Accurate and Personalized Academic Advising

Mohammad Poul Doust, Luis Galarraga, Gonzalo Gabriel Mendez

June 24, 2019

IRISA, INRIA, Universite Rennes 1, 35042 Rennes, France

Abstract

With the exponential growth of data generated through academic processes, Educational Data Mining (EDM) has emerged as a new research area which uses Machine Learning (ML), Data Mining (DM), and Statistics in educational settings. In this report we focus on the problem of academic advising through performance prediction. Being able to predict how a student will perform in a future course would make it possible to avoid academic problems by making proactive interventions. To that goal, we explore the different stages to design a system that will predict future performance based on historical data, starting from how to prepare the dataset and process it to extract insightful indicators and, finally training a Machine Learning algorithm for prediction. The dataset adopted in this work was taken from the Electrical and Computer Engineering College of ESPOL University in Guayaquil. The final goal is to build an interpretable academic advising system by employing domain knowledge to design hand-crafted features along with a machine learning algorithm. This system will help students to make better decisions when registering for new courses by simulating their performance for the next semester basing on historical data and the overall difficulty of newly registered courses taken at the same time. The techniques used in this report could be applied to any historical data to provide guided predictions about students performance. Interestingly, the suggested hand-crafted features proved to be valuable for prediction (reduced error rate by a factor of 23%). Finally, the results were re-assuring that previous academic history - along with other hand-crafted features - greatly impact the performance in future courses. Consequently, the achieved results were stable and encouraging to deploy the system into use (an average Mean Absolute Error of 0.6).

1 Introduction

With the constant growth of historical academic data, it is becoming difficult for human experts to process it manually and hence, more difficult to make data-driven academic advising. Therefore, Statistics, Data Analysis, Data Mining and Machine Learning techniques have been broadly used in this domain to help make better use of this data and provide more precise insights and hence, better decision-making.

Courses selection is an important decision that highly affect students' academic performance. Generally, the university's program is represented as a dependency graph where nodes are courses and edges are the dependency between them as shown in Figure 5. Furthermore, there are compulsory courses and optional ones. Students face difficulties choosing which courses to take at each semester, mainly because the little knowledge they have about the content of each course. Moreover, students differ in backgrounds and interests, hence, each one needs personalized guidance to make this decision. [1]. In this report, we focus on the problem of academic advising to help students in course selection. This is done by predicting student's grade in future courses by: firstly, prepare the data and transform it into more convenient form to be processed efficiently, secondly, using domain knowledge to extract meaningful academic indicators that affect student's grade, and finally feed them to a Machine Learning algorithm to predict student's final grade.

Having a white-box academic advising system will be helpful in many other scenarios also. For example, interpreting the model of each course will help programs coordinators to make guided decisions on the curriculum design. Moreover, will help educational institutions and students as well by effectively increase passing percentage and, hence decreasing dropouts.

The rest of the report is structured as follows: In section 2, we present related works. Section 3, describes the dataset and preprocessing phase. Section 4, discusses features engineering stage in details. Sections 5, briefly provides general background about the techniques used in this work. Section 6, presents the methodology followed along with the analysis for different case studies to predict the grade of a student with respect to previously discussed features. Section 7, presents the final results. Section 8, describes future work and discusses new avenues worth exploring to further improve the results. Finally, Section 9, concludes and highlight the important points of this study.

2 Related Work

The problem of academic data analysis and automated academic advising has been widely studied in the literature. In [2], several analytic techniques are suggested to help coordinators get more accurate idea about the possible problems in the program. The proposed techniques are applied on grades and could be summarized as techniques for: Course Difficulty estimation, Curriculum coherence and Dropouts path. Regarding difficulty estimation, the authors used two difficulty estimators: Multiplicative Magnitude (α) and Grading Stringency Index (β)[3], these estimators are based on the idea that the difficulty and importance of a course could be seen as how much a given course affects student's

GPA. These estimators were used to find difficult courses causing problems to students and have been studied on course level and teacher-course level as well. While for the curriculum coherence, an Exploratory Factor Analysis (EFA) has been performed and as result~~s~~ several factors are extracted and interpreted in the light of the curriculum. Finally, for dropout paths, sequence mining techniques have been proposed to extract frequent sequences of courses most related to dropping out of school.

[4] Performed a comparative study of classification and regression methods to predict a student's approval status (fail, success) and the student's grade in the course. This is done by building two models (regression and classification) for each course independently. Furthermore, same non academic features were used for both models. For instance, age, sex, nationality, delayed courses, etc. Several methods were applied and evaluated for each problem. For classification, SVM (Support Vector Machines) performed the best between all methods, with average F1 score of 0.60. Similarly, for regression, SVM as well performed the best with avg RMSE of 4.65.

[5] Used SVM to classify student's performance into 3 categories based on CGPI (Cumulative Grade Percentile Index): High, Average and Low using psychological-based features that relates to: personality, motivation, socioeconomic factor and learning strategies. The method achieved an accuracy of 90% approximately using SVM with a Radial Basis Function kernel.
Similarly, [6] compared different classification algorithms to predict student's GPA class (High, Medium, Low). Specifically, Decision Trees, K- Nearest Neighbours (KNN), and Naive Bayes have been applied. The feature set consisted of 20 non academic indicators such as, sex, native language, marital status, family annual income, parents' qualifications, etc. KNN performed the best with classification accuracy of 0.54. [7] focuses on predicting the final student's GPA as a numerical value instead of predicting a class. To that goal, several regression algorithms have been compared using features like theory marks, term test marks, and practical marks. Linear Regression achieved best results between other regression models(Knn, Decision Tree, SVM and Random Forest). The study confirmed that previous marks are vital features to decide final results of student (GPA). This system was applied to assign tutors to students that are predicted to need help.

[8] Tackled the problem of predicting student grade as a recommendation problem with a two-stage user-based collaborative filtering. Moreover, for prediction, students were clustered using artificial immune systems tested using the Karl Pearson coefficient and the Cosine similarity measure. The adopted method clusters the students based on all their previous marks, after that the prediction for newly not taken courses will be made based on the similarity to the students in the same group. The best result that was achieved in all experiments was Mean Average error of 0.5719 with the cosine weighted similarity. remove the number, just mention the configuration that led to the best results

In [9], the authors employed Electrical Circuit Theory to advise the student about choosing between an Electrical & Computer Engineering path. As for features, it was only based on historical data, after calculating pairwise correlations between all courses and the target course, the grades for 10 courses with highest correlation have been chosen as features. The dataset in total was of 200 students' record. For classification methods, two algorithms have been tested, Naive Bayes and Random forest. Both algorithms gave comparable results (around 70% of accuracy). However, Random forest comes with the benefit of providing a feature importance analysis. It was found that for the Electrical & Computer Engineering path (represented as Electrical Circuit Theory), Mathematics, Physics and Chemistry have the most predicting power and students could be advised accordingly based on these marks.

