IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 5, 2018, accepted November 14, 2018, date of publication November 20, 2018,
date of current version December 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2882297

Predicting Academic Performance Based on Learner Traces in a Social Learning Environment

ELVIRA POPESCU[1], **(Member, IEEE), AND FLORIN LEON**[2]
[1]Computers and Information Technology Department, University of Craiova, 200585 Craiova, Romania
[2]Department of Computer Science and Engineering, Gheorghe Asachi Technical University of Iasi, 700050 Iasi, Romania

Corresponding author: Elvira Popescu (popescu_elvira@software.ucv.ro)

ABSTRACT Predictive modeling is an important part of learning analytics, whose main objective is to estimate student success, in terms of performance, knowledge, score, or grade. The data used for the predictive model can be either state-based data (e.g., demographics, psychological traits, and past performance) or event-driven data (i.e., based on student activity). The latter can be derived from students' interactions with educational systems and resources; learning management systems are a widely analyzed data source, while social media-based learning environments are scarcely explored. In this paper, our objective is to predict students' performance based on their social media traces. Data is collected from a Web Applications Design course, in which students use wiki, blog, and microblogging tools, for communication and collaboration activities in a project-based learning scenario. A total of 343 students, from six consecutive course installments, are included in the study. In addition to the novel settings and performance indicators, an innovative regression algorithm is used for grade prediction. Very good correlation coefficients are obtained and 85% of predictions are within one point of the actual grade, outperforming classic regression algorithms. From a pedagogical perspective, results indicate that, as a general rule, a higher engagement with social media tools correlates with a higher final grade.

INDEX TERMS Learning analytics, performance prediction, large margin nearest neighbor regression, social learning environment, social media.

I. INTRODUCTION

Learning analytics (LA) is a growing research area, which aims at selecting, analyzing and reporting student data (in their interaction with the online learning environment), finding patterns in student behavior, displaying relevant information in suggestive formats; the end goal is the prediction of student performance, the optimization of the educational platform and the implementation of personalized interventions [17]. According to the Society of Learning Analytics Research,[1] LA can be defined as "the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs" [47]. The topic is highly interdisciplinary, including machine learning techniques, educational data mining, statistical analysis, social network analysis, natural language processing, but also knowledge from learning sciences, pedagogy and sociology [16], [21]; up-to-date overviews of the area are provided in [5], [18], [28], [34], and [37].

Various educational tasks can be supported by learning analytics, as identified in [43]: analysis and visualization of data; providing feedback for supporting instructors; providing recommendations for students; predicting student's performance; student modeling; detecting undesirable student behaviors; grouping students; social network analysis; developing concept maps; constructing courseware; planning and scheduling. Similarly, seven main objectives of learning analytics are summarized in [13]: monitoring and analysis; prediction and intervention; tutoring and mentoring; assessment and feedback; adaptation; personalization and recommendation; reflection.

The prediction of students' performance is one of the most popular goals of LA [14], [31], which aims to estimate future learning outcomes and identify indicators for learning success [47]; more specifically, the objective is to develop a model which can infer the students' academic performance (i.e., the *predicted variable*, generally in the

[1] https://solaresearch.org

form of grades or scores) from a combination of various indicators (i.e., *predictor variables*) from the educational dataset [5]. The predictive information is highly valuable, as it can offer instructors the ability to monitor the learning progress and provide students with personalized feedback and interventions [51]; in particular, the instructor can be advised about students at-risk, who are in need of more assistance [8], [52]. In addition, individualized strategies for improving participation may also be suggested. Furthermore, the automatic prediction mechanism may be used for a formative assessment tool, which has the potential to decrease the instructors' assessment loads [52]. Finally, providing prediction results and personalized feedback can foster students' awareness [51].

Performance prediction has been extensively studied in web-based educational systems and, in particular, in Learning Management Systems (LMS) [14]. This is due to the availability of large amounts of student behavioral data, automatically logged by these systems, such as: visits and session times, accessed resources, assessment results, online activity and involvement in chats and forums, etc [41]. Thus, student performance prediction models based on Moodle log data have been proposed in multiple previous studies [12], [41], [42], [45], [53]. Additionally, log data from intelligent tutoring systems (ITS) have also been used for performance prediction [23], [35], [36]. In contrast, the students' engagement with social media tools in emerging social learning environments has been less investigated as a potential performance predictor [14], [25], [46].

