ANALYZING UNIVERSITY STUDENT DATA TO DETERMINE THE KEY FACTORS OF COURSE FAILURE

Photiou, Antonis, University of Cyprus, Nicosia, Cyprus, aphot01@ucy.ac.cy

Abstract

The field of data analysis has seen an incredible blossom in recent years. Organizations around the world increase their reliance on analytical approaches in order to boost their productivity and improve their profitability. Similarly, academic institutions try to exploit their data in a way to get meaningful insights into their institution, which will help them increase the quality of education, reputation, and economic status. One way to improve the quality of education is to find the key determinants of student failure and be able to support them. This paper employs machine learning to model student data with the task of finding patterns of student performance within the data. The analysis has proven that student performance could be predicted early through the course spell using various static and dynamic features. The key predictor of student performance was the interaction and the engagement of students with the virtual learning environment.

Keywords: Machine Learning, Ensemble Methods, Classification, Learning Analytics, Academic Analytics, Educational Data Mining.
1 Introduction

The field of data science has seen an incredible blossom in recent years. Organizations around the world increase their reliance on analytical approaches to boost their productivity and improve their profitability. The flexible nature of data science approaches promotes its extent to every domain including education. Although data science has proven itself in many different realms, within academic institutions is yet in its early stages. Nowadays, educational institutions, produce unmanageable amounts of data which in the majority of the institutions stay unexploited, neglecting the significant guidance of in-house knowledge.

Academic institutions are making substantial investments to improve their quality of education, enhance their brand and increase their research output. However, the investments are mainly based on competition and do not reflect the needs of the students and academics. The main objective of higher education institutions is to provide quality education to their students. One way to achieve the highest level of quality in higher education is to provide the students with meaningful knowledge about their studies. This knowledge is hidden in the institutions’ databases and is extractable using data analytics and data mining techniques. Higher education institutions with the use of those techniques are able to evaluate in-campus student’s performance, online student performance, called learning analytics, and provide help for the ongoing problem of student retention.

Student dropout is a major concern in the education and policy-making communities[8]. About 40% of undergraduate students do not complete their degree within six years, with universities losing great amounts of revenue each year. The abundance of data and the rise of new quantitative and statistical techniques enables higher education institutions to measure the performance of individual students. Predicting a student’s performance based on in-class assessments, homework assignments and more importantly through students Learning Management System (LMS) activity, can potentially provide the needed early intervention for students that are at risk of failing a course.

Based on this background, the aim of this paper is to exploit machine learning to find out whether the academic performance could be predicted early through the course duration, enabling institutions to help students with the problem of course failure. Additionally, an important task is to identify which factors provide predicting capability, making the models more interpretable for practitioners.

Previous studies investigated the task of predicting student performance employing machine learning [1,7,9,13] and the importance of LMS data for achieving the task [2,3,4,5,6]. Furthermore, several studies explored the possibility of developing early warning systems for at-risk students [10,11,12]. Marbouti, Diefes-Dux, and Madhavan (2016) confirmed the effectiveness of machine learning to predict student performance using in-semester data without LMS activity, while Hu, Lo, and Shih (2014) and Macfadyen and Dawson (2010) showed the potential of LMS in early prediction of academic performance. However, many of the studies did not use LMS activity which is crucial in today’s educational settings. Furthermore, studies related to early prediction of academic performance have shown the potential of LMS while using limited and unbalanced data related to a single course presentation, potentially compromising the generalizability of the results.

In this study, the aim is to take advantage of different aspects of learners activity in LMS. Nowadays, higher education institutions increasingly rely on LMSs to manage the learning process of their students. LMSs are central for the communication between students and educators, for the smooth administration of classes and also provide log activity data which can potentially lead to early conclusions on students’ future academic performance. On-time feedback about the learners’ activity is essential when trying to support in-danger students during the term. Therefore, the usage of easily accessible and timely data is vital for an effective system which can provide meaningful information to students and academics.

In this paper I compare machine learning algorithms, to determine which models perform better on given data. Also, I do feature engineering on LMS data in order to find the features with the highest predicting ability. The goal is to find the best model to predict in-danger students and determine which factors provide the most significant knowledge. Motivated by the development of a machine learning model which utilizes data widely available for every course, and not course-specific, to test the generalizability.
of the task and also to welcome automation of the process. Therefore, this paper will focus on answering the following questions:

- Could machine learning algorithms be used to predict student performance early for an ongoing course?
- Which LMS features contain essential information in predicting academic performance?
- Which applied method provides the most suitable solution for the given task?