In [10], the authors focused on the problem of early predicting student's GPA to help non-performant at an early time. To that goal, several machine learning algorithms have been tested and tuned, namely Neural Networks (NN), Support Vector Machine (SVM) and Extreme Learning Machine (ELM). As for the dataset, it contains a total of 127 student records that covers scores for 49 different courses. Furthermore, two scenarios were tested. Firstly, predicting the final GPA based only on the first two years scores (covering 24 courses). While for the latter scenario, the first three years scores will be taken into account (covering 38 courses). As for testing, 5-fold cross validation has been used with testing data containing around 25 records. Among the tested algorithms, for the first scenario, SVM achieved best results with Root Mean Square Error (RMSE) of 0.1146. On the other hand, Neural networks achieved the worst (RMSE: 0.2068) where NN was with one hidden (25 neurons for first scenario and 39 neurons for the second), for input 24 and 38 neurons depending on the scenario and one output neurons. The performance of NN could be explained that the dataset is not sufficient to train a neural network with this size. Similarly, for the second scenario, the performance got better with the same models. SVM still performing the best with (RMSE: 0.0708).

Finally, [11] compared different approaches to design a feedback model that is made of different steps. Firstly, build student's profile, which is a matrix R where rows represents the students and columns represent the courses. R_{ij} is student i mark in course j, if the student i have not taken course j yet, the value would be empty. Secondly, predict the student i grade in course j that has not been taken yet by student i. To that end, several algorithms have been applied. Namely, Matrix Factorization (MF) using two techniques: Singular Value Decomposition (SVD) and Non-Negative Matrix Factorization, Restricted Boltzmann Machines (RBM) and User-based Collaborative Filtering (UBCF). In case of MF, it was used to decompose the dataset matrix R into student-feature space and course-feature space. Gradient descent algorithm would be able to specify a smaller rank for dataset matrix and yet still be meaningful. Moreover, for RBM, it is a probabilistic model that is represented as a bipartite graph with

Work	Year	Methods	Description
Curricular design analysis: a data-driven perspective [2]	2014	Data Analysis, Sequence Mining, Factor Analysis	Data driven analysis to find: Course Difficulty Estimators, Dropout paths
A Comparative Study of Classification and Regression Algorithms for Modelling Students' Academic Performance [4].	2015	Support Vector Machines (SVM)	Predicting GPA as numerical value and class (if student will approve) using SVM and non academic features
Predicting Students Academic Performance Using Support Vector Machine [5]	2019	SVM RBF Kernel	Using psychological features to predict performance
Comparative study of supervised learning algorithms for student performance prediction [6]	2019	Decision Trees, K-Nearest Neighbours (KNN) and Naive Bayes	Predicting student GPA as class (High, Medium, Low) using non academic features
Prediction of Student's Performance Using Machine Learning[?]	2019	Compared different algorithms: KNN, Decision Trees, SVM and Random Forests	Predicting GPA as numerical value, the academic history as features proved to be most efficient features
A hybrid course recommendation system by integrating collaborative filtering and artificial immune systems [8]	2016	Collaborative Filtering, Artificial Immune system	Predicting GPA as numerical value by using collaborative filtering (similar students get similar results)
Predicting Academic Performance via Machine Learning Methods [9]	2017	Naive Bayes and Random Forests	Predicting Grade class in Electrical Circuits course as representative course to choose Electrical Engineering path.
Early Prediction of Students' Grade Point Averages at Graduation: A Data Mining Approach [10].	2014	Neural Networks, SVM, Extreme Learning Machine	Predicting GPA as numerical value using academic history. Early prediction is used to provide early help for student in need
Machine learning based student grade prediction: A case study [11]	2017	User-based Collaborative filtering with Dimensionality reduction; Hidden Markov model	Designed system that models student's knowledge in each domain (user collaborative filtering) and the inference is done using Hidden Markov model

Table 1: Related Work

two layers of nodes: Visible Layer (Course Grades) and Hidden Layer linked with weights. It is widely used in recommender systems. Lastly, for UBCF, it is one of the most popular techniques in recommender systems, which is based on the idea that if a student is similar to other student in common courses marks, most probably they will have same marks for the non-common courses. Similar students are found using k-nearest neighbours and the GPA is predicted using weighted average (similarity as weight) for the similar students to the target student. The third step in the feedback system is to compute the student knowledge in specific domain by averaging the student's grades in all courses belonging to each domain. The final component is the inference step using a Hidden Markov Model to compute the student knowledge in a specific course based on the previously calculated domain knowledge to warn the student to put extra effort.

Overall, despite the fact that this problem has been studied thoroughly in literature, none of them explored the features suggested in this work yet. Specifically, finding the impact of course difficulty, semester course-load and repeating a course on final student's grade. Additionally, the interpretability of the model was not taken into account in literature.

3 Dataset & Preprocessing

3.1 Dataset Description

The dataset used in this report represents the academic history (from 1978 to 2012) of students in the Electrical Engineering faculty of ESPOL University in Guayaquil, Ecuador. It consists a total of 8924 record of students grades. The grades are real values in the range [0 - 10] and the student is passing a given course if he/she scores a grade equal or greater than 6. Moreover, there are selective and mandatory courses where students manage their choices for each semester. The dataset comes into three CSV files representing three database tables as shown in Figure 1.

- Student: contains student basic information
- Course: contains course basic information
- AcademicHistory: contains basic information about pair of course-student, like student's grade in a specific course

The dataset is extended by adding course's credits information using a scraper that fetches the needed information from University's website.

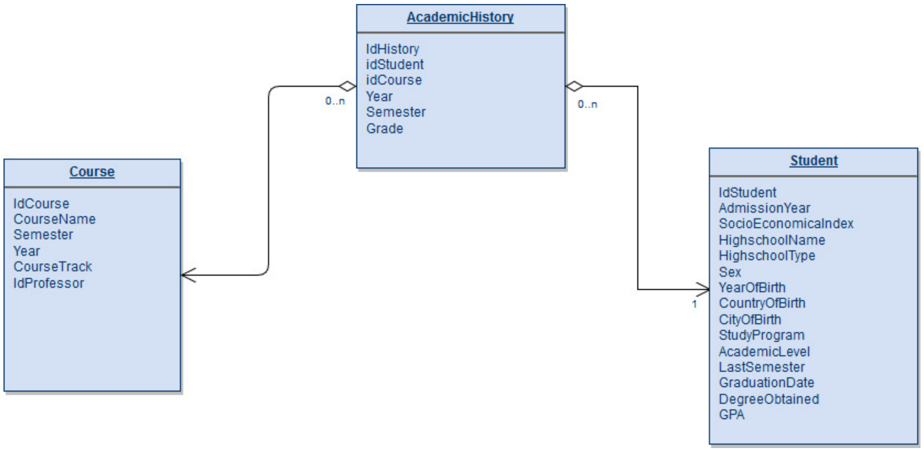


Figure 1: Dataset Structure

3.2 Data Preprocessing

In this section we will present the pre-processing steps that have been conducted in order to process the data to clean it and transform it into a more convenient structure for analysis and to be fed later to different machine learning algorithms. Firstly, we perform data anonymization on the dataset, Secondly, we process all the three data tables and join them into one table to make it easier to analyze later. Lastly, we transform the shape of the data (by aggregating) into a more convenient form to better suits the methods adopted in this study.