Therefore, our objective is to address academic performance prediction based on social media traces, in the novel context of social learning environments. More specifically, we focus on our eMUSE platform [38], which integrates three social media tools (wiki, blog and micro-blogging tool). These tools were used by Computer Science students enrolled in a Web Applications Design course, to support communication and collaboration activities in a project-based learning (PBL) scenario. Data was collected from six consecutive course installments (unfolding over six years), with a total of 343 students, leading to a relatively large educational dataset. A further novelty of our approach consists in the use of an innovative regression algorithm called "Large Margin Nearest Neighbor Regression" (LMNNR) for grade prediction, based on students' activity on wiki, blog and micro-blogging tool. Very good results are obtained, outperforming commonly used regression algorithms.

The rest of the paper is structured as follows: in section II we provide an overview of related work on performance prediction. Subsequently, in section III we present a short technical background, including a description of the LMNNR algorithm and an overview of the algorithms used for comparison. The results obtained by applying these algorithms in our social learning environment context (which is described in more detail in section IV) are reported and discussed in section V. We end the paper with some conclusions and future research directions.

II. RELATED WORK

Predictive modeling for teaching and learning is an important part of learning analytics; its main objective is to predict student success, in terms of academic achievement [10]. The predicted values can be performance, knowledge, score or grade; classification approaches are generally employed for categorical/discrete values, and regression approaches for numerical/continuous values [43].

Two types of data can be used for the predictive models: i) state-based data, e.g.: demographics, psychological traits, past performance; ii) event-driven data, i.e., based on student activity, as derived from the students' interactions with educational systems and resources [10]. The latter may be structured (e.g., server logs) or unstructured (e.g., forum postings) [48]. It may come from centralized educational systems (e.g., LMS) or distributed learning environments (e.g., formal and informal platforms, spread across space, time and media) [13]. The data sources may include also MOOCs (massive open online courses), social media or wearable sensors, their integration leading to a higher accuracy of the learner models [34]. A further classification of the performance indicators, provided in [11], includes three categories: i) dispositional indicators (e.g., age, gender, previous learning experiences); ii) activity and performance indicators (e.g., number of logins, time spent, number of discussion posts); iii) student artifacts (e.g., essays, blog posts, forum discussions) [47]. About 200 indicators were identified in a review conducted in [15]; among them, the most commonly used are: demographic characteristics, previous grades, portfolios, multimodal skills, levels of participation and engagement, mood and affective states [34].

As far as computational techniques are involved, a wide variety of methods have been applied for predicting students' performance [31], such as linear regression [40], logistic regression [6], neural network models [12], support vector machines and k-nearest neighbors [23], [44], Bayesian networks [35], [36], decision trees [50] or genetic algorithms [51], [53].

While providing a full review of the literature is beyond the scope of this paper, in what follows we describe a few (recent) initiatives in academic performance prediction, which are more closely related to our work.

Romero *et al.* [41] explored students' usage data in Moodle LMS as a predictor for their final exam grade. 438 students from seven engineering courses were included in the study. Eight attributes related to learner activity on quizzes, assignments and forum messages were computed for each student. The authors applied various data mining techniques for classifying students with similar final grades (statistical classifier, decision tree, rule induction, fuzzy rule learning, neural networks). Performance comparisons were carried out, with various pre-processing techniques (filtering, discretization and rebalancing). Overall, the accuracy obtained is not very high (around 65%), indicating the difficulty of the prediction task.

Saqr *et al.* [45] also analyzed students' online activity in a LMS, in the context of a blended medical course, aiming to correlate it with the learners' final performance. 133 students used Moodle for six weeks and various types of data were collected: logins, resource views, forum posts and reads, time spent using educational materials, formative assessment results. Five engagement indicators were computed based on students' traces. Automatic linear modeling was used for grade prediction, leading to a 63.5% accuracy. In addition, binary logistic regression was employed for predicting students at risk, with an accuracy of 80.8%.

Romero *et al.* [42] focused on the use of students' participation in a discussion forum as an indicator of learner performance. Data was collected from 114 students enrolled in an introductory computer science course. They used the forum included in Moodle LMS for discussing the course contents, asking questions or providing help to peers and took a final exam at the end of the semester. The authors aimed to predict whether students passed or failed the course based on their forum usage, in terms of quantitative, qualitative and social network indicators. A comparison between traditional classification and clustering algorithms implemented in Weka [22] was performed, together with various approaches for instance and attribute selection. Good results were obtained both for the prediction at the end of the course and for an early prediction carried out mid-course.

