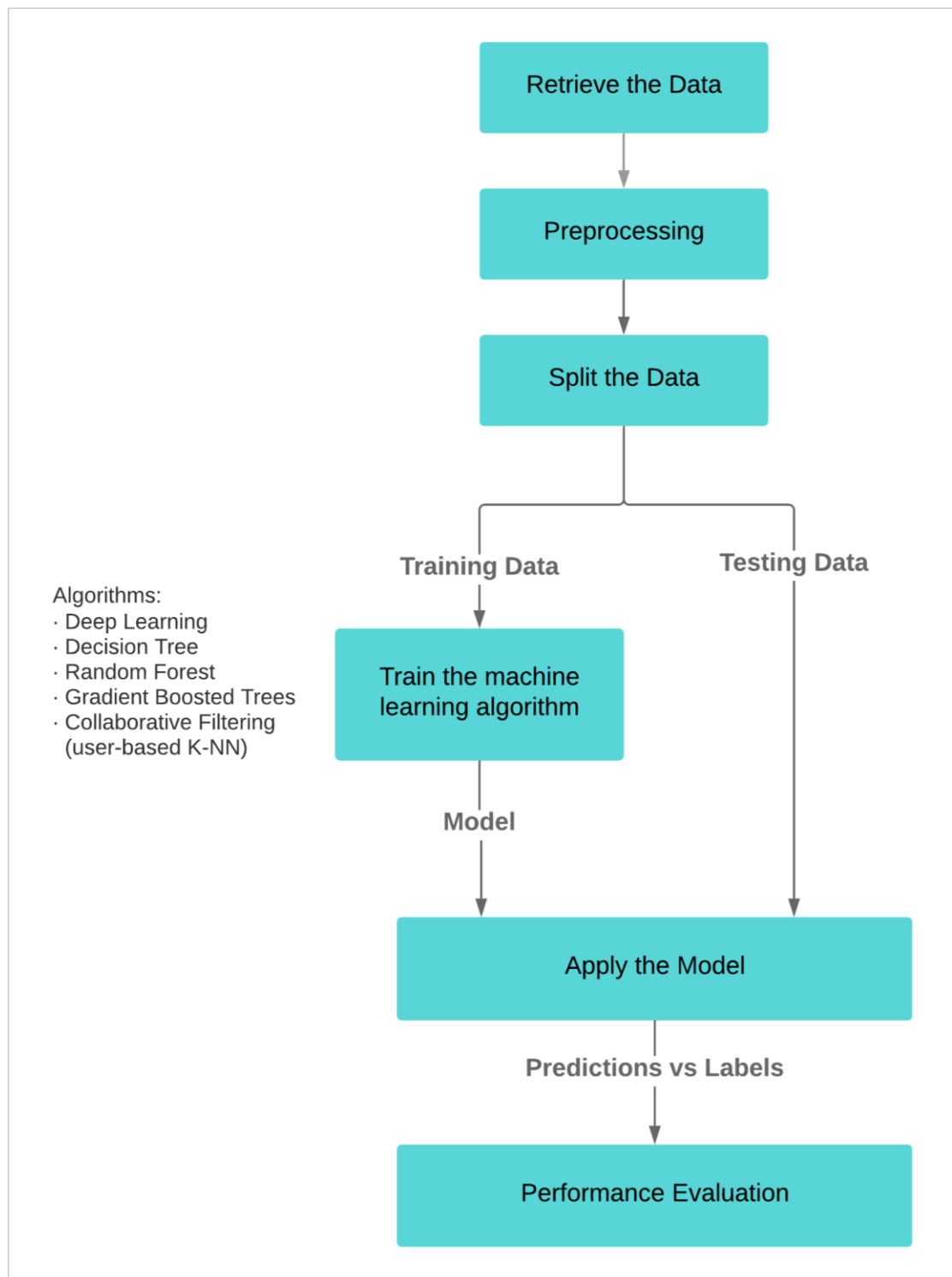


Figure 8: Overall approach for prediction tasks

In the following sections, I describe in detail the three main tasks of the proposed framework which provide answers to the main research questions. Section 4.1 “Predicting Major GPA” answers RQ1 “How effectively can student performance in a Major be predicted at enrollment?” It has the data used for predictions, the preprocessing tasks, the algorithms used, the performance of algorithms, and the strongest predictors. Section 4.2 “Predicting Concentration GPA” answers RQ2 “How effectively can student performance in a Concentration be predicted after year one?” It also describes the data, the preprocessing, the algorithms, the results, and the predictors. Finally, Section 4.3 “Predicting Course Grade” answers RQ3 “How effectively can student performance in a course be predicted?”

4.1 Predicting Major GPA (at enrollment)

When students first join the college, they need to choose a major. The only information available is their enrollment data. This data is used to predict student GPA in each of the three main offered majors: Business, IT, and Engineering to help him/her make an informed decision while choosing one of the majors. When a student joins college, we could run his/her enrollment data through each prediction model, and it will give the predicted GPA in each major. For example, if a student’s predicted GPA in one major turned to around 2.0 while in another major it was around 3.0, he/she might opt for the major with higher probability of success, and avoid the possibility of wasting time in a major that might not be best suited for his or her capabilities and strengths.

In order to do that, students’ historical enrollment data and their achieved GPA (after finishing two years) is used to train the algorithms for GPA prediction. In the coming

section, I describe the data, the preprocessing steps, the algorithms trained for predictions, the performance of the different algorithms, a discussion of the results as well as the strongest predictors of student performance.

4.1.1 Data

I collected enrollment data of 7,230 students studying in the year 2018 in 3 different majors: Information Technology (1,725 students), Business (2,412 students) and Engineering (3,093 students). Table 2 lists the used features obtained, and Figure 9 shows the frequency distribution of student' GPAs.

Feature	Values
High school Average	0-100
High school English	0-100
High school Math	0-100
High school Arabic	0-100
IELTS Band	0-9
IELTS Reading	0-9
IELTS Writing	0-9
IELTS Listening	0-9

IELTS Speaking	0-9
College placement tests (CEPA) English	0-210
College placement tests (CEPM) Math	0-210
College placement tests (CEPW) Writing	0-210
Major	Polynomial
Concentration	Polynomial
Employment	Yes/No
Gender	M/F
GPA (label):	Continuous value between 0 – 4

Table 2: Enrollment data features and range of values

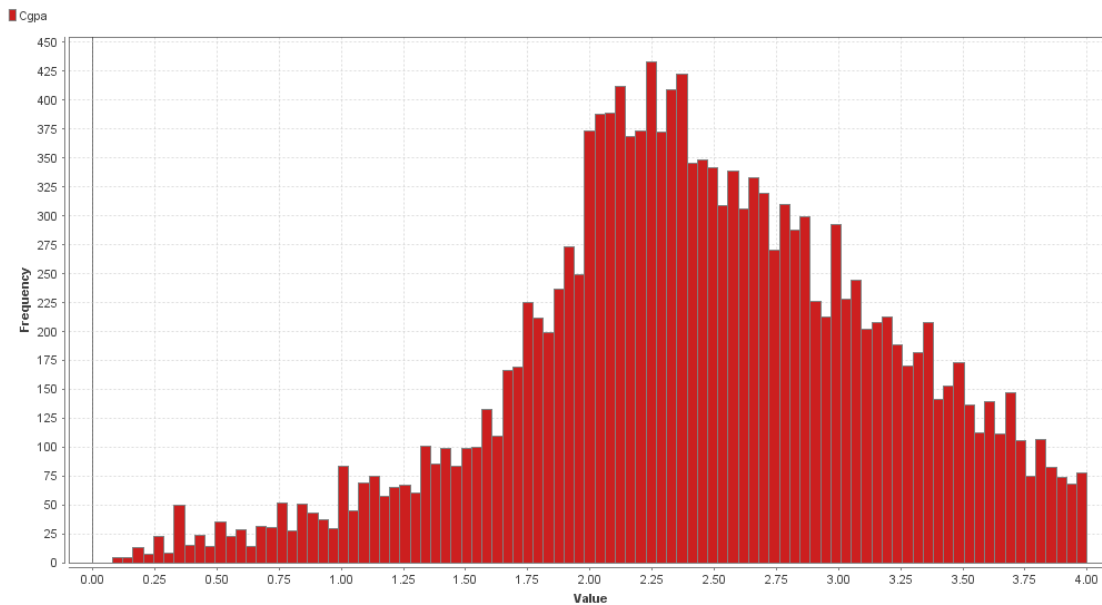


Figure 9: GPA frequency distribution

4.1.2 Preprocessing

Data collected had to be cleaned and made ready for prediction. Below are the main tasks of preprocessing:

- **Anonymized the dataset** –removed over 20 features that contained personal details of students such as IDs, names, and contact details.
- **Removed noise** –removed records that had mistakes such as letters instead of numbers , or wrong data such as 0218 for a year, instead of 2018.
- Removed students with GPA=0, as this is usually to students not showing up and getting **a failing grade in** all courses due to attendance.
- Removed all newly registered student records, by filtering their catalogue term, since they would not have any GPA,

- Generated a new feature for “Employment” to indicate whether a student is working or not. Original data only had the company name. This feature was generated to find out if employment is a contributing factor to performance prediction.
- Filtered students in Year 3 and Year 4 only. To achieve this, the number of credits completed was checked. Students who have completed more than 60 credits were assumed to have finished year 2.