3.2.1 Data Anonymization

Data anonymization is a technique for privacy protection in data so that the people identity remain anonymous and untraceable directly, or indirectly. This is done by altering and removing the personal information that identify the individual. For instance in our case, student's name, last name, family name, address, phone number and post code have been removed from the dataset as well as for professors. New incremental fields are added as an id for each record. The new anonymized dataset is depicted in Figure 1.

3.2.2 Data Integration

Data integration is a technique that is used to combine data from different sources into one homogeneous dataset suitable for further analysis. The simplest version of data integrity is applicable to the dataset in this study, since the dataset is scattered into 3 different tables. We combined them together into one table by joining the tables by the defined foreign keys between them. Consequently, one compact dataset table is formed and stored to be used as the main dataset file. Moreover, as the number of credits of the courses are not included in the dataset, we implemented a Web Scraper to scrape the University website and store the credits information for each course. Finally, the scraped data were merged into the final dataset.

3.2.3 Data Cleansing

Data cleansing is a process in which the dataset is cleaned from any false, corrupted and inconsistent data. This is done by removing and correcting the false records. It is an important step that has effects on all following steps and the final results as known of "Garbage in, Garbage out" [12]. Consequently, several cleaning techniques have been performed. Specifically for, Data Consistency, Data Integrity, Missing Values and Duplication.

3.2.3.1 Data Consistency

Several inconsistent naming convention have been spotted in the dataset, we will present some of them in this section. For instance, as we can see in Table3, the field HighschoolTyoe have 3 different values, like: "Particular", "Fiscal", "Nacional" (National), etc. However, Fiscal and Nacional are the same. Consequently, we replaced all the occurrences of Nacional with Fiscal. Similarly, for the field Semester, it have different possible values, such as: 1S, 2S, 3S, 1T, 2T, etc. Despite the fact that S (for semester) is equivalent to T (for Term) and represent the same thing. Furthermore, the GradeDate field ~~is~~ contains non-date values (Not Graduate) as depicted in Table 3. These inconsistencies will cause problems during the analysis. To avoid that, such values have been replaced

StudentID	AdmissionYear	SocioIndex	Sex	HighschoolType	BirthYear	BirthCountry	GradDate	GPA	CourseID	CourseDate	Semester	GRADE	CoursePracticalCredits	CourseTheoriticalCredits
1	2003	7	M	Particular	1977	ALEMANIA	14/12/2006	8.097058824	FIEC04341	2003	1S	8,30	1	4
1	2003	7	M	Particular	1977	ALEMANIA	14/12/2006	8.097058824	FIEC04358	2003	1S	7	0	4
41	2000	3	F	Fiscal	1978	ECUADOR	14/08/2003	7.814706	FIEC04341	2003	1T	9	1	4
41	2000	3	F	Fiscal	1978	ECUADOR	14/08/2003	7.814706	ICF00471	2003	1S	8,60	3	0
120	2000	5	M	Nacional	1981	ECUADOR	Not Graduate	7.036207	ICF00471	2003	1T	5,50	3	0
120	2000	5	M	Nacional	1981	ECUADOR	Not Graduate	7.036207	ICF00471	2004	2T	8,10	3	0

Table 2: Data Integration Sample

StudentID	AdmissionYear	SocioIndex	Sex	HighschoolType	BirthYear	BirthCountry	GradDate	GPA	CourseID	CourseDate	Semester	GRADE
1	2003	7	M	Particular	1977	ALEMANIA	14/12/2006	8.097058824	FIEC04341	2003	1S	8.30
1	2003	7	M	Particular	1977	ALEMANIA	14/12/2006	8.097058824	FIEC04358	2003	1S	7
41	2000	3	F	Fiscal	1978	ECUADOR	14/08/2003	7.814706	FIEC04341	2003	1S	9
41	2000	3	F	Fiscal	1978	ECUADOR	14/08/2003	7.814706	ICF00471	2003	1S	8.60
120	2000	5	M	Fiscal	1981	ECUADOR	NA	7.036207	ICF00471	2003	1S	8.10

Table 3: Data Cleansing

with Nulls (NA) to mark students who have not yet graduated. Finally, the GRADE field contains values with floating point as Listing Comma "," instead of decimal separator, which make this field not parsable to numeric values.

3.2.3.2 Data Integrity

To maintain data integrity, we have deleted all records that have no link with other tables after joining them. For example, some students do not enrol in any course.

3.2.3.3 Missing Values & Duplication Removal

Rows with empty main attributes were removed. For instance, rows with all empty grades were removed, Furthermore, full-duplicated rows were removed. Similarly, as the student may take a course more than once in cause of failure, we have duplication records for the same student-course. Consequently, we kept the latest of them in our dataset. However, those duplicated records are employed as features as we will see in the next section.

3.2.4 Data Transformation

Table 3, shows the dataset after data integration, where all separate tables have been combined into one table. Rows represent students and columns are student's attributes. As we can notice each record holds information about student's grade in a specific course - along with other personal information, meaning there are more than one record for the same student. However, this form is not suitable for the suggested methods of this study. Consequently, data transformation and reduction was applied so each record would represent

StudentID	AdmissionYear	SocioIndex	Sex	HighschoolType	BirthYear	BirthCountry	GradDate	GPA	FIEC04341	FIEC04358	ICF00471	CourseDate
1	2003	7	M	Particular	1977	ALEMANIA	14/12/2006	8.097058824	8.30	7	NA	2003
41	2000	3	F	Fiscal	1978	ECUADOR	14/08/2003	7.814706	9	NA	8.60	2003
120	2000	5	M	Fiscal	1981	ECUADOR	NA	7.036207	NA	NA	8.10	2003

Table 4: Data Transformation Sample

only one student by removing Course ID column and scatter its values as new column. The values of new column will be filled by student's mark in each course accordingly. Finally, the column GRADE is removed. As result, as we can see in Table 4, each row now represents the complete academic history for a given student. Courses that have not been taken by a student yet are filled with NA values.