2 Literature Review

There is an increasing research activity in recent years regarding the usage of data science in education. Analytical approaches are employed to address some of the critical challenges of education, including student performance, admission decisions, and for the analysis of course and instructor evaluations.

2.1 Predicting Academic Performance

In recent years, many studies have explored the possibility of predicting academic performance given a set of different factors such as demographical data, family data, and pre-university performance. While earlier studies have focused on next-term predictions and the usage of historical data, more recent studies have utilized LMS activity features to their models. Sweeney, Rangwala, Lester, and Johri (2016) used historical student data in order to compare machine learning algorithms related to next-term performance predictions. Amrieh, Hamtini, and Aljarah (2016) and, Dutt and Ismail (2018) employed ensemble classifiers using a combination of historical and personal data with behavioral aspects of student activity. Daud, et al. (2017) exploited personal data and family-related data to achieve the task. Despite the predicting ability of the aforementioned models, they have several limitations. The models are used for next-term predictions or to provide correlations between finished courses and student data. While they provide good foundations for further development, they do not offer easily interpretable results for future usage.

2.2 LMS and Student Performance

The abundance of data available from LMSs enabled practitioners to move from historical and demographical data to using metrics of student online activity. Many studies have shown that LMS data are good predictors of academic success. Damianov, et al. (2009) found that time spent online correlates with the final grade, while Baugher, Varanelli, and Weisboard (2003) and Biktimirov and Klassen (2008) described the consistency of access as the most important indicator. Amrieh, Hamtini, and Aljarah (2016) and, Dutt and Ismail (2018) found that visited resources and views of the announcements are the best predictors. On the other side, Conijn, Snijders, Kleingeld and Matzat (2017) support that student characteristics are more important than LMS activity data and, Conijn, Snijders, Kleingeld and Matzat (2016) endorse that LMS data provide little additional value to predictive models. Overall, we see that studies do not agree regarding the importance of LMS data for student performance prediction. Due to the diversity of the courses, the universities and the methods that were used, it is difficult to compare results between these studies. Although, the common limitation of the aforementioned studies is the time of prediction, which is made after the end of the course which is not suitable for early intervention.

2.3 Early Warning systems

Having evidence from previous studies about the feasibility of academic success prediction, many researchers have experimented with early alert systems. Early alert systems could be crucial in the future in assisting students to avoid course failure and to provide academics with a platform to support those students. Popular examples of early alert systems include E2Coach and Purdue Signals. Purdue University has developed the signals learning analytics program which informs students in real-time about their
status in a course based on various data, including LMS interaction [14]. The University of Michigan developed the E2Coach digital tool which supports students in introductory STEM classes using an open-source software used by public health professionals which offers targeted communication to people who need help, by sharing messages from peers. Furthermore, Marbouti, Diefes-Dux, and Madhavan (2016) examined different prediction models and found that ensemble classifiers and Naïve Bayes Classifier are the best algorithms for identifying at-risk students using in-semester data without LMS activity. Macfadyen and Dawson (2010) used LMS data and logistic regression to classify students to ‘at-risk’ and ‘not at-risk’. Hu, Lo, and Shih (2014) reported great results, predicting the course outcome on the fourth week of the course in progress. The aforementioned studies have reported promising results, using different prediction models and data. However, they share a common limitation. They operated on small datasets consisting of a single course with class imbalance, compromising the generalization of their findings.

3 Methodology

The aim of this paper is to predict student performance in the higher education context. Static information about students is combined with in-class performance data to create a model that can predict whether a student will be successful in a particular course. The secondary aim of this task is to use prediction models to predict student performance halfway through the course. Being able to predict early enough the performance of students, helps to support students who are in danger of failing the class and give them the opportunity to get back in track. Predictions will enable the observation of patterns which result to student failure allowing institutions to implement intervention strategies.

The data are in the tabular format where each row contains information about a specific student for a single course. Static data include demographic characteristics, personal information, and previous education information. Dynamic information include data generated during the course such as assignment grades, exam grades and LMS interaction. The data were filtered accordingly in order to contain dynamic information during the first half of the course. Specific information about the utilized data will be covered in a later section.