4.1.3 Algorithms

The following algorithms are popular in literature for regression, and hence are used in this study:

- Deep Learning
- Decision Tree
- Random Forest
- Gradient Boosted Trees

RapidMiner auto model default values are used (unless otherwise stated).

Below is the description and the configurations of the algorithms used to perform the regression task in Rapid miner:

Deep Learning:

- RapidMiner H2O Deep Learning operator is used to predict GPA. Since the label is real, regression is performed.

- The hidden layer sizes parameter is set to 2 layers, each with 50 neurons. (Please refer to Appendix A, Figure A 1 for a complete list of the parameters).

Decision Tree

- RapidMiner H2O Decision Tree operator is used to predict GPA.
- “GPA” is set as a label.
- To use the Decision Tree for regression, 'least_square' is selected as a criterion. (Please refer to Appendix A, Figure A 2 for a complete list of the parameters).

Random Forest

- RapidMiner, Random Forest operator, is used for regression. The model port provides the ensemble of random trees used in combination to obtain a combined prediction. At each leaf of a tree, the average value of GPAs is shown.
- “GPA” is set as a label.
- To use Random Forest for regression, 'least_square' is selected as a criterion. (Please refer to Appendix A, Figure A 3 for a complete list of the parameters).

Gradient Boosted Trees

- RapidMiner H2O Gradient Boosted Trees operator is used to predict the GPA. Since the label is real, regression is performed.
- The operator's distribution parameter is changed to "gamma".

- The algorithm was used to generate 60 Trees to create an ensemble model. (Please refer to Appendix A, Figure A 4 for a complete list of the parameters).

4.1.4 Results

Table 3 shows a summary of the RMSE, standard deviation (STDV), and runtime (in milliseconds), for each algorithm, on each major (Business, Engineering and Information Technology). It also shows the number of records used in each major. Figure 10 shows a graphical comparison of the RMSEs.

	Business			Engineering			Information Technology			
Records	2,412			3,093			1,725			
Algorithm	RMSE	STDV	Run time	RMSE	STDV	Run time	RMSE	STDV	Run time	Avg RMSE
Gradient Boosted Trees	0.469	0.012	33805	0.45	0.008	16662	0.465	0.025	11867	0.461
Random Forest	0.486	0.017	87814	0.462	0.007	84087	0.458	0.033	31177	0.469
Deep Learning	0.484	0.016	16375	0.454	0.004	7629	0.481	0.019	3662	0.473
Decision Tree	0.53	0.015	10911	0.512	0.007	2971	0.515	0.026	1503	0.519

Table 3: Summary of algorithms' performance for the Major GPA prediction

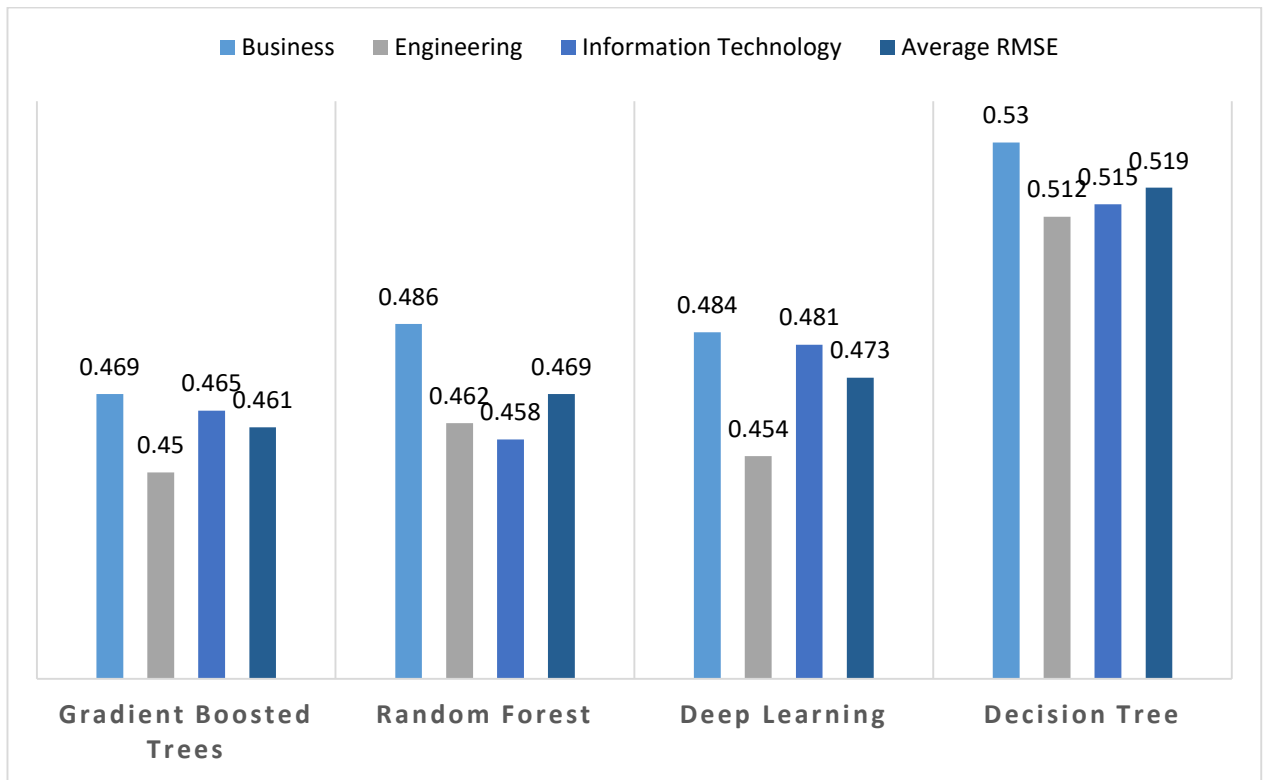


Figure 10: RMSEs of the algorithms used in predicting Major GPA

Elbadrawy et al. (2016) predicted next-term GPA using regression-based methods and Matrix Factorization techniques. Their RMSE was 0.7381 (GPA scale is between 0 and 4). In our research, Gradient Boosted trees algorithm performed the best in predicting Business and Engineering majors GPA (RMSE 0.469 and 0.45 respectively) while Random forest performed slightly better in predicting Information Technology GPA (RMSE 0.458). Deep closely followed with an average RMSE of 0.473. Decision Tree was the least performing across all data sets with an RMSE average of 0.519. It is interesting to find out that ensemble methods improve the accuracy of predictions. Standard deviations were low in

general (the highest was 0.03). Hence, standard deviations are not taken into consideration while comparing the performance.

4.1.5 Predictors

The strongest predictors of GPA in the different Majors: Business, IT, and Engineering are shown in Figure 11, Figure 12, and Figure 13 respectively. For the two majors of Business and IT, it was interesting to find out that most of the algorithms identified high-school average as the strongest predictor of students' GPA, while CEPA college placement tests and IELTS tests had less prediction power. However, for the Engineering Major, college placement CEPA Math test was the strongest predictor of student's GPA, which emphasises the importance of mathematical skills in this major.