4 Features Engineering

Feature engineering is one of the most fundamental phases in any data analysis procedure as it affects all the following steps and consequently the final model performance[13]. Usually, it involves employing domain knowledge from human expertise to extract meaningful indicators from the raw data[14]. For the case of this report, several features will be adopted, namely repeating frequency, semester Load, course difficulty estimators and last achieved mark.

4.1 Course Difficulty Estimators

Having a single numerical value that estimates the difficulty of a course would be an interesting feature for predicting student's performance. However, extracting such feature is not a trivial task. In this report, we will be using difficulty estimators: Multiplicative Magnitude (α) and Grading Stringency Index (β)[3] presented in [3], [2]. The estimators are computed according to equations (1 , 2) respectively.

$$\alpha_j = \frac{\sum_i GPA_i^2}{\sum_i (r_{ij} * GPA_i)} \tag{1}$$

$$\beta_j = \frac{\sum_i (GPA_i - r_{ij})}{N_s^j} \tag{2}$$

Where:

- GPA_i is the total GPA for student i
- r_{ij} is student's i grade on course j
- N_s^j is the overall number of students enrolled in course j

Moreover, as for β we can see it is basically the average of how much GPA is affected in a specific course j and therefore, the smaller the beta is, the easier the course is, since the negative part (-r_{ij}) will dominate. Similarly, the alpha estimates how much on average the ratio of GPA is affected by the grade of course j but in a multiplicative manner. To conclude, we use the mentioned estimators as a feature for each semester by using the following equation:

$$Feature\,\beta_{\,i,sem} = \sum_{j}^{courses(i,sem)} \beta_j \tag{3}$$

$$Feature\,\alpha_{\,i,sem} = \prod_{j}^{courses(i,sem)} \alpha j \qquad (4)$$

Simply, we are aggregating the grades of each student (i) in all enrolled courses in semester (sem, the semester of the target grade we want to predict). For α aggregation is done by summing the individual values whereas for β ~~its~~ the aggregation is multiplicative.

4.2 Repeating Frequency

The number of times a student repeated a course could be a good indicator of how the student will perform in the target course. For each course, an additional feature will be added, representing how many times the student has taken the corresponding course before. Normally, if the student passed a course from the first time, this feature will be zero. It is calculated by aggregating the number of time the same course taken by the same student. However, this aggregation will account for the currently predicted course which should not be counted.

4.3 Last achieved mark in a course

Much like the number of times a student repeated a course plays a role in the prediction of performance, similarly, previous marks can also help in more accurate prediction. However, we have identified two scenarios. First one, we take the average of all previous repetitions. In The second scenario we take the grade of the latest repetition before the passing mark (since it might represent the last state of mind for the student). Moreover, another issue would be non-repeaters since they have no previous marks for the predicted course. Consequently, we will try two separate models for repeaters and non repeaters. Finally, we will address the problem of building one model for both of them. The main problem is the different number of features for each model which will lead to undefined features for non-repeaters (no previous marks). This is tackled using imputation technique as explained in Section 6.5.

4.4 Semester Load

One important factor that affects student's performance in a give semester could be the total load of subjects taken in the semester. Logically, the more taken courses the worse the performance would be. However, also the weight of the course would play a role. Consequently, a new feature was added as the sum of credits taken at the semester of the target prediction course.

5 Background

In this study, Regularized Linear Regression has been adopted as a Machine Learning algorithm for its simplicity and interpretability. Moreover. it is im-

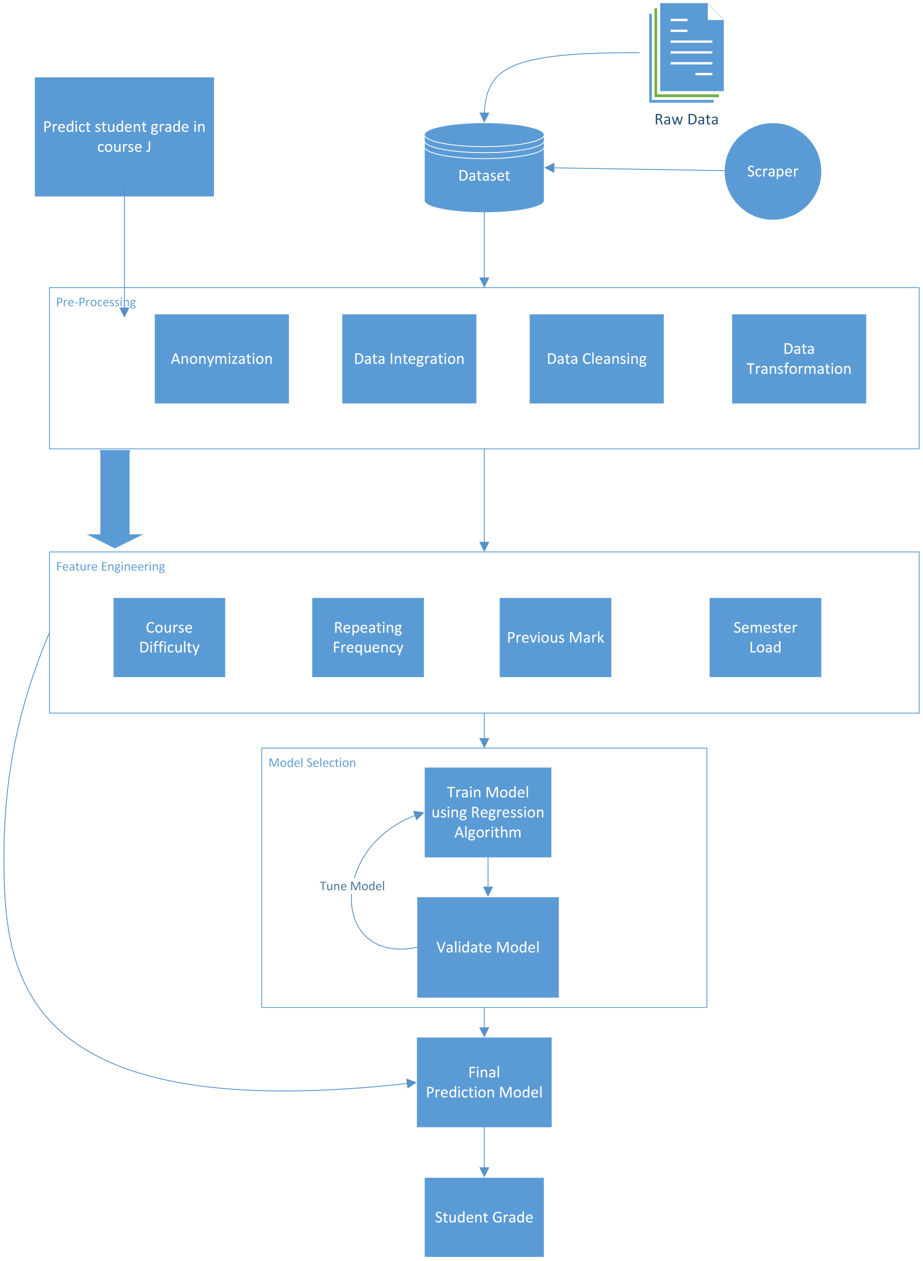
#	Attribute	Type	Description
1	AdmissionYear	Numerical	Year of admission
2	SocioEconomicalIndex	Numerical	Social Factor
3	Sex	Nominal	-
4	HighschoolType	Nominal	-
5	HighschoolName	Nominal	-
5	YearOfBirth	Numerical	-
6	CityOfBirth	Nominal	-
7	Year_Statistics	Numerical	Year of taking statistics course
8	Semester_Statistics	Nominal	-
9	ProfessorID	Numerical	
10	Grade_Statistics	Numerical	Student's Grade in Statistics
11	Grade_LinearAlgebra	Numerical	Student's Grade in Linear Algebra
12	Grade_DifferentialCalculus	Numerical	Student's Grade in Differential Calculus
13	Grade_IntegralCalculus	Numerical	Student's Grade in Integral Calculus
14	Count_Statistics	Numerical	How many time the student taken Statistics before
15	Count_LinearAlgebra	Numerical	How many time the student taken Linear Algebra before
16	Count_DifferntialCalculus	Numerical	How many time the student taken Differential Calculus before
17	PrevGrade_Statistics	Numerical	Student's Previous grade in Statistics. if first time ,it is the average between all non repeaters
18	PrevGrade_DifferentialCalculus	Numerical	Student's Previous grade in Differential. if first time, it is the average between all non repeaters
19	PrevGrade_IntegralCalculus	Numerical	Student's Previous grade in Integral Calculus. if first time, it is the average between all non repeaters
20	SumAlpha	Numerical	The sum of Alphas for courses taken by student in the semester of Statistics
21	SumBeta	Numerical	The sum of Betas for courses taken by student in the semester of Statistics
22	Semester_LOAD	Numerical	Student's Semester in LOAD