The target variable is categorical and contains two levels, 'Pass' and 'Fail'; thus, it represents a classification task which aims to find information that better divide students into the two classes. Furthermore, the algorithms used belong to the supervised learning category of machine learning as the target value, the student grade, is known.

3.1 Data

The algorithms were trained and tested using the Open University Learning Analytics database. The database contains anonymized data about students, courses, and interactions with the Virtual Learning Environment (VLE). There are seven selected courses with presentations in four consecutive semesters. The database accommodates seven tables connected using unique identifiers. The constructed dataset contains a total of 15 features, gathered from the different tables.

The first two features contain information about the course. Features 3 to 11 are unique, static information about each student. Features 12 to 14 are the dynamic data of the student during the first half of the course. Res is the student’s final result in the module presentation and the target variable of this analysis.

The features \textit{vleAvBef}, \textit{vleSumNorm} and \textit{avAssNorm} contain summarized information about the student’s performance, collected from the tables \textit{studentAssessment} and \textit{studentVle}.

- \textit{vleAvBef}: The average interaction of a student with the module’s learning materials. Interaction is quantified by the number of clicks a student had in a visit to the Virtual Learning Environment (VLE). The information was gathered from the \textit{studentVle} table and summarized into one value for each student using mean.
- \textit{vleSumNorm}: The total interaction each student had with the VLE, measured in clicks.
• **avAssNorm**: The cumulative grade that a student achieved for the first half of the course's duration. The scores were gathered from the `studentAssessment` table and calculated the value using the weight of each assessment from the `courses` table.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. code_module</td>
<td>Categorical</td>
</tr>
<tr>
<td>2. code_presentation</td>
<td>Categorical</td>
</tr>
<tr>
<td>3. id_student</td>
<td>Categorical</td>
</tr>
<tr>
<td>4. region</td>
<td>Categorical</td>
</tr>
<tr>
<td>5. gender</td>
<td>Categorical</td>
</tr>
<tr>
<td>6. highest_education</td>
<td>Categorical</td>
</tr>
<tr>
<td>7. imd_band</td>
<td>Categorical</td>
</tr>
<tr>
<td>8. age_band</td>
<td>Categorical</td>
</tr>
<tr>
<td>9. num_of_prev_attempts</td>
<td>Numerical</td>
</tr>
<tr>
<td>10. studied_credits</td>
<td>Numerical</td>
</tr>
<tr>
<td>11. disability</td>
<td>Categorical</td>
</tr>
<tr>
<td>12. vleAvBef</td>
<td>Numerical</td>
</tr>
<tr>
<td>13. vleSumNorm</td>
<td>Numerical</td>
</tr>
<tr>
<td>14. avAssNorm</td>
<td>Numerical</td>
</tr>
<tr>
<td>15. res</td>
<td>Categorical</td>
</tr>
</tbody>
</table>

**Figure 1. Database schema(left). The features of the constructed dataset(right).**

**Data Pre-processing**

Information from the `courses` table about the length of each course was used to filter the data and calculate the mid-course performance data. Students that withdraw the course at any point were filtered out. The `res` variable initially had three levels, 'Distinction,' 'Pass,' 'Fail' but during the data processing, 'Distinction' was transformed to 'Pass.' This transformation helps to make the models simpler and better serve the initial aim, to focus on the students with a higher probability of failing the course. The variables `vleSumNorm` and `avAssNorm`, were normalized to the range of 0 to 1 using Min-Max Scaling equation shown below.

\[ x'_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \]

Missing values were treated using regression imputation. Regression imputation estimates the missing data based on models created using the other variables. Entries with more than two missing values were dropped in order to maintain the quality of the data.

**3.2 Approach**

In figure 2, the pipeline of the machine learning process is visualized. Stage 1 involves the data aggregation from the different tables of the database, the data cleaning and imputation. In stage 2, different algorithms were tested in order to find the ones that performed better on the dataset. The algorithms Random Forest Classifier, Gradient Boosted Machine, Adaptive Boost Classifier had the best performance on the dataset and were used in the rest of the analysis. A brief description of each algorithm is presented in a later section. The algorithms were trained on data from the first three semesters and tested on the fourth semester. In the fourth semester, all 7 courses were active. Stage 3 involves the repetitive optimization and evaluation of the different algorithms in order to find the optimal subset of parameters for each one. In Stage 4 the creation of additional features aims to better describe the data in order to
improve the results. In Stage 5, the repetitive optimization and evaluation of the algorithms was employed on the refined dataset. In the final stage, the different algorithms were compared using different metrics.