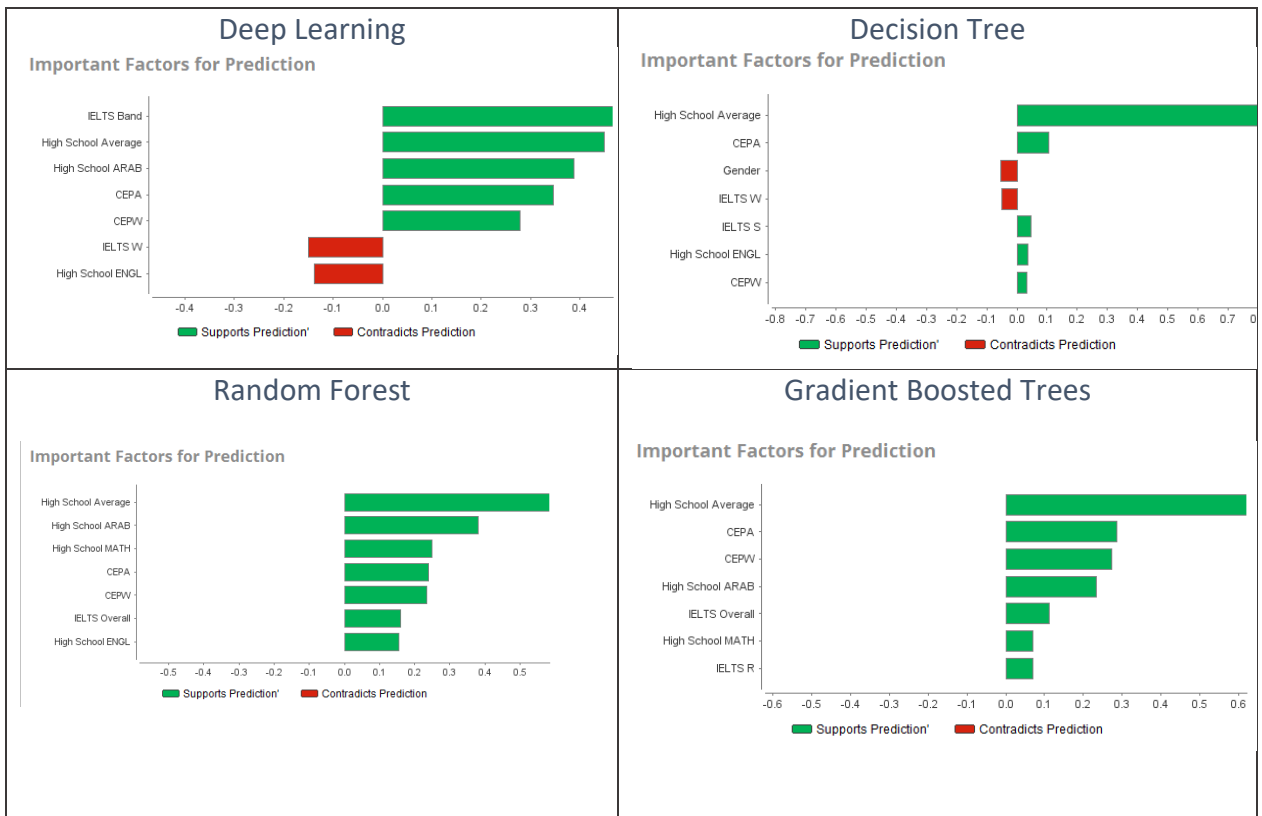


Figure 11: Predictors of performance in the Business major for each machine-learning algorithm



Figure 12: Predictors of performance in the IT major for each machine-learning algorithm



Figure 13: Predictors of performance in the Engineering major for each machine-learning algorithm

4.2 Predicting Concentration GPA (after year 1)

After joining a major, and at the end of their first year, students are asked to choose a concentration. For example, the Information Technology major has multiple concentrations, namely: Security, Programming, and Networking. Many students have difficulty choosing between the concentrations and are not sure which of them better matches their strengths and offers them the best chances of success.

To assist students in choosing a concentration by the end of year 1, this work predicts their GPA in the different concentrations, using their marks in five IT-related courses that they take in their first year. The five courses are:

- CIS 1003 - Introduction to Information Systems
- CIS 1103 -Introduction to Networking
- CIS 1203 - Introduction to Web Technologies
- CIS 1403 - Introduction to Programming
- CIS 1303 - Introduction to Database concepts

I built a prediction model for the three concentrations of the IT major. However, the approach applies to any major (which is planned for future research). When a student finishes Year 1 and wants to predict his/her GPA in different concentrations, we can run his/her five courses data through each prediction model, and it will give the predicted GPA in each concentration. This can help students decide on concentrations best suited for their capabilities.

In the coming section, I describe the data, the preprocessing steps, the algorithms trained for predictions, the performance of the different algorithms, a discussion of the results as well as the strongest predictors of student performance.

4.2.1 Data

The data collected consists of the student's grades in the five courses taken in Year 1, along with their GPA. The total number of grades is 7,740 grades of 1,560 senior students (in Year 3 and Year 4). The number of records in each concentration was as follows: Security (1,715 grades of 343 students), Programming (1,260 grades of 252 students) and Networking (1,160 grades of 232 students).

The features are shown in Table 4

Feature	Values
CIS 1003 Grade -Introduction to Information Systems (IS) (0-4)	0-4
CIS 1103 Grade -Introduction to Networking (NW)	0-4
CIS 1203 Grade - Introduction to Web Technologies (WEB)	0-4
CIS 1403 Grade - Introduction to Programming (PRG)	0-4
CIS 1303 Grade - Introduction to Database concepts (DB).	0-4
GPA (Label)	0-4

Table 4: Year 1 data features and the range of values

4.2.2 Preprocessing

To filter students in year 3 and year 4 only, the number of credits completed is checked.

Students who have completed more than 60 credits are assumed to have finished Year 2.

Data is anonymized, and cleared of errors. Certain columns were combined, as surprisingly enough, the same course grade was stored in different columns for different students because the same course had multiple codes (with a different suffix).

The individual grades obtained were in Letter format (A, A-, B+,..., F), so a new feature was generated to compute Grade Points (GP) in numbers (between 0 and 4), following the college grading system as shown in Table 5.

Grade Letter	Grade Points
A	4
A-	3.7
B+	3.3
B	3
B-	2.7
C+	2.3
C	2
C-	1.7
D+	1.3
D	1
F	0

Table 5: Grade letters and their corresponding grade points

4.2.3 Algorithms

To predict Concentration GPA, the same general approach is applied- as outlined in methodology (section 4) and used the same algorithms used for predicting Major GPA (section 4.1.3), namely, Deep Learning, Decision Tree, Random Forest, Gradient Boosted Trees. In this stage, the algorithms were trained on Year 1 course grades data (as opposed to enrollment data in the previous stage).

4.2.4 Results

Table 7 shows a summary of the RMSEs, and standard deviations (STDV) for each algorithm, and in each concentration (Security, Programming, and Networking). It also shows the number of records used in each concentration. Figure 14 shows a graphical comparison of the RMSEs.

	Security		Programing		Networking		
# of Grades	1,715		1,260		1,160		
# of students	343		252		232		
Algorithm:	RMSE	STDV	RMSE	STDV	RMSE	STDV	Average RMSE
Deep Learning	0.18	0.05	0.24	0.02	0.22	0.03	0.22
Random Forest	0.22	0.03	0.27	0.01	0.26	0.02	0.25
Gradient Boosted Trees	0.22	0.02	0.27	0.03	0.27	0.03	0.25
Decision Tree	0.33	0.02	0.31	0.02	0.35	0.05	0.33

Table 6: Summary of algorithms' performance for the Concentration GPA prediction

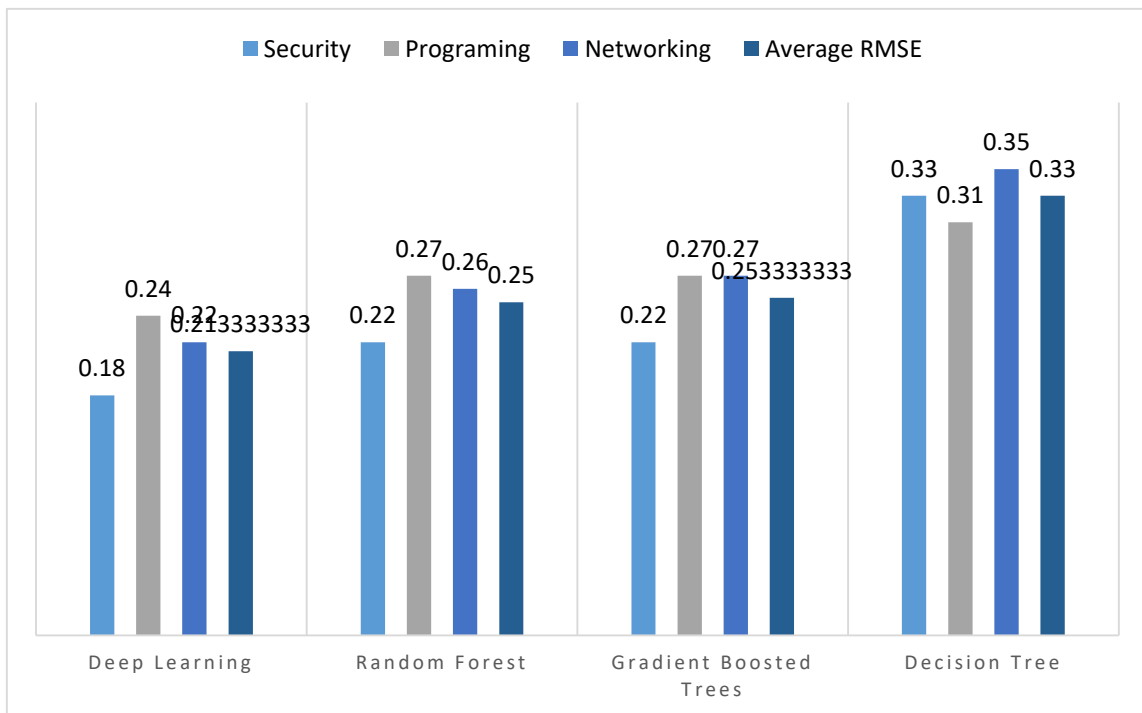


Figure 14: RMSEs of the algorithms used in predicting Concentration GPA

The results are exciting, as the RMSEs for predicting concentration's GPA are relatively low. The lowest average RMSE obtained was 0.21 using Deep Learning, followed by Random Forest and Gradient Boosted Trees (both had an average of 0.25 RMSE). Deep learning performed particularly well for the security concentration, with an RMSE of 0.18, perhaps due to the relatively larger number of records in the security dataset. Decision trees had the highest RMSE with an average of 0.33. It is worth noting that Decision Trees also had the least performance in the previous task of Major GPA prediction (section 4.1).