In the context of a different learning environment, Junco and Clem [24] investigated students' interaction patterns with digital textbooks as predictors of final course grades. Data was collected from 233 students from 11 courses (such as Introduction to Accounting, American Judicial Process, Human Resource Management etc.) who used a digital textbook offered by CourseSmart provider. The authors performed linear regression analyses on textbook usage metrics and found out that time spent reading was a strong predictor of final course grade. The Engagement Index score (computed by CourseSmart based on various usage metrics) was also a good indicator of the course outcome (better than prior academic achievement).

As it can be seen, most of the above studies were performed in the context of Moodle LMS; the social learning environments and students' social media traces were much less explored in the literature. There are however a couple of notable exceptions, as summarized below.

Junco *et al.* [25] used mixed effects analysis of variance (ANOVA) models to evaluate the impact of using Twitter on college student engagement and learning outcome. Engagement was measured with a dedicated instrument called National Survey of Student Engagement. Results showed a significant increase in both engagement and grades for the experimental group, in which students used Twitter for various types of academic discussions.

Stafford *et al.* [46] investigated whether students' engagement with a collaborative wiki tool can predict academic performance. Significant correlations were found between wiki activity indicators (number of page edits, number of different articles edited, number of days on which the student edited the wiki) and the final grade. Overall, the students who were engaged with the wiki (both high- and low-grading ones) obtained higher exam scores, with an average increase of 5 percentage points.

The novelty of our current work consists in the use of an original algorithm, called Large Margin Nearest Neighbor Regression (rather than classic algorithms, available in various data mining engines, as mentioned in the related works). A preliminary study based on only one student cohort yielded encouraging results [31]; this paper extends the pilot study to a much larger number of students (six cohorts, over the course of six years), also providing a refinement of the LMNNR algorithm, as described next.

III. BACKGROUND ON ALGORITHMS

As mentioned in the previous section, estimating the final grade of the students based on different attributes related to their learning activities can be viewed as a typical regression problem. Currently, there are many available models and several collections of machine learning algorithms, among which the best known is arguably *Weka* [22]. In this paper, we aim to compare the performance of classical algorithms with an original one, called *Large-Margin Nearest Neighbor Regression* (*LMNNR*). Following our experience from previous work [31], here we only consider two classical models that provided the best results for our particular problem, i.e., Random Forest (RF) and k-Nearest Neighbors (kNN), which we briefly describe below.

A *random forest* [9] is composed of a collection of classification or regression trees. Each tree is generated using random split tests on slightly different data, using bagging. The output of a new instance is computed by aggregating the outputs of the individual trees, by voting or averaging.

The *k-Nearest Neighbors* algorithm is based on the choice of k nearest neighbors using some distance function, and the output is computed by aggregating the outputs of those $k$training instances. As a distance function, one can use Euclidian or Manhattan distance, usually particularizations of the Minkowski distance. Choosing the value of k is very important: if k is too small, then the classification can be affected by the noise of the input data, and if the value of k is too large, then some of the neighbors considered may be irrelevant for the classification. To avoid the difficulty of finding an optimum value for k, one can weight the influence of the neighbors. The neighbors that are closer to the query instance have a greater weight, while those farther apart have a smaller weight. Cross-validation can also be used to assess the optimal number of neighbors.

kNN is a simple yet powerful method, typically when the data is not affected by noise and does not have too many dimensions. However, in its classical formulation it does not adapt the distance metric, which is of great importance in this case, to the problem at hand. Therefore, an extension of the method is to change the distance metric of the problem space by using a matrix **M**:

$$d_M(\mathbf{x}_i, \mathbf{x}_j) = \left(\mathbf{x}_i - \mathbf{x}_j\right)^T \mathbf{M} \left(\mathbf{x}_i - \mathbf{x}_j\right) \tag{1}$$

One way of computing such a matrix for classification problems was provided in [49], which incorporated the idea of a large margin, typical of support vector machines. The authors designed a convex semidefinite programming optimization problem so that by finding **M**, the classes of the data should be separated by a margin larger than an arbitrary value, e.g., 1.

The concept of a large margin in a regression setting was used in [29] and [30], resulting in the LMNNR algorithm, which is based on the optimization of two conflicting objective functions (and an additional, optional one that ensures regularization). It also simplifies the interpretation of the results by imposing that **M** is a diagonal matrix, and thus the weights of the neighbors are:

$$w_{d_M}(\mathbf{x}, \mathbf{x}') = \frac{1}{d_M(\mathbf{x}, \mathbf{x}')} = \frac{1}{\sum_{i=1}^{n} m_{ii} \cdot \left(x_i - x_i'\right)^2} \tag{2}$$

Eq. (2) involves a single, global matrix **M** for all the instances. However, it is possible to have different distance metrics for different instances or groups of instances. Thus, *prototypes* can be used. They are defined as special locations in the input space of the problem, and each prototype P has its own matrix $\mathbf{M}^P$. When computing the distance weight to a query point, an instance uses the weights of its nearest prototype, i.e., m_{ii}^P instead of m_{ii} in Eq. (2).