Figure 2. Machine Learning process.

3.3 Algorithms

The algorithms that were used for this analysis belong to the category of ensemble learning. Ensemble methods combine several models to provide better results. They are used to decrease variance and model bias or to improve predictions. They can be divided into boosting methods, where models are being generated sequentially weighting previously wrong predictions, and bagging methods, where independent models are created averaging their results. Furthermore, stacking algorithms which combine two or more other models belong to the ensemble learning category.

Bagging algorithms create random samples of the training data and then build a model on each sample. The results are taken using the average for regression problems or majority voting for classification problems. Because each model is exposed to a different set of data, the final output decreases variance and prevents overfitting.

Random forest is a bagging algorithm which creates a decision tree from each sample with replacement. Additionally, random forest randomly chooses between the features to be used in each tree, reducing the correlation between the trees. The decision tree algorithm builds a tree structure classification or regression model. The tree consists of decision nodes and leaf nodes. A decision node has two or more branches based on the training data while a leaf node is the result of the algorithm. A tree is implemented by splitting the data based on a feature value recursively. The algorithm first breaks the data in order to create homogenous branches. The tree is constructed top-down, each time selecting the split attribute which gives the highest gain. The splitting stops when the new split does not offer any additional predicting power to the model.

Boosting algorithms train a sequence of weak models based on the results of a previous model. The algorithms increase the weight of an incorrectly predicted instance of a prior model. The predictions of each model are combined with majority voting or weighted sum. The influence of each model in the final prediction is based on its performance.

Adaptive Boost Classifier, a boosting algorithm, starts by using weighting coefficients that are equal for every observation. In every iteration, the coefficients are increased for incorrectly classified predictions and decreased for the correct ones.

Gradient Boosting Classifier, another boosting algorithm, sequentially builds a tree, aiming to minimize the loss function in each iteration.

3.4 Evaluation

There are many classification metrics, and one should select the most appropriate to evaluate a classifier. Evaluation metrics play a critical role in finding the proper algorithm for a given problem. Those metrics are employed during the training of a model, to optimize the classifier best, and during the testing, to
evaluate the effectiveness of the algorithm for new data. For binary classification problems, the effectiveness of a model could be presented with a confusion matrix.

<table>
<thead>
<tr>
<th>Actual ‘Fail’</th>
<th>Predicted ‘Fail’</th>
<th>Predicted ‘Pass’</th>
</tr>
</thead>
<tbody>
<tr>
<td>True Positive (tp)</td>
<td>False Negative (fn)</td>
<td></td>
</tr>
<tr>
<td>Actual ‘Pass’</td>
<td>False Positive (fp)</td>
<td>True Negative (tn)</td>
</tr>
</tbody>
</table>

*Table 1. Confusion Matrix.*

True positive and true negative values, represent the number of instances which the model has classified correctly, while false positive and false negative denote the number of incorrectly classified ones. In this case, the positive class is regarded as the ‘Fail’ class due to the significant goal of predicting the students who are in danger of failing a course. The following evaluation methods were used.

\[
\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}
\]

\[
\text{Precision} = \frac{tp}{tp + fp}, \quad \text{Recall} = \frac{tp}{tp + fn}
\]