Overall, the standard deviation is also small (an average of 0.03 across concentrations) which means that this small error in prediction (the RMSE) is relatively consistent. Hence, it was not taken into consideration while comparing the performances of the algorithms.

I manually inspected the prediction results, both for high and low GPAs, to see how close the prediction are to the actual values, and they were very close (Please see Figure 15 and Figure 16).

Row No.	Prior Term ... ↓	prediction(Prior Term GPA)
48	3.950	3.692
70	3.950	3.692
54	3.900	3.623
41	3.870	3.514
80	3.840	3.692
67	3.810	3.609
64	3.800	3.623
68	3.800	3.662
69	3.800	3.662
86	3.790	3.692
21	3.780	3.495
10	3.770	3.547
22	3.740	3.623
23	3.730	3.492
6	3.720	3.605
...

Figure 15: Actual vs. Predicted Concentration GPA- High GPAs

Row No.	Prior Term ... ↑	prediction(Prior Term GPA)
32	2	2.381
60	2.070	2.243
12	2.080	2.054
65	2.090	2.331
7	2.110	2.392
5	2.130	2.768
43	2.150	2.039
50	2.150	2.239
9	2.180	2.387
59	2.200	2.205
40	2.240	2.124
58	2.240	2.403
14	2.260	2.716
72	2.310	2.783
20	2.320	2.083
64	2.350	2.467

Figure 16: Actual vs Predicted Concentration GPA- Low GPAs

4.2.5 Predictors

Another interesting finding was the predictors for concentration's GPA. The strongest predictors of GPA in the three Concentrations: Programming, Networking, and Security are shown in Figure 17, Figure 18, and Figure 19 respectively. It was surprising that for the Programming concentration, the programming course (CIS 1403) was one of the least predictors, while the database course (CIS 1303) was the strongest predictor. For the Networking concentration, all the algorithms also identified the database course (CIS 1303) as the strongest predictor, and the Deep learning algorithm also picked the networking

course (CIS 1103) as well. I have met with some of the teachers of the courses to discuss the reasons for such findings, and it looks like the database course is well structured and measures students' ability to think logically, which is required in all IT concentrations. According to them, the programming course could benefit from improvements in structure and content. These results require further investigation and could be of great value to discuss and put future strategies for the offered courses

For the security concentration, the Deep Learning algorithm identified the Information Systems course (CIS 1003) to be the strongest predictor, while the rest of the algorithms identified the Web Technologies course (CIS 1203) as a stronger predictor. Table 7 has a summary of the main performance predictors in each concentration.

Concentration	Main Performance Predictor(s)
Security	Intro. to Databases
Programming	Intro. to Databases
Networking	Intro. to Databases & Intro. To Networking
All IT concentrations	Intro. to Information Systems & Intro. to Databases

Table 7: summary of the main performance predictors in each concentration

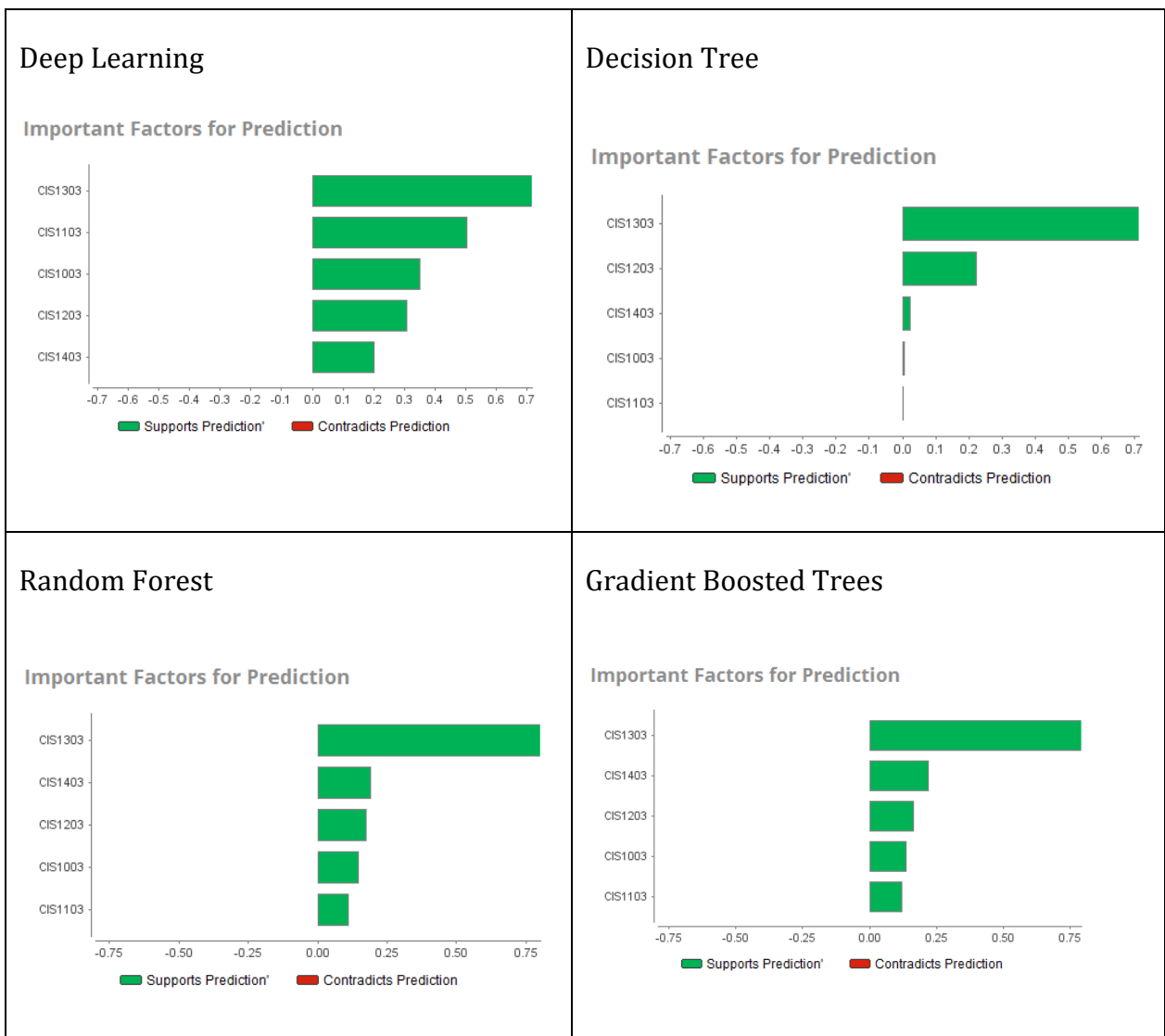


Figure 17: Predictors of performance in the Programming concentration for each machine-learning algorithm