Like all regression algorithms, LMNNR tries to find a function $\tilde{f}$ which is an approximation of the dependent variable, i.e., the output of the model:

$$\tilde{f}(\mathbf{x}) = \frac{\sum_{\mathbf{x}' \in N(\mathbf{x})} w_{d_M}(\mathbf{x},\ \mathbf{x}') \cdot f(\mathbf{x}')}{\sum_{\mathbf{x}' \in N(\mathbf{x})} w_{d_M}(\mathbf{x},\ \mathbf{x}')}, \tag{3}$$

where w_{d_M} are the weights found by the algorithm, f ($\mathbf{x}$) is the output value corresponding to instance $\mathbf{x}$ in the dataset, and $N(\mathbf{x})$ is the set of the nearest neighbors of $\mathbf{x}$ in the dataset. If we use the model for prediction, $\mathbf{x}$ does not belong to the training dataset, but the neighbors $N(\mathbf{x})$ do. The number of nearest neighbors is a parameter specified by the user. Basically, $\tilde{f}(\mathbf{x})$ is a weighted average of the output values of the neighbors of $\mathbf{x}$.

As mentioned earlier, the distance metric matrices are found by solving an optimization problem. In the simplified formulation (without regularization, which does not seem to be needed for our particular problem), the objective function F, to be minimized, takes into account two criteria, F_1 and F_2, described below. In order to explain the expressions of these functions, the following notations are used, where d_M is the weighted square distance function using the weights we search for: $d_{ij} = d_M(x_i, x_j)$, $d_{ik} = d_M(x_i, x_k)$, $g_{ij} = |f(x_i)–f(x_j)|$, and $g_{ik} = |f(x_i)–f(x_k)|$.

The first criterion is:

$$F_1 = \sum_{i=1}^{n} \sum_{j \in N(i)} d_{ij} \cdot \left(1 - g_{ij}\right), \tag{4}$$

where $N(i)$ is the set of the k nearest neighbors of instance i.

This is based on the following intuition. If two neighbor instances, i and j, have close output values, then g_{ij} is small, let us say it is close to 0. Therefore the second factor is large, close to 1. Thus the optimization process minimizes the distance d_{ij} between them. Now let us assume that i and j have different output values. Then the second factor is close to 0, and the effect of the minimization on d_{ij} is negligible. Since d and g belong to the [0, 1] interval, we can see that the closer the output values of the neighbors are, the closer their positions become in the transformed space.

This criterion states that the nearest neighbors of i should have output values similar to the output value of i, and more distant ones should have different values. The same effect is intended for all training instances, therefore we use the sum in Eq. (4)

While the equation of the first criterion is the same as the one used in an earlier version of the algorithm [31], the equation for the second criterion has been changed compared to the one used in previous works. It combines the idea from our previous formulation with that presented in [4], a study that defines a semidefinite programming problem by close analogy with the work in [49].

In order to explain the second criterion in its present form, let us assume that we have an instance i and two different neighbors of i: j and l. We use the "l" notation because "k" is the traditional notation for the number of neighbors in an instance-based algorithm such as kNN.

Let us define:

$$\Delta d_{ijl} = \max\left(d_{il} - d_{ij},\ 0\right). \tag{5}$$

This means that Δd_{ijl} is the difference of the corresponding distances only if l is farther from i than j. Otherwise, it is 0.

We can define a similar value for g:

$$\Delta g_{ijl} = \max\left(g_{ij} - g_{il},\ 0\right), \tag{6}$$

but one should notice the change in indices compared to the definition of Δd_{ijl}: the j and l indices are swapped.

The main idea of the second criterion is to reduce the situations when l is farther than j, but its output value is closer to that of i than the output value of j. We would prefer the situations where g decreases monotonically as d increases, and vice versa.

Thus, when both Δd_{ijl} and Δg_{ijl} are strictly positive, this is equivalent to a proximity order break. By learning the proper distance metrics, we aim at minimizing these cases, therefore we introduce a penalty term p_{ijl} which is strictly positive when $\Delta d_{ijl} \cdot \Delta y_{ijl} \neq 0$ and 0 otherwise.