Table 5: Case Study A (Statistics) - Features

Case Study	Courses	Dataset Size
A - Algorithms Analysis	Programming Fundamentals, Data Structure and Discrete Mathematics	87
B - Statistics I	Multivariate Calculus, Linear Algebra, Differential Calculus and Integral Calculus	654
C - Statistics II (Repeaters)	Calculus I, Calculus II and Linear Algebra	566
D - Statistics II (Non-Repeaters)	Calculus I, Calculus II and Linear Algebra	938
E - Statistics II (All)	Calculus I, Calculus II and Linear Algebra	1503
F - Programming Languages	Data Structures, Object Oriented Programming and Programming Fundamentals	163
G - Analysis of Electrical Networks	Calculus I, Calculus II, Calculus III, Linear Algebra, Physics I and Physics II	974

Table 6: Case Studies

Figure 2: Main Approach



portant for this report to discuss each model and have interpretable model to study to build reasonable and white box advising system. Furthermore, Attributes Selection have been employed in cooperate with Linear Regression. Specifically, M5 Prime algorithm is used for that purpose.

5.1 Regularized Linear Regression

Linear Regression is one of the most common machine learning algorithms. It is used to find linear relationship between target variable and indicators. This is done by minimizing an error function which is the Mean Squared Error in this case. minimizing this error function is equivalent to finding a line that best fits the data point with (the distance between the line and points as small as possible). Moreover, to prevent the model from over-fitting the data points, several techniques have been explored, in this report we will use Linear Regression with L2 Norm penalization term (Ridge Linear Regression)[15] and the error function J(θ) would be as follows:

$$J(\theta) = \tfrac{1}{2m}\left[\textstyle\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

$$\begin{aligned} y^{(i)} &= \text{ the real value to predict for instance i} \\ h_\theta(x^{(i)}) &= \text{ the predicted value for instance i} \\ \theta_j &= \text{ the weight for feature j} \\ \lambda &= \text{ ridge parameter} \\ m &= \text{ size of dataset} \\ n &= \text{ size of features set} \end{aligned} \tag{5}$$

5.2 Features Selection - M5 Prime

M5' is tree-based features selection algorithm [16], which use Standard Deviation Reduction (SDR) as an impurity measure. At each iteration it removes an attribute with the least impurity factor and evaluates the model quality, if it is increased, the attribute will be removed from the final features set. This process is repeated until all remaining attributes are important for model quality. This method has been applied for features selection in the following case studies.

6 Methodology

In this section, we will be presenting the main approach to predict student's future performance using the previously processed data. However, one challenge is to decide whether to build a separate model for each course or only one model for all courses. The latter choice would require representing the dataset as a big matrix with dimension of (4800 x 2097) where rows are students and columns are all possible courses. Having such a large number of redundant features would certainly make the model complex and hence prone to over-fit (Curse of Dimensionality). Furthermore, the average student enrol to 5 courses on each semester, that means for each row only few records would have values while the remaining would be empty which will create other problems (Sparsity), this sparsity will cause biased estimation. Additionally, the dataset is Heterogeneous since it contains records of student from different time frames, departments and nature. Consequently, we will propose several scenarios to split the dataset into different case studies, each has its own configuration of indicator courses. For each one of these cases, we will learn and tune a Linear Regression model and discuss its meaning. Furthermore, each one of these case studies will be analyzed and evaluated separately.

6.1 Case Study A: Predicting Algorithm Analysis (First Year Course)

In this configuration, the goal is to predict student's grade in Algorithm Analysis using a dataset comprised of 87 records with students marks in Programming Fundamentals, Data Structure and Discrete Mathematics. All these courses are taken in the first year. Additionally, other non-academic features have been included, such as, Socioeconomic Index, High school and Gender. Applying Linear Regression (with L1 Norm) to this dataset gave the following Model:

$$\begin{aligned} AlgorithmAnalysis \;&=\; 0.1046 * DataStructures + \\ &\quad 0.0948 * DiscreteMathematics \; + 5.6301 \end{aligned}$$

As we can see, even though we do not have enough records to draw general conclusions, the results show positive correlation between the score in Algorithm Analysis and Data Structures and Discrete Mathematics. On the contrary, Programming Fundamentals was excluded by the features selection algorithm, which indicates a low impact on the final grade.