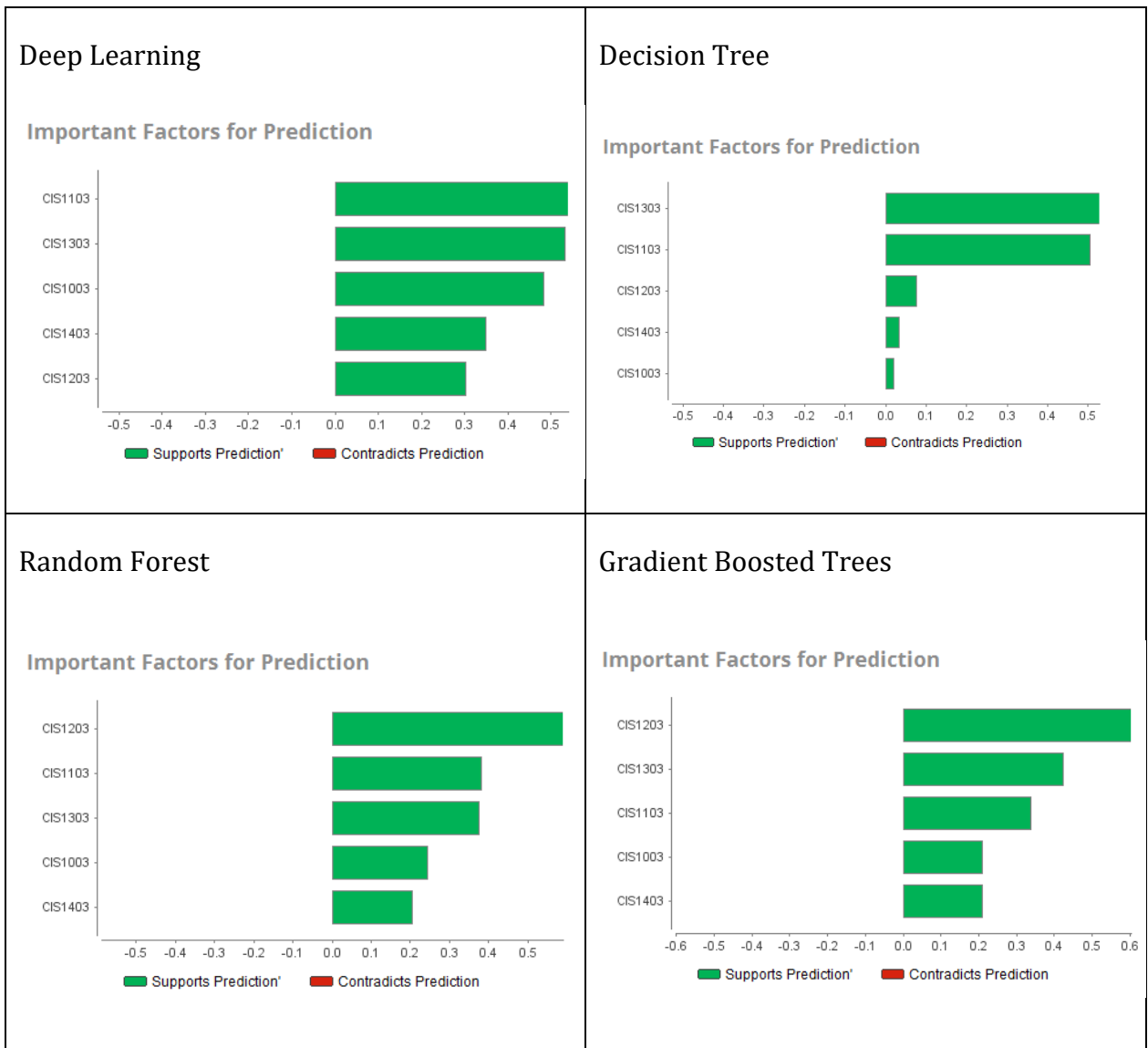


Figure 18: Predictors of performance in the Networking concentration for each machine-learning algorithm

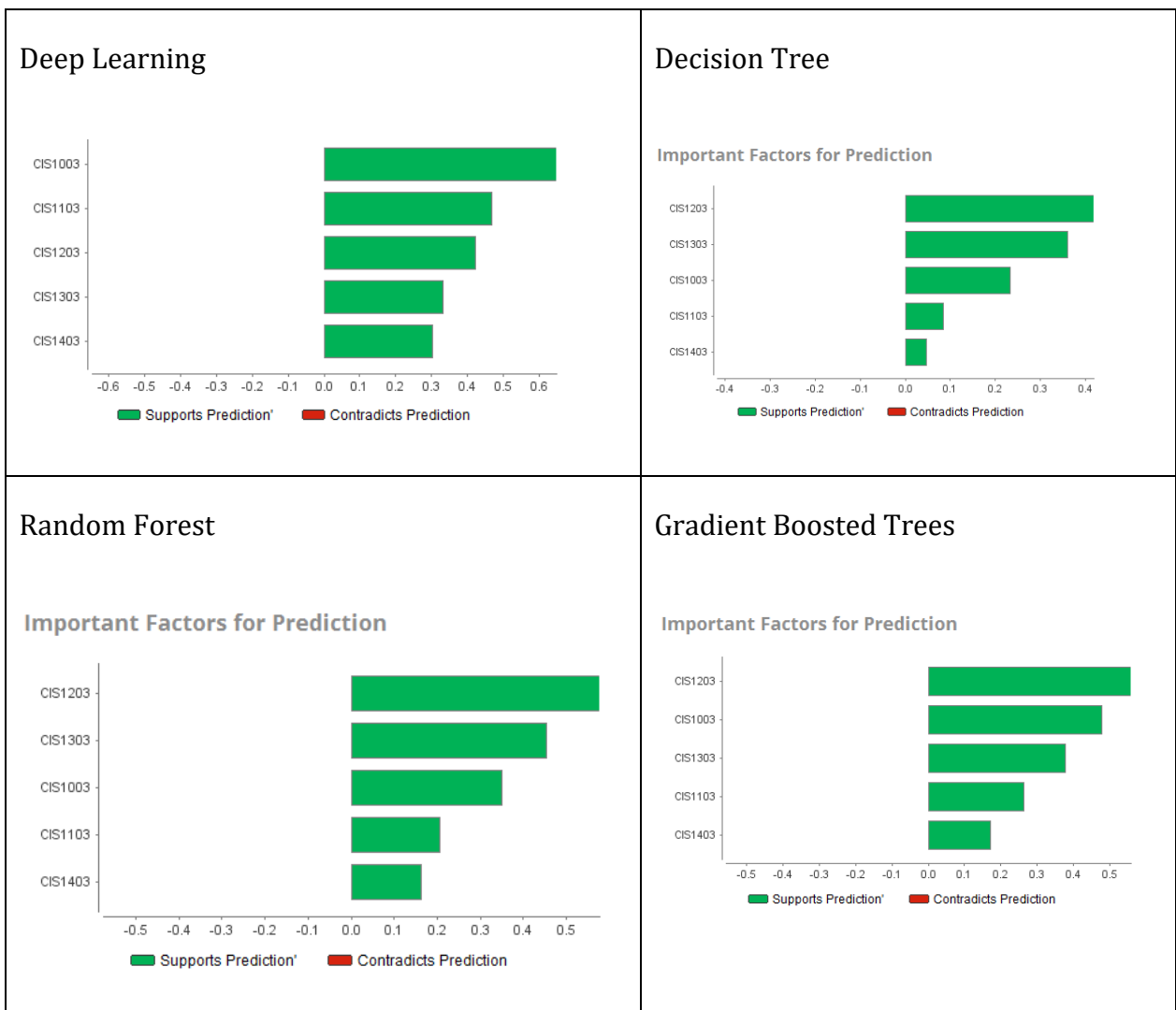


Figure 19: Predictors of performance in the Security concentration for each machine-learning algorithm

A significant observation is that the error in predicting concentration GPA after one year of study (using Year 1 courses) is considerably smaller than the error in predicting major GPA based on enrollment data (RMSE 0.22, and 0.46 respectively) as shown in Figure 20. This means that predicting performance becomes more accurate as students take courses in the college and that enrollment data is not as accurate in predicting GPA.

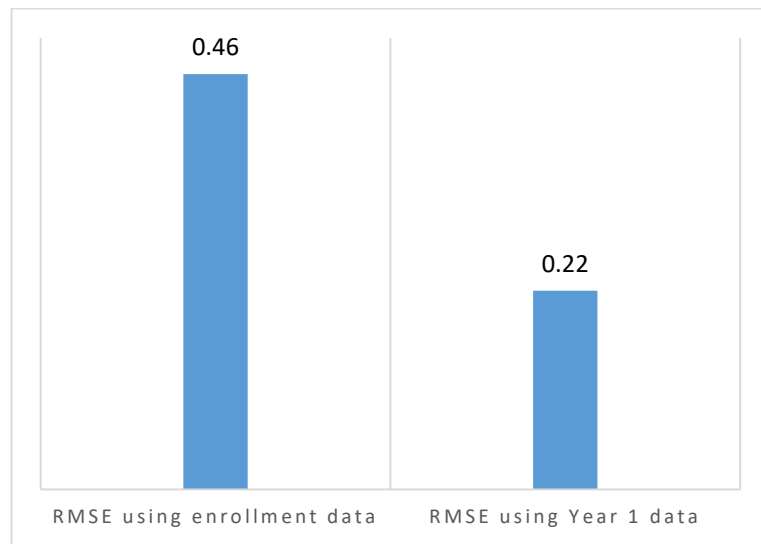


Figure 20: Average RMSEs using enrollment data vs Year 1 data

4.3 Predicting Course Grade

After finishing their first year, and throughout their study, many students, and in particular the ones at-risk, struggle to choose the next courses, especially with electives, and general studies courses. Students who are on probation are at risk of academic dismissal due to a low GPA. Choosing a course with the highest probability of success offers them a better chance to move out of probation. Furthermore, the college is promoting entrepreneurship and is moving to flexible degrees where students can customise their study plans and take interdisciplinary certificates. Hence, predicting student's performance in a course can greatly support students' decision in choosing a course for the above reasons. Future course Grade Point (0-4) in this task is predicted using a collaborative filtering approach. In this approach, the grades of any courses finished by the student, in addition the grades of other similar students, are used to predict future grade.

In the coming section, I describe the data, the preprocessing steps, the algorithms trained for predictions, the performance of the different algorithms and a discussion of the results.

4.3.1 Data

The obtained data consists of 227,507 grades in all offered courses across all the majors. There are 80,324 grades for the Business students, 60,440 grades for IT students, and 86,743 grades for Engineering students for the year of 2018. Figure 21 shows the frequency distribution of students' grade points and Table 8 shows the used features.