We can further incorporate the concept of a "large margin": we impose that p_{ijl} not only include $\Delta d_{ijl} \cdot \Delta g_{ijl}$, but also a “margin‘’ arbitrarily set to 1:

$$p_{ijl} = \begin{cases} 1 + \Delta d_{ijl} \cdot \Delta g_{ijl}, & \textit{if } \Delta d_{ijl} \cdot \Delta g_{ijl} \neq 0 \\ 0, & \textit{otherwise}. \end{cases} \quad (7)$$

As it was proven in [49] in the context of classification, the actual width of the margin is not important; different values simply lead to the scaling of the distance metric matrices, accordingly.

Again, the criterion should be minimized for all the instances i and all pairs of neighbors, therefore:

$$F_2 = \sum_{i=1}^{n} \sum_{j \in N(i)} \sum_{l \in N(i)} p_{ijl}. \quad (8)$$

A more compact expression for Eq. (8) is:

$$F_2 = \sum_{i=1}^{n} \sum_{j \in N(i)} \sum_{l \in N(i)} \Delta d_{ijl} \cdot \Delta y_{ijl} \cdot \left(1 + \frac{1}{\Delta d_{ijl} \cdot \Delta g_{ijl} + \varepsilon}\right), \quad (9)$$

where ε is a small positive real number that ensures that the fraction can be computed when $\Delta d_{ijl} \cdot \Delta g_{ijl} = 0$.

According to Eq. (9), when $\Delta d_{ijl} \cdot \Delta g_{ijl} = 0$, $p_{ijl} = 0$. When $\Delta d_{ijl} \cdot \Delta g_{ijl} = a > 0$:

$$p_{ijl} = a \cdot \left(1 + \frac{1}{a + \varepsilon}\right) = a + \frac{a}{a + \varepsilon} \approx 1 + a. \quad (10)$$

In [31], the expression for the second criterion was created by analogy with F_1 and it only took into account d and g:

$$F_2' = \sum_{i=1}^{n} \sum_{j \in N(i)} \sum_{l \in N(i)} \max\left(1 + d_{ij} \cdot (1 - g_{ij}) - d_{ik} \cdot (1 - g_{il}),\ 0\right). \quad (11)$$

Its intent was the same: to minimize the distance to the neighbors with close values (the positive term), while simultaneously trying to maximize the distance to the neighbors with distant values (the negative term). However, it was experimentally discovered that the present formulation of F_2 in Eq. (8) or Eq. (9) provides better results.

If the effect of F_1 is to attract similar neighbors, the effect of F_2 is to repel the neighbors that violate the proximity order, as described above.

An additional change from [31] is that in the final objective function F, the two criteria no longer receive equal weights, because it was empirically observed that F_2 has a much larger value than F_1 and tends to dominate F. Therefore, in order to balance the influence of the two terms, the final expression to be minimized is:

$$F = F_1 + \sqrt{F_2}. \quad (12)$$

This refined LMNNR algorithm was applied on a relatively large educational dataset, involving six student cohorts, as described in the next section.

IV. CONTEXT OF STUDY

A. INSTRUCTIONAL SCENARIO

Our study took place in the context of an undergraduate course for Computer Science students, on Web Applications Design (WAD). The instructional approach was project-based learning (PBL) in which the students had to work collaboratively on various complex, challenging and authentic tasks, over extended periods of time; learning was organized around team projects, while the teacher played the role of a facilitator [27]. More specifically, the students collaborated in teams of around 4 peers in order to build a complex web application of their choice (e.g., a virtual bookstore, an online auction website, a professional social network, an online travel agency, etc.). The project spanned over the whole semester and the evaluation took into account both the final product and the continuous collaborative work.

Since PBL has a strong social component, the increasingly popular social media tools appear suitable for communication and collaboration support in PBL framework [3], [26]. Hence, we implemented our PBL scenario with the help of several social media tools (wiki, blog, and microblogging tool) integrated in our social learning environment, called eMUSE [38]. More specifically, a blended learning approach was used consisting of weekly face-to-face meetings between each team and the instructor (for checking the project progress, providing feedback and answering questions), while students had to rely on the social media tools for the rest of the time, as a support for their communication and collaboration activities.