6.2 Case Study B: Statistics I

For this case, we predict the grade of Statistics as it represents an important course that is common among different departments. To that end, several courses have been chosen as predictors for this case study namely, Multivariate Calculus, Linear Algebra, Differential Calculus and Integral Calculus. The total number of records for this case is 654 records. The best results that could be achieved using features selections is the following:

Statistics = (0.17 * IntegCalculus) + (0.19 * LinearAlgebra)
- (0.84 * CountStatistics) - (0.21 * CountDiffCalculus)
- (0.15 * MultivariableCalculus) + 5.58

It could be seen from the equation that student's mark in Linear Algebra highly affects the their mark in Statistics. That could be an important information to warn those students with low marks in Linear algebra to put more effort on Statistics. However, in this program, Linear Algebra is not a direct prerequisite for Statistics. That could be taken into account by program coordinators. Moreover, from the negative coefficient for the "Count" features, we can induce that repeaters (in Statistics, and Differential Calculus) tend to perform badly in Statistics. It is worth noticing that by adding the course count, and courses' last achieved mark and difficulty estimators as features, we reduced the RMSE (Root Mean Squared Error) by 23%, and the MAE (Mean Absolute Error) by 16%. The formulas for MAE and RMSE are briefly explained in Section 7.1. In the next two case studies, we will investigate the same problem of predicting Statistics grade but using different courses and with additional features.

6.3 Case Study C: Statistics II (Repeaters)

In this case study, we look at the problem of predicting final grade in Statistics in terms of the grades of: Calculus I, Calculus II and Linear Algebra. However, we will distinguish between the students who have passed Statistics the first time (Non Repeaters) and the people who have failed at least once to see if this will affect the model. Additionally, we are considering new feature for repeaters, which is the previous mark taken in each course, including Statistics. The total number of records for this case is 566 records. [we were predicting the mean]

$$\begin{gathered} Statistics \;=\; 0.10 * LinearAlgebra \;+\; 0.12 * CalculusII \\ 0.06 * PrevStatistics \;-\; 0.08 * PrevCalculusII \\ -\, 0.10 * CountCalculusI \;-\; 0.14 * CountLinearAlgebra \\ -\, 0.20 * CountCalculusII \;+\; 6.06 \end{gathered}$$

The results in this case agrees with the the previous case results where Linear Algebra still plays an important role to specify Statistics mark. Moreover, Calculus II turned to be an important indicator too. Surprisingly, the equation suggests that repeating Statistics has positive effect on the the final statistic mark, on the contrary, repeating one of Calculus I, Linear Algebra or Calculus II have an opposite effect.

6.4 Case Study D: Statistics II (Non Repeaters)

For non repeaters, we will be predicting the final grade in Statistics in terms of the same configuration of the repeaters case, however, without the previous mark in Statistics.

Statistics = (0.21 * CalculusI) + (0.18 * LinearAlgebra)
+ (0.1793 * CalculusII) + (0.1319 * CountCalclusI) -
118.59

We can see that we are getting comparable indicator-importance and results for Repeaters and Non-Repeaters. Which encourage us to try merging them in one unified model.

6.5 Case Study E: Statistics II (Both)

As we have seen in previous two sections, we could predict the grade of Statistics with good results by distinguishing between repeaters and non-repeaters. In this section, we will try to build a unified one model for both cases. The challenge is to unify the features set. Since for non-repeaters we do not have a value for previous mark in Statistics, we deal with the problem of "Missing Values", and for that, we use Average Imputation. The considered average is the mean of statistics mark between all non repeaters. Furthermore, a new attribute were added "IS_REPEATER" that will naturally have the value of zero for non-repeaters to help the Regressor distinguish between them implicitly.

Statistics = (0.09 * CalculusI) + (0.2 * LinearAlgebra)
+ (0.15 * CalculusII) - (0.15 * CountCalclusII) + (0.06
* PrevCalculusI) - (0.15 * CountLinearAlgebra) + (0.04
* PrevStatistics) + (0.4 * IS_REPEATER) + 3.2

Adding the difficulty estimators mentioned in section 4, gave better results:

Statistics = (0.09 * CalculusI) + (0.2 * LinearAlgebra)
+ (0.15 * CalculusII) - (0.15 * CountCalclusII) + (0.07 *
PrevCalculusI) - (0.15 * CountLinearAlgebra) + (0.07
* PrevStatistics) + (0.4 * IS_REPEATER) - (0.01 *
LOAD) - (0.07 * Beta) + 3.2

We can see that approximately, this model captures the important behavior from both models. For instance, the grade of statistics still correlates positively with Linear Algebra, Calculus I and Calculus II. While being a repeater for Linear Algebra or Calculus II correlates negatively. Moreover, repeating Statistics still have positive impact on your final grade. Consequently, the results is reassuring, especially, since we have around 1503 records in the dataset and we can trust the results more.

6.6 Case Study G: Analysis of Electrical Networks

Finally, a course that is common between different domains, "Electrical Networks" is chosen to check if we will still get acceptable results regardless of the differences between students belonging to different careers. To that end, several indicator courses have been suggested, namely: Calculus I, Calculus II, Calculus III, Linear Algebra, Physics I and Physics II. As we can see also the base courses cover mainly Mathematics and Physics. The Linear Regression final model was:

ElectricNetworks = (0.09 * CalculusI) + (0.2 * LinearAlgebra)
+ (0.09 * CalculusII) + (0.09 * CalculusIII) + (0.03 *
PrevPhysicsI) - (0.2 * CountLinearAlgebra) + (0.16 *
PrevElectricNet) + (0.7 * IS_REPEATER) + (0.06 *

PrevCalculusIII) + (0.08 * PrevCalculusI) + 2.1274

7 Results

7.1 Evaluation

There are many metrics to help in evaluating machine learning models. Since we are interested in predicting the final grade of student as numeric variable, Regression evaluation metrics are used. Specifically, the most used metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Root Absolute Error (RAE) and Root Relative Squared Error (RRSE).

$$MAE = \frac{\sum_{i=1}^{n} |p_i - a_i|}{n} \tag{6}$$

$$\begin{aligned} RMSE &= \sqrt{\tfrac{\sum_{i=1}^{n} (p_i - a_i)^2}{n}} \\ a &= \text{ actual target} \\ p &= \text{ predicted target} \end{aligned} \tag{7}$$

$$RAE = \frac{\sum_{i=1}^{n} |p_i - a_i|}{\sum_{i=1}^{n} |\overline{a} - a_i|} \tag{8}$$

$$RRSE = \sqrt{\frac{\sum_{i=1}^{n} (p_i - a_i)^2}{\sum_{i=1}^{n} (\overline{a} - a_i)^2}} \tag{9}$$

The evaluations is done using repeated 10-fold cross validation method [17] where dataset is divided into 10 equal-sized groups, each time, one group is considered as a testing set while the remaining are used for training a model. This process is repeated 10 times and a separate model is trained and evaluated each time. The final evaluation would be the average of evaluation for the 10 models.