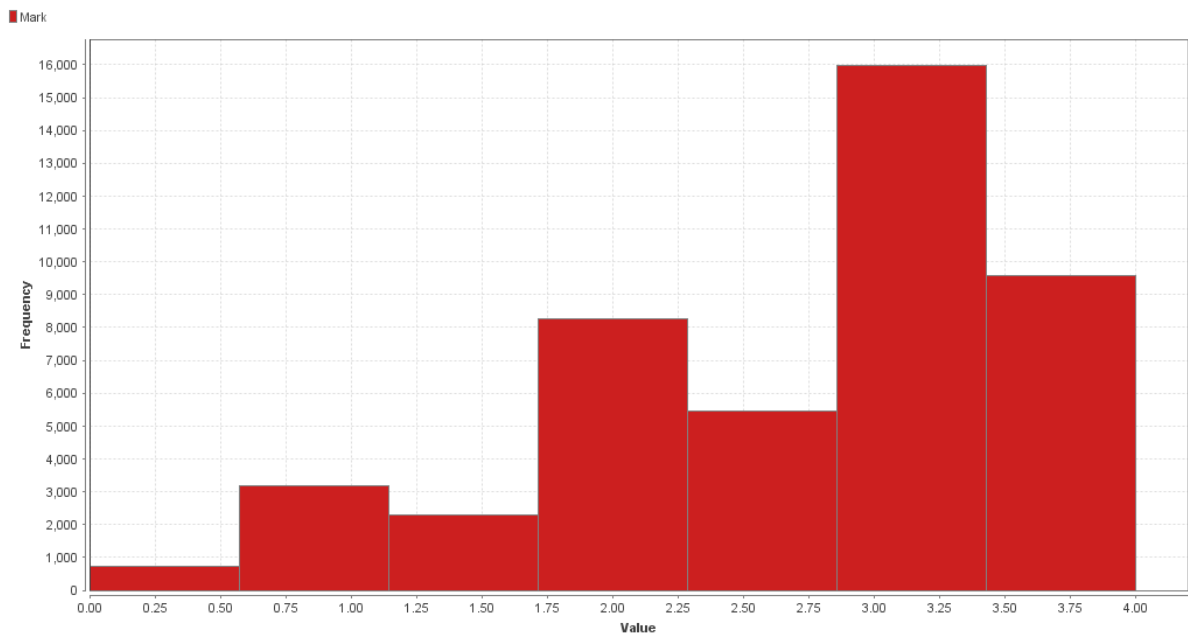


Figure 21: frequency distribution of students' grade points

Feature	values
Student ID	Polynomial
Course Code	Polynomial
Major	Polynomial
Grade in course	0-4

Table 8: Course grade prediction features and the range of values

4.3.2 Preprocessing

The following preprocessing steps are performed on the data:

- Integrate data from multiple files

- Un-pivot the data to follow the format (user, item, rating) that is necessary for prediction. The matching format for this research would be (student, course, grade) as shown in Table 9.

Student_ID	Course_Code	Grade Point
1	TEC-112	4
1	GEN-453	3.3
1	TEC-001	2.3
2	TEC-112	1
2	GGB-100	2

Table 9: Un-pivoted format of (student, course, grade) data

- Remove records that have no grades.
- Remove records that have missing values
- Clean course codes from duplicates
- Remove records that have unwanted grades (such as “W” for Withdrawn courses as opposed to A, B, C, D, F)
- The individual grades obtained were in Letter format (A, A-, B+, ..., F), so a new feature is generated to compute Grade Points (GP). The Grade Point is between 0 and 4 using the college grading system as shown below in Table 10.

Grade Letter	Grade Point
A	4
A-	3.7
B+	3.3
B	3
B-	2.7
C+	2.3
C	2
C-	1.7
D+	1.3
D	1
F	0

Table 10: Grading system

4.3.3 Algorithm

To predict course Grade point (0-4) this work uses the Collaborative Filtering approach; a recommender system approach (commonly applied in recommender systems to predict ratings, but here it is used to predict grades). This approach predicts one student's grade on non-graded courses based on similarity with other students. Recent studies started using this approach for predicting students grades such as Elbadrawy and Karypis (2016), Iqbal et al. (2017), Polyzou and Karypis (2016), Ng and Linn (2017), Chang, Lin, and Chen (2016), and Dwivedi and Roshni (2017).

I tested the algorithm on the courses of the three majors: Business, IT, and Engineering. If a student is a Business student, his record could be compared to business students' records to speed up the process. I also tested the algorithm on all records combined, in case a student wants to take courses from different majors.

The algorithm used is Weighted User-Based K-Nearest Neighbor with Pearson Similarity (available through RapidMiner recommender system extension > item rating prediction > user k-NN). It executes a Collaborative Filtering recommender based on (student, course, grade) matrix. The algorithm compares student grades to other students' grades, and similar students are found. By similar, we mean students who took the same courses and achieved close results. The K-Nearest Neighbor in Collaborative Filtering works as follows:

1. The algorithm looks for students who share the same grades patterns with the current student (the student whom the prediction is for).
2. The algorithm measures how similar each student in the database to the current student using K-Nearest Neighbors with Pearson's correlation coefficient as a similarity measure (explained in section 2.2.5)
3. The resulting similarity is used as a weight while calculating the weighted average of the grades of similar students.
4. The resulted grade is used as a prediction for the current student's grade.

An advantage of using collaborative filtering is that there is no need to build a profile of features for each course. The approach has limitations such as cold start and data sparsity (as discussed in section 2.2.5), however, these limitations are at their minimal in our case, since we have ample data, with many students taking the same courses. This method is most useful after year one as per our plan. Hence, the student would have finished some courses, and this avoids the cold start issue and allows for better predictions.

Unlike regression algorithms where we only need to specify the “Label” column to be predicted, in the collaborative filtering we need to specify both the “Label” and the “Item identification” columns. Table 11 shows the feature and the target role assignment in RapidMiner. The k value chosen was 20 (it was found to have the best prediction). The minimum rating is set to 0, and maximum is set to 4 since course grades fall in this range (Please refer to Appendix A, Figure A 5 for a complete list of the parameters).

Feature name	Target role
Mark (or Grade)	Label
Short_Course (shortened Course Code)	Item identification

Table 11: Target role assignment for user-based k-NN in RapidMiner

4.3.4 Results

Table 12 shows the number of records and the RMSE for each major. RMSEs for Business, IT, and Engineering were 0.69, 0.46, and 0.66 respectively. When all records were

combined, RMSE was 0.66. The least error that could be achieved was 0.457 for IT grades prediction.

	Business	IT	Engineering	All combined	Average RMSE
Records	80,324	60,440	86,743	227,507	
RMSE	0.69	0.46	0.67	0.66	0.62

Table 12: Summary of the performance for the Course grade prediction

Even though the error of predicting a Course Grade at this stage (with an average RMSE of 0.62) is larger than previous stages (average Major GPA prediction was 0.46, and Concentration GPA prediction was 0.22 - Figure 22), I believe this is still acceptable, for multiple reasons.

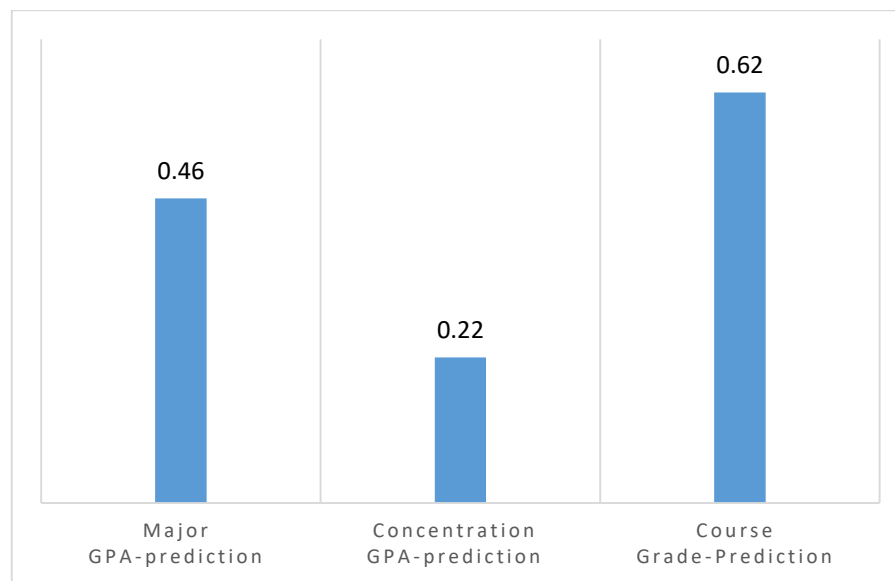


Figure 22: Average RMSEs of the main stages in this research

First, unlike GPA prediction where the GPA is the average of many courses, and most GPAs will be within a smaller range, namely between 1.75 and 4.0 (because students must maintain a GPA above 2.0 to proceed), here we are trying to predict a single course grade. This can take any value in the range between 0-4, and not necessarily in the upper range. This can make the prediction problem harder and the chance of getting a higher error is bigger, because the range of the data is larger.

Second, let us say the actual course grade was 4.0 (A), and the algorithm predicted 3.5 (B+), or even 3.0 (B). This is an error of 0.5 and 1.0 respectively. I would argue that this would not be considered a very far prediction since the grade is still relatively high. It's very unlikely that the algorithm would predict 0(F) or 1 (D) for that student.