In particular, MediaWiki[2] was used for collaborative writing tasks, for gathering and organizing the team knowledge-base and resources, and for documenting the project. Blogger[3] was used for reporting the progress of each project, similar to a “learning diary” in terms of publishing ideas and resources, as well as for providing feedback and solutions to peer problems. Each team had its own blog, but inter-team cooperation was encouraged as well. Twitter[4] was meant to foster additional connections between peers and to encourage the posting of short news, announcements, questions, and status updates regarding each project. The eMUSE social learning platform provides an integration point for the social media tools, together with additional support for both students and teachers: basic administrative services, learner tracking and data visualizations, as well as evaluation and grading functionalities [38]. eMUSE also offers data collection mechanisms, as detailed in the next subsection.

Of course, students could choose to use additional communication channels for working on their projects, including face-to-face meetings, phone calls, chats, email, document sharing or other social media tools. Obviously, these could not be monitored by eMUSE; this means that a part of learner data may not be collected, which is a general limitation of

[2] https://www.mediawiki.org
[3] https://www.blogger.com
[4] https://twitter.com

learning analytics approaches based on student activity indicators. In order to mitigate this problem in our PBL scenario, we provided specific instructions to the learners at the beginning of the semester: students were clearly informed that their collaborative learning activity needs to be documented on the social media tools integrated in eMUSE, so that it can be assessed by the instructor. We therefore expect that a large part of the students' communication and collaboration activities indeed took place on the three recommended social media tools.

B. DATA COLLECTION AND PREPROCESSING

The instructional scenario described above has been applied over 6 consecutive winter semesters (Year 1: 2010/2011 – Year 6: 2015/2016), with 4th year undergraduate students in Computer Science from the University of Craiova, Romania. Small improvements and refinements were made from one year to the consecutive one, based on students' feedback and instructor experience.

A total of 343 students, enrolled in the WAD course, participated in this study. All student actions on the three social media tools were monitored and recorded in the eMUSE platform. The system retrieves learner actions from each of the disparate Web 2.0 tools (by means of open APIs or Atom/RSS feeds) and stores them in a local database, together with a description and an associated timestamp. Thus, a total of almost 19000 social media contributions were recorded: 2609 blog posts and comments, 5470 tweets, 10895 wiki page revisions and file uploads.

Based on these actions, a set of 14 numeric features were computed for each student:

- *NO_BLOG_POSTS* (the number of blog posts)
- *NO_BLOG_COM* (the number of blog comments)
- *AVG_BLOG_POST_LENGTH* (the average length of a blog post)
- *AVG_BLOG_COM_LENGTH* (the average length of a blog comment)
- *NO_ACTIVE_DAYS_BLOG* (the number of days in which a student was active on the blog, i.e., wrote at least a post or a comment)
- *NO_ACTIVE_DAYS_BLOG_POST* (the number of days in which a student wrote at least a post on the blog)
- *NO_ACTIVE_DAYS_BLOG_COM* (the number of days in which a student wrote at least a comment on the blog)
- *NO_TWEETS* (the number of tweets)
- *NO_ACTIVE_DAYS_TWITTER* (the number of days in which a student was active on Twitter, i.e., posted at least a tweet)
- *NO_WIKI_REV* (the number of wiki page revisions)
- *NO_WIKI_FILES* (the number of files uploaded on the wiki)
- *NO_ACTIVE_DAYS_WIKI* (the number of days in which a student was active on the wiki, i.e., revised at least a page or uploaded at least a file)
- *NO_ACTIVE_DAYS_WIKI_REV* (the number of days in which a student revised at least a wiki page)
- *NO_ACTIVE_DAYS_WIKI_FILES* (the number of days in which a student uploaded at least a file on the wiki) [31].

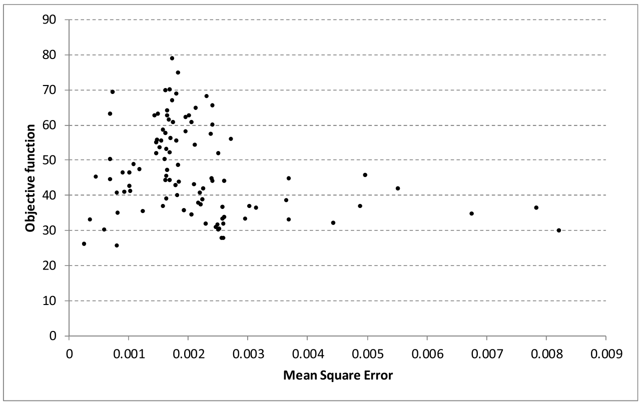

FIGURE 1. LMNNR convergence examples.