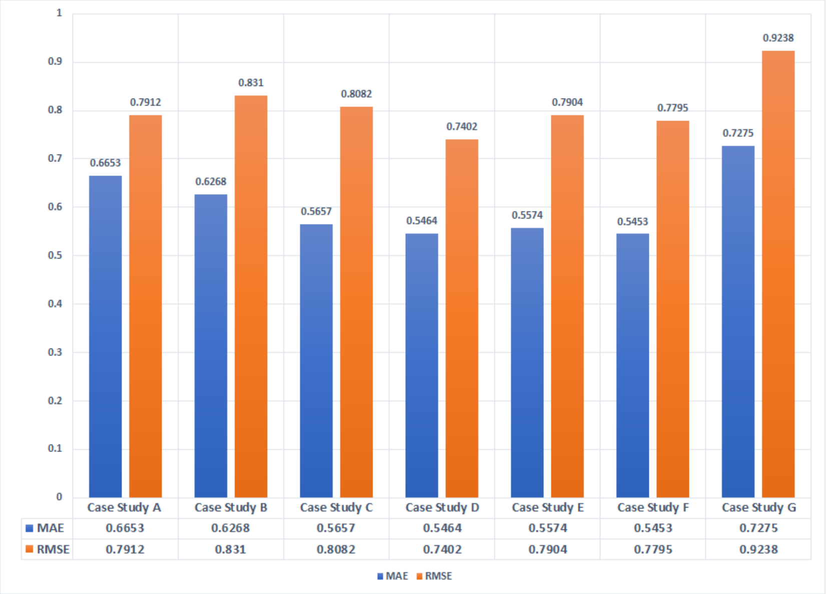
7.2 Case Studies Results

This section presents the different results achieved in previous section. Table 7 shows that, despite the differences between case studies, we have relatively good error rate for all cases. Moreover, it is worth noticing in Figure 3 that the variance in results is low across different cases which means that the same behavior persist between different lectures and we can trust the results more. Consequently, this is encouraging to have an online working system. Finally, we can explain that the results of Case A is different due to the small dataset size and the fact the it is a case of predicting a first year course. Therefore, probably previous courses will have low prediction impact. In other words, it would be difficult to predict the grade in Algorithm Analysis based only on Programming Fundamentals, Data Structure and Discrete Mathematics. On the other hands, for the rest, we can see that the results of the best model of Case B agree with the results of Cases C, D, E and F. For case G, the small deviation could be explained that this course is taken by large number of students from different departments and majors (Heterogeneity), yet we still have acceptable error rates.

Case Study	Courses	Dataset Size	MAE	RMSE	Notes
A - Algorithms Analysis	Programming Fundamentals, Data Structure and Discrete Mathematics	87	0.6653	0.7912	
B - Statistics I	Multivariable Calculus, Linear Algebra, Differential Calculus and Integral Calculus	654	0.8364	1.2138	
B - Statistics I	Multivariable Calculus, Linear Algebra, Differential Calculus and Integral Calculus	654	0.6268	0.831	Feature Selection, adding (frequency, course difficulty, averaging previous courses)
C - Statistics II (Non-Repeaters)	Calculus I, Calculus II and Linear Algebra	938	0.5657	0.8082	
D - Statistics II (Repeaters)	Calculus I, Calculus II and Linear Algebra	566	0.5464	0.7402	
E - Statistics II (All)	Calculus I, Calculus II and Linear Algebra	1503	0.5574	0.7904	
F - Statistics II (All)	Calculus I, Calculus II and Linear Algebra	1503	0.5453	0.7795	After including Beta and Course Load
G - Analysis of Electrical Networks	Calculus I, Calculus II, Calculus III, Linear Algebra, Physics I and Physics II	974	0.7275	0.9238	

Table 7: Results

Figure 3: Evaluation of case studies (MAE, RMSE)

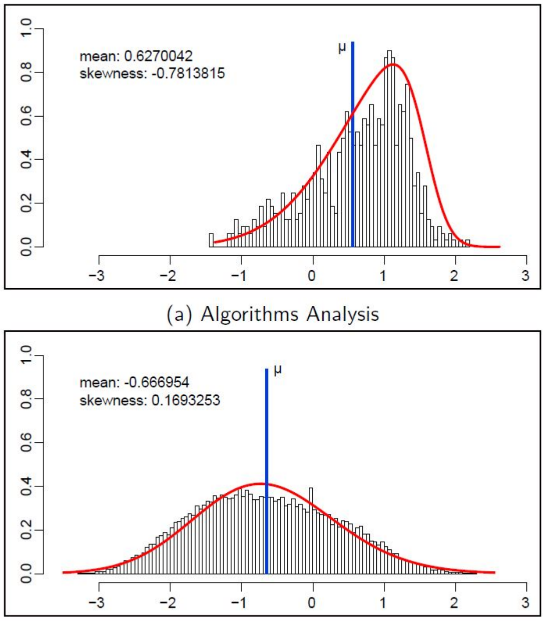


8 Future Work

For future work, the method could be further improved by employing new features, such as : engineer more fine-grained features to get more homogeneous indicators. Furthermore, other new features worth explore as difficulty estimator (Skewness). Additionally, applying dimensionality techniques on the whole dataset could be useful to build the unified model.

8.1 Include New Features

It is reasonable to enhance the same feature-set be more fine-grained. In other words, we calculate the same features per Department, Time Frame and Professor. Consequently, we will get more accurate indicators and hence, more homogeneous results. For example, since the dataset represents students mark from 1978 onward to 2012, it is logical that each time frame had different trend. Same goes for professor. Moreover, for difficulty estimator, the skewness of the statistical distribution for the differences between students' GPA and the grade in specific course grade at that time will reflect how much in general this course shifted the students' grade positively or negatively. As depicted in the Figure 4, the course Algorithm Analysis is negatively skewed, which means that the majority of values are on the right (the GPA is large then the mark), and hence it



(a) Algorithms Analysis

(b) Oral and Written Communication Techniques

Figure 4: Skewness [2]

is a difficult course. On the contrary, Oral and Written Communications course is positively skewed (Easy Course)

8.2 Dimensionality Reduction (Towards a Unified Model)

As mentioned before in Section 6, training a unified model is challenging. However, applying dimensionality reduction techniques, such as Principal Component Analysis (PCA), could help in solving this problem. Mainly, having a large number of redundant features. Applying PCA for all courses will result in projecting the data into lower dimensional space where each component might represent courses that have commonality. [2] explored that, the dataset was reduced into 10 main factors and each factor has an interpretation in terms of course (Mathematical courses Factor). Furthermore, being able to represent each student in terms of those factors will give an idea about student's general performance in terms of more Coarse-grained domains and hence, this could be used as general features regardless of the mark in each course. On the other hand, we could extract those features also by employing domain knowledge to cluster courses according to well-defined criteria (Mathematical courses, Humanity Courses, etc), the department offering the course could be used for this as well. As an example, according to Figure 5 from [2], it can be seen that each each course is contributed by one factor or more.