Third, the measure used here is RMSE, which penalizes larger errors, hence it is considered stricter as compared to other measures of error such as MAE.

Lastly, the results obtained in this research (average RMSE of 0.62) are comparable (and sometimes better) than other published research predicting course grades on a scale 0-4. For example, Elbadrawy and Karypis (2016) reported an RMSE of 0.65 using collaborative filtering for predicting course grades, and Polyzou and Karypis (2016) reported RMSEs between 0.60 and 0.75 using both linear regression and matrix factorization techniques.

Having said that, I am still very interested in finding ways to improve prediction accuracy to maximize the value of these predictions. I have tried different approaches, such as filtering for only a specific type of courses and concentrations for example, but none of them

improved the performance. I would like to investigate more ways such as incorporating hybrid approaches or improving the prediction algorithm itself.

5. Conclusion and Future Work

Student success is of great importance to students, their families, higher education institutions, society, and nations. Predicting students' future performance can help students, teachers, and advisors make informed choices. This research developed a framework to predict student performance (as measured by GPA or grade) in different Majors, Concentrations, and Courses they are yet to take, using machine-learning.

Literature has covered one area or another, but this research offers comprehensive support to students' decisions throughout their study journey. Multiple machine-learning algorithms were used, and their performance is compared. Furthermore, the strongest predictors of students' performance are identified.

Below are the research questions of this study and a summary of the findings detailed in previous sections:

RQ1: How effectively can student performance in a Major be predicted at enrollment?

1.1. What are the best performing machine-learning algorithms?

At enrollment, student enrollment data (such as high school grades, IELTS scores, college placement tests in English and math) is used to predict student's GPA (scale 0-4) in different majors. Deep learning, Decision Tree, Random Forest, and Gradient

Boosted Trees algorithms are used. Gradient Boosted trees algorithm performed the best in predicting Business and Engineering majors GPA (RMSE 0.469 and 0.45 respectively) while Random forest performed slightly better in predicting Information Technology GPA (RMSE 0.458).

1.1. What are the main predictors of student's GPA in the majors?

For Business and Information technology majors, it was interesting to find out that high-school average was the strongest predictor of students' GPA, while college placement CEPA Math test was the strongest predictor of the Engineering Major GPA.

RQ2: How effectively can student performance in a Concentration be predicted after year one?

2.1 What are the best performing machine learning algorithms?

After year 1, students are asked to choose concentrations within their majors. This study uses year one courses to predict student's GPA at different concentrations in the IT major. The error in predicting Concentration's GPA was considerably smaller than the error in predicting Major GPA (RMSE 0.22 vs 0.46). Deep Learning algorithm achieved the least average RMSE of 0.21, followed by Random Forest and Gradient Boosted Trees (both had an average of 0.25 RMSE). Decision Tree had the least performance with an average RMSE of 0.33.

2.2 What are the main predictors of a student's GPA in concentrations?

"Introduction to Database" course was the strongest predictor for Programming and Networking concentrations, while the "Introduction to Programming" course was of the least predictors. For the Security concentration, Deep Learning algorithm identified the "Information Systems" course as the strongest predictor, while the rest of the algorithms identified the "Web Technologies" course as the stronger predictor.

RQ3: How effectively can student performance in a course be predicted?

At any point after finishing some courses, student's grades of previously finished courses can be used to predict their grade in future courses. A Collaborative Filtering approach using K-Nearest Neighbor is used to predict student grade point (0-4). An average RMSE of 0.62 was achieved. Improving the accuracy of prediction is an area for further exploration.

In future research, I am planning to investigate different methods to optimise the performance of the algorithms used in this research and investigate other algorithms and methods. Furthermore, I am interested in obtaining more data to include other factors in predictions, such as student's feedback about the courses and the teachers.

In order to fully utilize the power of machine learning, it would be of most value to operationalize this work and integrate it within the business solutions currently available for planning and selecting courses and for advising. It would be exciting to show students

the predicted performance in the majors, concentrations, or courses they are interested in. Another significant addition to this work would be to include explanations of predictions, possible interventions, and guidance on how to improve student's chances of success in case they opt for a choice with less predicted GPA. This can be of support to all stakeholders.

It has been an unmatched and pleasant experience to use my knowledge in serving the people I care about. Working on this dissertation enabled me to utilize my newly acquired skills and knowledge in data mining and machine learning for a worthy purpose. I am hoping that the work I did supports students in every stage of their education journey, and helps them make better and more informed decisions on their path to success. There is nothing more satisfying than working to help others prosper.

6. Bibliography

Al-Barrak, M.A. and Al-Razgan, M., 2016. Predicting students final GPA using decision trees: A case study. *International Journal of Information and Education Technology*, 6(7), p.528.

Asif, R., Merceron, A., Ali, S.A. and Haider, N.G., 2017. Analyzing undergraduate students' performance using educational data mining. *Computers & Education*, 113, pp.177-194.

Aulck, L., Velagapudi, N., Blumenstock, J. and West, J., 2016. Predicting student dropout in higher education. *arXiv preprint arXiv:1606.06364*.

Baker, R.S. and Inventado, P.S., 2014. Educational data mining and learning analytics. In *Learning analytics* (pp. 61-75). Springer, New York, NY.

Bautista, R.M., Dumlaio, M., Ballera, M.A. and City, V.A.S.B.Q., 2016, May. Recommendation System for Engineering Students' Specialization Selection Using Predictive Modeling. In *The Third International Conference on Computer Science, Computer Engineering, and Social Media (CSCESM2016)* (p. 34).

Chai, T. and Draxler, R.R., 2014. Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature. *Geoscientific model development*, 7(3), pp.1247-1250.

Chang, P.C., Lin, C.H. and Chen, M.H., 2016. A hybrid course recommendation system by integrating collaborative filtering and artificial immune systems. *Algorithms*, 9(3), p.47.

Del Río, C.A. and Insuasti, J.A.P., 2016. Predicting academic performance in traditional environments at higher-education institutions using data mining: A review. *Ecos de la Academia*, 2016(7).

Dutt, A., Ismail, M.A. and Herawan, T., 2017. A systematic review on educational data mining. *IEEE Access*, 5, pp.15991-16005.

Dwivedi, S. and Roshni, V.K., 2017, August. Recommender system for big data in education. In *E-Learning & E-Learning Technologies (ELELTECH)*, 2017 5th National Conference on (pp. 1-4). IEEE.

Elbadrawy, A. and Karypis, G., 2016, September. Domain-aware grade prediction and top-n course recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems* (pp. 183-190). ACM.

Elbadrawy, A., Polyzou, A., Ren, Z., Sweeney, M., Karypis, G. and Rangwala, H., 2016. Predicting student performance using personalized analytics. *Computer*, 49(4), pp.61-69.

Goga, M., Kuyoro, S. and Goga, N., 2015. A recommender for improving the student academic performance. *Procedia-Social and Behavioral Sciences*, 180, pp.1481-1488.

Ihantola, P., Vihavainen, A., Ahadi, A., Butler, M., Börstler, J., Edwards, S.H., Isohanni, E., Korhonen, A., Petersen, A., Rivers, K. and Rubio, M.Á., 2015, July. Educational data mining and learning analytics in programming: Literature review and case studies. In *Proceedings of the 2015 ITiCSE on Working Group Reports* (pp. 41-63). ACM.

Iqbal, Z., Qadir, J., Mian, A.N. and Kamiran, F., 2017. Machine Learning Based Student Grade Prediction: A Case Study. arXiv preprint arXiv:1708.08744.

Isinkaye, F.O., Folajimi, Y.O. and Ojokoh, B.A., 2015. Recommendation systems: Principles, methods and evaluation. Egyptian Informatics Journal, 16(3), pp.261-273.

Jones, Z. and Linder, F., 2015, April. Exploratory data analysis using random forests. In Prepared for the 73rd annual MPSA conference.

Knox, S.W. & Overdrive.com 2018, Machine learning: a concise introduction, John Wiley & Sons, Inc, Hoboken, NJ.

Lison, P., 2015. An introduction to machine learning.

Liu, H., Hu, Z., Mian, A., Tian, H. and Zhu, X., 2014. A new user similarity model to improve the accuracy of collaborative filtering. Knowledge-Based Systems, 56, pp.156-166.

Manhães, L.M.B., da Cruz, S.M.S. and Zimbrão, G., 2014, March. WAVE: an architecture for predicting dropout in undergraduate courses using EDM. In Proceedings of the 29th Annual ACM Symposium on Applied Computing (pp. 243-247). ACM.

Mostafa, L., Oatley, G., Khalifa, N. and Rabie, W., 2014, March. A case based reasoning system for academic advising in egyptian educational institutions. In 2nd International Conference on Research in Science, Engineering and Technology (ICRSET'2014) March (pp. 21-22).