Students' performance at the end of the semester was assessed on a 1 to 10 scale, with 5 being the minimum passing grade; the evaluation took into consideration both the final project, as well as each student's continuous collaborative work. Our aim is to predict this final grade based on the above set of features, using the LMNNR algorithm, as described in the next section.

It should be noted that these features cover relevant quantitative characteristics which could be computed based on the recorded student actions: number of posts/edits, the length of their content and their time distribution. Furthermore, these actions include various types of learning activities: creating and organizing content, social interactions, communication and feedback. These data represent tacit student actions, which are not usually directly assessed as part of the learner's educational progress (as are the explicit student actions, such as completing assignments and taking exams) [38]. Of course, these are quantitative indicators only, as they do not take into account the quality of the learner actions (e.g., correctness of the wiki page, content of the blog post, relevance of the tweet). They are however a good measure of the level of involvement of the student, especially when considering also the number of active days per semester.

While some features may be considered more important than others from a pedagogical perspective, the LMNNR algorithm was proven to perform equally better with or without feature selection; there was even a slight increase of correlation when the full set of attributes was used [31]. This can be explained by the nature of the algorithm, which implicitly searches for the importance of the input attributes. Therefore, there is no need to manually reduce the number of features; on the contrary, it seems that the algorithm is able to use the additional information better than other regression algorithms [31].

V. DATA ANALYSIS – RESULTS AND DISCUSSION

In this section, we present the results obtained with the LMNNR algorithm, in comparison with those of the

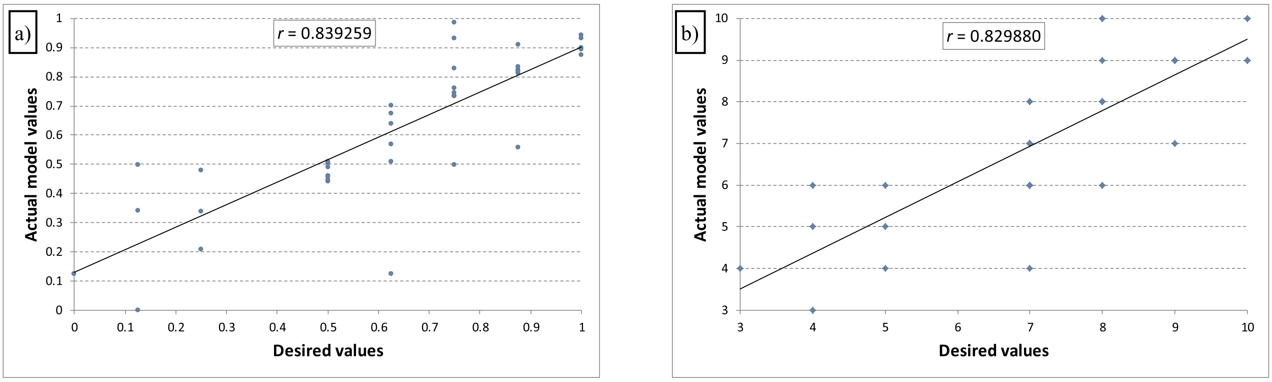

FIGURE 2. Comparison between the predictions of the model and the expected data for Year 1: a) the normalized data processed by the algorithm; b) the data transformed into integer grades.

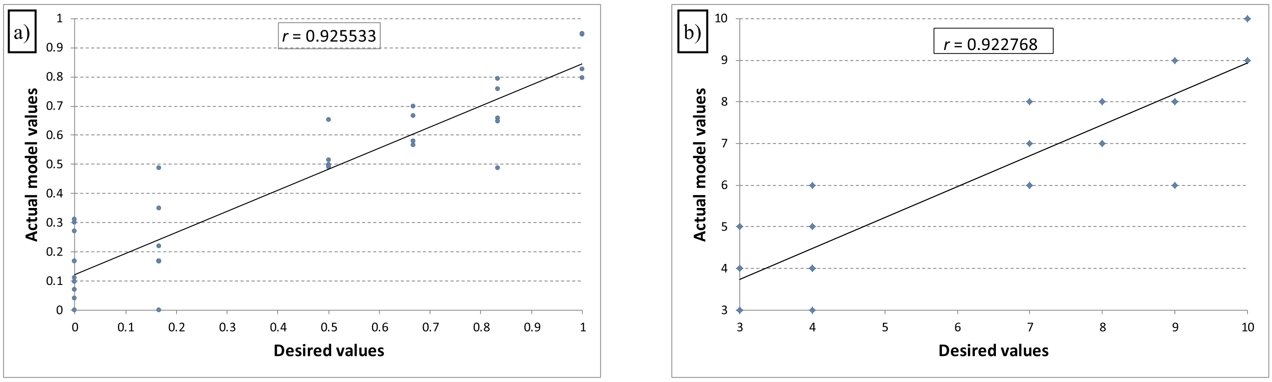

FIGURE 3. Comparison between the predictions of the model and the expected data for Year 2: a) the normalized data processed by the algorithm; b) the data transformed into integer grades.