\[
F1\text{Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

4 Results

4.1 Method Comparison

The results of every model are summarized in the following table along with their metric scores.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient Boosted Machine (GBM)</td>
<td>0.8149</td>
<td>0.7585</td>
<td>0.5851</td>
<td>0.6606</td>
</tr>
<tr>
<td>Adaptive Boost Classifier (ADA)</td>
<td>0.8099</td>
<td>0.7460</td>
<td>0.5925</td>
<td>0.6605</td>
</tr>
<tr>
<td>Random Forest (RF)</td>
<td>0.8117</td>
<td>0.7654</td>
<td>0.5668</td>
<td>0.6513</td>
</tr>
<tr>
<td>GBM + Feature Engineering</td>
<td>0.8942</td>
<td>0.8798</td>
<td>0.7492</td>
<td>0.8092</td>
</tr>
<tr>
<td>ADA + Feature Engineering</td>
<td>0.8853</td>
<td>0.9002</td>
<td>0.6941</td>
<td>0.7838</td>
</tr>
<tr>
<td>RF + Feature Engineering</td>
<td>0.8914</td>
<td>0.8979</td>
<td>0.7195</td>
<td>0.7989</td>
</tr>
<tr>
<td>Stacking Classifier</td>
<td>0.8840</td>
<td>0.8170</td>
<td>0.7898</td>
<td>0.8032</td>
</tr>
</tbody>
</table>

*Table 2. Evaluation metrics for every method used.*

The results among the tested algorithms do not deviate substantially from one another when it comes to F1-Score. The differences between the algorithms with the higher and lower F1-Score was 2.54% and 0.93% for the methods with and without feature engineering respectively. The reason for this might be the similar structure of the algorithms which relies on the creation of multiple models, forcing the predictions to converge. To answer the first analysis question; the method with the best results is the Gradient Boosting Machine (GBM) on engineered data with average F1-Score of 80.33% based on 10-fold cross-validation testing. Regarding the other metrics, the Stacking Classifier achieved the highest Recall score with significant deviation from the other algorithms but with the expense of a lower Precision score. In terms of Precision, the Adaptive Boost Classifier achieved the highest score.
An additional subject of interest is the comparison between the results of the initial dataset and the engineered dataset. On average, the results on engineered data improve the F1-Score of the models by 13.95% in relation to the first dataset. Initially, describing each observation with additional metrics of VLE interaction, boosted the predictability of the models and helped to increase the average accuracy of the learners by 7.8%. Furthermore, excluding the features which resulted in mispredictions further bolstered the results. More importantly, feature engineering amplified the Recall scores for 13.9% on average. The algorithms were able to boost the correct classification of failing students by 14%.

4.2 Feature Engineering

During the process of feature engineering, an important task was feature creation. Guided back to the database, new features describing the student’s online behavior were created. Among the new features created, three have given a significant boost to the results. The feature importances for the three algorithms are presented in figure 3.

- **times**: The total number of unique days a student used the VLE.
- **distance**: The difference between the first and the last day a student logged-in the VLE.
- **max_dist**: The maximum gap between two consecutive log-ins to the VLE.

![Figure 3. Feature Importance for Gradient Boosting Machine (left), AdaBoost (centre) and Random Forest (right).](image)

4.3 Findings

Figure 3 contains the feature importance for the three algorithms. Feature importance indicates how valuable the feature was to the model. The higher the importance of a feature the higher its ability to homogenously divide the data points into ‘Pass’ or ‘Fail’.

4.3.1 Virtual Learning Environment

The five features, distance, max_dist, times, vleSumNorm and vleAvBef, which describe the student interaction with the VLE, rank among the top six features for every algorithm. The results displayed above indicate the great predicting ability of the VLE interaction data. A model that only contains the five features achieved an F1-Score of 0.76 which even surpasses the performance of the initial models.

Figure 4 shows the density of the distance for the two classes. The students who failed the course have a uniform distribution while the students who were successful present a negative skew where 90% of the observations are valued above 110. A perfect distance score means that a student has visited the VLE the first and the last day of the analysis timeline among others. Distance ranks first for Adaptive Boost Classifier and Random Forest algorithms which highlights the importance of accessing the learning materials throughout the length of the course as it increases the probability of being successful. However, distance may account for some misclassifications as a student with limited VLE access could get a high distance metric. Thus, it is better to be evaluated along with other measures.
The profound importance of demographic data in all three models is a hint that there is no or little deviation in the course result based on those characteristics. Using the demographic features in the Adaptive Boost Classifier and Random Forest not only increased the complexity of the models but more...
importantly penalized the overall predicting power of the algorithms. On the other side, in the results of GBM, there is an improvement with the use of these data of 0.5% in terms of F1-Score.

The most essential demographic feature has come out to be the region of the student. In figure 7 the final result proportion is shown for each of the thirteen regions. The deviation between the regions in terms of passing percentage reasonably justifies the importance of the feature.