- Naser, S.A., Zaqout, I., Ghosh, M.A., Atallah, R. and Alajrami, E., 2015. Predicting student performance using artificial neural network: In the faculty of engineering and information technology. *International Journal of Hybrid Information Technology*, 8(2), pp.221-228.
- Natekin, A. and Knoll, A., 2013. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7, p.21.
- Ng, Y.K. and Linn, J., 2017, August. CrsRecs: A personalized course recommendation system for college students. In *Information, Intelligence, Systems & Applications (IISA), 2017 8th International Conference on* (pp. 1-6). IEEE.
- Nithya, P., Umamaheswari, B. and Umadevi, A., 2016. A survey on educational data mining in field of education. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 5(1), pp.69-78.
- Osmanbegović, E. and Suljić, M., 2012. Data mining approach for predicting student performance. *Economic Review*, 10(1), pp.3-12.
- Patil, A.P., Ganesan, K. and Kanavalli, A., 2017, December. Effective Deep Learning Model to Predict Student Grade Point Averages. In *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*(pp. 1-6). IEEE.
- Peña-Ayala, A., 2014. Educational data mining: A survey and a data mining-based analysis of recent works. *Expert systems with applications*, 41(4), pp.1432-1462.

Polyzou, A. and Karypis, G., 2016. Grade prediction with models specific to students and courses. *International Journal of Data Science and Analytics*, 2(3-4), pp.159-171.

Sara, N.B., Halland, R., Igel, C. and Alstrup, S., 2015. High-school dropout prediction using machine learning: A danish large-scale study. In *ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence* (pp. 319-24).

Shahiri, A.M. and Husain, W., 2015. A review on predicting student's performance using data mining techniques. *Procedia Computer Science*, 72, pp.414-422.

Sin, K. and Muthu, L., 2015. Application of big data in education data mining and learning analytics--A literature Review. *ICTACT journal on soft computing*, 5(4).

Sorour, S.E., Mine, T., Goda, K. and Hirokawa, S., 2015. A predictive model to evaluate student performance. *Journal of Information Processing*, 23(2), pp.192-201.

Sukhija, K., Jindal, M. and Aggarwal, N., 2015, October. The recent state of educational data mining: A survey and future visions. In *2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE)* (pp. 354-359). IEEE.

Sweeney, M., Rangwala, H., Lester, J. and Johri, A., 2016. Next-term student performance prediction: a recommender systems approach. *arXiv preprint arXiv:1604.01840*.

Tekin, A., 2014. Early Prediction of Students' Grade Point Averages at Graduation: A Data Mining Approach. *Eurasian Journal of Educational Research*, 54, pp.207-226.

Upendran, D., Chatterjee, S., Sindhumol, S. and Bijlani, K., 2016. Application of predictive analytics in intelligent course recommendation. *Procedia Computer Science*, 93, pp.917-923.

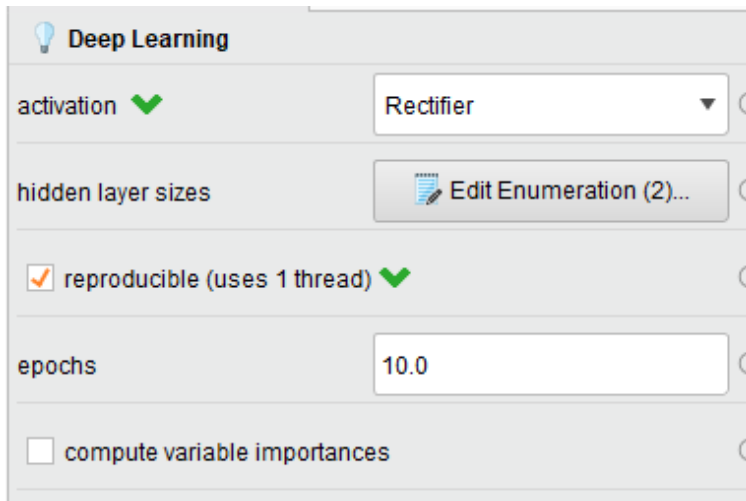
Witten, I.H., Frank, E., Hall, M.A. and Pal, C.J., 2016. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Wolff, A., Zdrahal, Z., Nikolov, A. and Pantucek, M., 2013, April. Improving retention: predicting at-risk students by analysing clicking behaviour in a virtual learning environment. In *Proceedings of the third international conference on learning analytics and knowledge* (pp. 145-149). ACM.

Yang, S.J., Lu, O.H., Huang, A.Y., Huang, J.C., Ogata, H. and Lin, A.J., 2018. Predicting Students' Academic Performance Using Multiple Linear Regression and Principal Component Analysis. *Journal of Information Processing*, 26, pp.170-176.

Yukselturk, E., Ozekes, S. and Türel, Y.K., 2014. Predicting dropout student: an application of data mining methods in an online education program. *European Journal of Open, Distance and E-learning*, 17(1), pp.118-133.

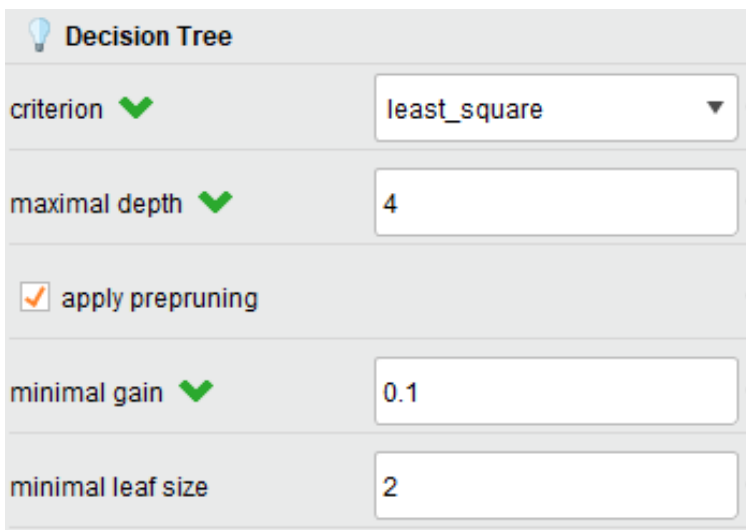
Appendix A



The screenshot shows the configuration panel for the 'Deep Learning' operator. It includes a title bar with a lightbulb icon and the text 'Deep Learning'. Below this, there are several parameter rows. The first row is 'activation' with a green checkmark icon and a dropdown menu set to 'Rectifier'. The second row is 'hidden layer sizes' with a button labeled 'Edit Enumeration (2)...'. The third row is 'reproducible (uses 1 thread)' with a checked checkbox and a green checkmark icon. The fourth row is 'epochs' with a text input field containing '10.0'. The fifth row is 'compute variable importances' with an unchecked checkbox. Each row has a small information icon (i) on the right side.

Deep Learning	
activation	Rectifier
hidden layer sizes	Edit Enumeration (2)...
<input checked="" type="checkbox"/> reproducible (uses 1 thread)	
epochs	10.0
<input type="checkbox"/> compute variable importances	


Figure A 1: Parameters of Deep Learning operator in RapidMiner



The screenshot shows the configuration panel for the 'Decision Tree' operator. It includes a title bar with a lightbulb icon and the text 'Decision Tree'. Below this, there are several parameter rows. The first row is 'criterion' with a green checkmark icon and a dropdown menu set to 'least_square'. The second row is 'maximal depth' with a green checkmark icon and a text input field containing '4'. The third row is 'apply prepruning' with a checked checkbox. The fourth row is 'minimal gain' with a green checkmark icon and a text input field containing '0.1'. The fifth row is 'minimal leaf size' with a text input field containing '2'. Each row has a small information icon (i) on the right side.

Decision Tree	
criterion	least_square
maximal depth	4
<input checked="" type="checkbox"/> apply prepruning	
minimal gain	0.1
minimal leaf size	2

Figure A 2: Parameters of Decision Tree operator in RapidMiner

 **Random Forest**






number of trees 	<input type="text" value="100"/>
criterion 	<input type="text" value="least_square"/>
maximal depth 	<input type="text" value="7"/>
<input checked="" type="checkbox"/> apply prepruning 	
minimal gain	<input type="text" value="0.01"/>
minimal leaf size	<input type="text" value="2"/>
<input checked="" type="checkbox"/> guess subset ratio	

Figure A 3: Parameters of Random Forest operator in RapidMiner

💡 Gradient Boosted Trees	
number of trees	60
<input checked="" type="checkbox"/> reproducible	
maximum number of threads	1
maximal depth	2
min rows	10.0
min split improvement	0.0
number of bins	20
learning rate	0.1
sample rate	1.0
distribution	AUTO

Figure A 4: Parameters of Gradient Boosted Trees operator in RapidMiner

 **User k-NN (2) (User k-NN)**








k	<input type="text" value="20"/>	
Min Rating	<input type="text" value="0"/>	
Range	<input type="text" value="4"/>	
Correlation mode	<input type="text" value="pearson"/>	
<i>reg u</i>	<input type="text" value="10.0"/>	
<i>reg i</i>	<input type="text" value="5.0"/>	
<i>schrinkage</i>	<input type="text" value="10.0"/>	

Figure A 5: Parameters of User k-NN operator in RapidMiner