9 Conclusion

One concern of EDM is to provide a decision support system to help students making fruitful decisions to improve the overall performance and avoid potential academic problems. Specifically, being able to predict student's grade in future course will give the student, academic institution the advantage of taking proactive measures. In this report we proposed a method to predict future performance using previous academic data. The main method depicted in Figure 2, starts by preparing the data for preprocessing. In preprocessing, several procedures were applied to clean the data and make it in standard form to be analyzed. Afterwards, the data is further enhanced by adding new meaninful features inspired by the domain knowledge. Finally, Regularized Linear Regression algorithm is applied on the data and evaluated using 10-folds cross validation. To better trust the results and have more homogeneous data, the dataset was split into different case studies, each one for predicting specific target course depending on a subset of perquisites courses. The results shows consistent behavior across all data splits, which is encouraging and re-assuring that we could depend on the suggested features to predict accurately the final grade in future course (RMSE less than 1 in all experiments). Moreover, we found that the introduced features suggested in this work helped in reducing the RMSE by 23%, and the MAE by 16%. Finally, this comes with the benefits of having interpretable results by investigating the final model coefficients for each course.

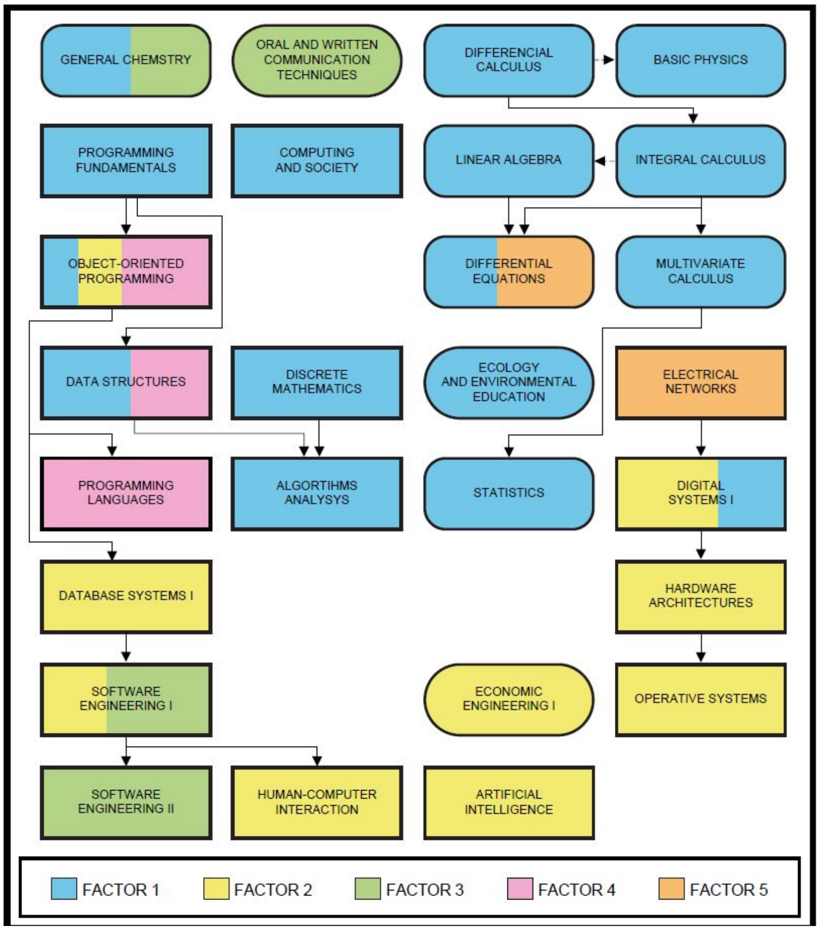


Figure 5: Factor Analysis

References

[1] Omiros Iatrellis, Achilles Kameas, and Panos Fitsilis. Academic advising systems: A systematic literature review of empirical evidence. *Education Sciences*, 7(4):90, 2017.

[2] Gonzalo Mendez, Xavier Ochoa, Katherine Chiluiza, and Bram de Wever. Curricular design analysis: a data-driven perspective. *Journal of Learning Analytics*, 1(3):84–119, 2014.

[3] Jonathan P Caulkins, Patrick D Larkey, and Jifa Wei. Adjusting gpa to reflect course difficulty. 1996.

[4] Pedro Strecht, Luís Cruz, Carlos Soares, João Mendes-Moreira, and Rui Abreu. A comparative study of classification and regression algorithms for modelling students' academic performance. *International Educational Data Mining Society*, 2015.

[5] Iti Burman and Subhranil Som. Predicting students academic performance using support vector machine. In *2019 Amity International Conference on Artificial Intelligence (AICAI)*, pages 756–759. IEEE, 2019.

[6] Mehdi Mohammadi, Mursal Dawodi, Wada Tomohisa, and Nadira Ahmadi. Comparative study of supervised learning algorithms for student performance prediction. In *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 124–127. IEEE, 2019.

[7] Nupur Chauhan, Kimaya Shah, Divya Karn, and Jignasha Dalal. Prediction of student's performance using machine learning. *Available at SSRN 3370802*, 2019.

[8] Pei-Chann Chang, Cheng-Hui Lin, and Meng-Hui Chen. A hybrid course recommendation system by integrating collaborative filtering and artificial immune systems. *Algorithms*, 9(3):47, 2016.

[9] Qingyu Wu. *Predicting Academic Performance via Machine Learning Methods*. PhD thesis, 2017.

[10] Ahmet Tekin. Early prediction of students' grade point averages at graduation: A data mining approach. *Eurasian Journal of Educational Research*, 54:207–226, 2014.

[11] Zafar Iqbal, Junaid Qadir, Adnan Noor Mian, and Faisal Kamiran. Machine learning based student grade prediction: A case study. *arXiv preprint arXiv:1708.08744*, 2017.

[12] Mong Li Lee, Hongjun Lu, Tok Wang Ling, and Yee Teng Ko. Cleansing data for mining and warehousing. In *International Conference on Database and Expert Systems Applications*, pages 751–760. Springer, 1999.

[13] Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and Lofti A Zadeh. *Feature extraction: foundations and applications*, volume 207. Springer, 2008.

[14] Michael Pecht. Prognostics and health management of electronics. *Encyclopedia of Structural Health Monitoring*, 2009.

[15] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[16] Francisco Azuaje. Witten ih, frank e: Data mining: Practical machine learning tools and techniques 2nd edition. *BioMedical Engineering OnLine*, 5(1):51, Sep 2006.

[17] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.