![Pass/Fail % per UK Region](image)

*Figure 7. Final result by region.*

4.3.3 Module

The module feature seems to be necessary for the models as it appears relatively high in the feature importance ranking. Figure 8 highlights the proportion of students who have failed to pass each course. Extending the work of this project to further improve the classification quality, different prediction models could be made on each course. This approach would probably result in better classification outcome because each course has different learning materials, grading criteria and online presence. As it appears in figure 8, there are deviations between the modules that could justify further development. Regarding this dataset, a similar approach did not improve the results. The reason may be that after dividing the dataset, the data for each course were insufficient to build an effective model.

![Fail % by module](image)

*Figure 8. Passing/Failing % for each module (left). Average values for times, avAssNorm, vleSumNorm, max_dist for each module (right).*
4.4 Misclassifications

The task of identifying and addressing the reasons behind a model’s misclassifications is crucial in machine learning. In this analysis, the instances which were misclassified as ‘Pass’ require careful attention as they operate against the aims of the analysis. Using data from the best classifier, GBM, the misclassifications were visualized in order to find what caused them. In figure 9, the data of the misclassified students are plotted against their assigned probability of passing the course. The visualized features, $avAssNorm$, $distance$, and $vleSumNorm$, have a positive correlation with the assigned probability which denotes that as they increase, the likelihood of passing the course raises. The vertical lines represent the average values for ‘Pass’ and ‘Fail’ instances as well as the average of the misclassified instances. In general, the students who were inaccurately predicted to pass the course had an attitude closer to the students who actually passed the course. Their online presence was active or their achieved grades, to that point, were adequate to consider them among the successful students, or both.

![Figure 9. The probability from the GBM model of the misclassified failing students vs. their dynamic data. The green lines represent the average value of the feature for the passing students. The red lines represent the average value of the feature for the failing students. The blue dotted lines represent the average value of the misclassified instances.](image)

5 Discussion

This section will address the evaluation and discussion of the achieved results. Furthermore, the benefits and limitations of the analysis will be discussed. Finally, this section will reflect on the conclusions that could be drawn from the analysis. Therefore, this section will focus on answering the research questions.

5.1 Usability

In order to answer the first research question, machine learning algorithms could be used to predict academic performance. Based on the experiment results, the algorithms could classify correctly 9 out of 10 students halfway through the course duration. Furthermore, to support the initial aim of helping the in-danger students, machine learning has proven to correctly identify 78.98% of failing students without compromising the overall performance of the model. The results of this analysis show promising signs for the actual implementation of the method by higher education institutions.
At the time of the prediction, the maximum cumulative percentage of the final grade is 37.50% (table 3). Therefore, the identified students have much time to amend their performance and get back in track to pass the course. As mentioned before, the analysis uses data that are widely available in every university and along with the promising results, its establishment is plausible.

<table>
<thead>
<tr>
<th>Module</th>
<th>Cumulative Grade</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>25.00%</td>
<td>0.72222222</td>
</tr>
<tr>
<td>B</td>
<td>31.00%</td>
<td>0.77298851</td>
</tr>
<tr>
<td>C</td>
<td>13.30%</td>
<td>0.79683377</td>
</tr>
<tr>
<td>D</td>
<td>28.50%</td>
<td>0.78583474</td>
</tr>
<tr>
<td>E</td>
<td>36.00%</td>
<td>0.87553648</td>
</tr>
<tr>
<td>F</td>
<td>37.50%</td>
<td>0.85751634</td>
</tr>
<tr>
<td>G</td>
<td>0.00%</td>
<td>0.79207921</td>
</tr>
</tbody>
</table>

*Table 3. Cumulative percentage of the final grade until the middle of the semester for each course. Along with the F1-Score of the GBM predictions for each course.*

### 5.2 Predicting Factors

Summarizing the findings of Section 4.3, the most powerful features for academic performance prediction are the metrics about the students VLE interaction. More particularly, the variables vleAvBef, vleSumNorm, distance, max_dist and times, which are related to usage and consistency of access, explain 64%, 54% and 82% of the final result variance for GBM, AdaBoost, and RF respectively. In short, the findings confirm the rationale that more engaged students have a higher probability of being successful in a course. While the statistics seem unsurprising, the positive outcome of the study is that the process of alerting in-danger students early could be automated to better support students and inform academics about the activity of their students.

The aforementioned variables are simple metrics of a student’s log activity, which can be obtained and calculated from every LMS and institution website. Furthermore, despite being available, no course-specific data related to the assessments, the available materials or the announcements were used. This exclusion enabled the algorithms to run on the whole dataset successfully. Additionally, it was noticed that the code_module feature was consistently ranked among the most important features of each model. This translates that the algorithms were able to distinguish between the differences of each course. While it is important to build different prediction models for each course due to their different characteristics, this approach could be useful in occurrences where limited data are available or as a cold start alternative for new courses or newly installed LMSs.

### 5.3 Best Performing Model

Early alert systems focus essentially on in-danger students. However, focusing only on the instances of a single class could lead to unwanted results. Consequently, the most important evaluation metric is F1-Score which balances precision and recall, helping to find a model which operates well on both classes of a binary classification problem. Based on F1-Score, the best classifier for the task is the GBM. GBM sequentially builds classification trees, optimizing the preceding tree in each iteration. The prediction is then made based on majority voting of the constructed trees, in case of a classification problem. Optimizing the loss function, allowed the algorithm to correctly classify more difficult data points in relation to the other ensembles. Furthermore, the creation of many trees enabled the distinction between different courses. Additionally, the importance of feature engineering must be noted. Feature engineering gave a great boost to the results of all the algorithms, highlighting the importance of feature selection and feature creation.
5.4 Implications

The findings of this study contribute to the literature in several ways. First, the analysis provides ground for the practical implementation of an early warning system from institutions or LMS providers. The used data were summarized log activity data which are available on every website, including LMSs and institutions’ websites. The various features derived directly from the log information of the LMS to describe the online activity of the students, avoiding data sources which may delay the development of prediction models. Most importantly, the standardized pre-processing of the data welcomes the automation of the process which establishes it as an appealing solution for practitioners. The analysis benefited from the usage of data for several different courses in a single prediction model, endorsing the generalizability of the used features.

This paper is one of the first to successfully perform early academic performance predictions on multiple courses. Advancing the study of Conijn, Snijders, Kleingeld and Matzat (2017), it was proved that ensemble methods are capable of handling data about different courses with different characteristics. Furthermore, this analysis confirmed the results of the previous studies on early warning systems[10,11,12], using a considerably larger dataset.

Furthermore, it does not provide any novel results concerning the most powerful factors in predicting academic performance. It confirms that machine learning could be used for the prediction of the task [1,7,9,13] and also that LMS activity data have high predicting capability[2,3,6]. While it is the first paper to support that GBM is the best algorithm for the task, the comparison between the studies, regarding the algorithms used, is not possible due to the different data in each application.

5.5 Limitations and Future Research

Although the use of the data produced substantial and significant results, this analysis had several limitations that could be addressed in future research. First of all, there was a difficulty in obtaining student data related to this task from a higher education institution. For that reason, the analysis was conducted on open data that was found on the internet. Furthermore, due to the reason that the data collection was out of the scope of this analysis, there was an absence of valuable information that would probably enhance the results. More importantly, the Grade Point Average (GPA) of the students at the time of the prediction, which was found to be one of the best predictors of student performance, was not available. Moreover, it would have been ideal to have more information about the modules such as the credits, the field of study and the instructor, even anonymized. These data would facilitate further in-depth analysis concerning each course individually and may produce useful and accurate results beyond the scope of this analysis. Further research is proposed to determine which machine learning algorithms operate better for the task of early academic performance prediction. For this task, there is a need for detailed information about the courses and it would be ideal to combine data from several institutions for different courses in order to test the efficiency of the algorithms.

5.6 Conclusion

The data analysis has proven that student performance in academic institutions could be predicted early enough, providing the in-danger students with a long interval to amend their engagement to the course. The interaction of students with the Virtual Learning Environment was found as a critical predictor of student performance, inferring that the involvement of students is crucial for their academic performance. The feature engineering process showed that there are different aspects of student interaction with the VLE and each one provides additional predicting ability to the algorithms. The difficulty in processing the vast amounts of data produced my LMSs call for the employment of analytical approaches as it would help the institutions to tackle key challenges such as student retention. In the future, further research will focus on the misclassified instances in order to determine the factors that resulted in failure given that the students were engaged to the course up to the point of the analysis.
References