algorithms that provided the best results in [31], namely *Random Forest* with 100 trees and *k-Nearest Neighbors*, with k obtained by cross-validation and inverse-distance weighting of the neighbors. In [31], an additional setting for *kNN* implemented in Weka was that mean absolute error was used when doing cross-validation. In this paper, we use mean squared error (MSE) instead, as we observed that it slightly improves the accuracy of the *kNN* model.

For the analysis of individual years, we chose 3 neighbors and 1 prototype for LMNNR, because it is a simple model that also provides good results. A drawback of LMNNR training is that it sometimes converges into local optima. Figure 1 shows several examples of convergence, plotting the value of the obtained objective function F against the corresponding *MSE*. It can be seen that the lowest value of F does lead to the lowest value of the *MSE*, but there are also many situations when the obtained *MSE* is not so good, even for low values of F. That is why we run the algorithm several times and retain the best results. Even if this requires additional training time, we consider that the quality of the obtained results, which are clearly better than those found by the other models, compensate for this inconvenience. In a previous work [29], an evolutionary algorithm was used for training, but the gradient-based method used here is much faster, although it needs multiple starting points.

TABLE 1. Performance of the considered regression algorithms (correlation coefficient).

Dataset	*RF*	*kNN*	*LMNNR*
Year 1	0.6167	0.5387, $k = 9$	0.839259
Year 2	0.6021	0.5318, $k = 10$	0.925533
Year 3	0.2164	0.2722, $k = 8$	0.835959
Year 4	0.6733	0.706, $k = 9$	0.918769
Year 5	0.738	0.6919, $k = 9$	0.899269
Year 6	0.6795	0.5807, k=5	0.868894
All years combined	0.6593	0.6374, $k = 10$	0.754682
All data normalized by year	0.6464	0.6288, $k = 10$	0.732289

In Table 1, we present the correlation coefficients r obtained by the algorithms for our problems. We chose to use r instead of *MSE* because we considered it to be more intuitive, as each experiment can be evaluated in terms of how close its correlation coefficient is to 1, i.e., perfect match.

In the following, we evaluate the results provided by the LMNNR algorithm.

Figures 2-7 display the comparisons between the test results and the desired outputs by year. The scale of the charts on the left side is the [0, 1] interval because all the data is normalized attribute-wise, before applying the algorithm.

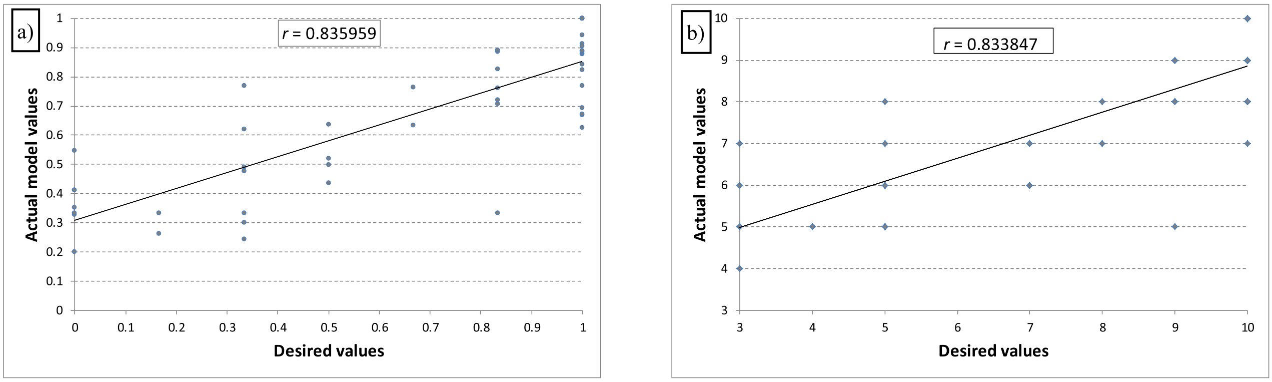

FIGURE 4. Comparison between the predictions of the model and the expected data for Year 3: a) the normalized data processed by the algorithm; b) the data transformed into integer grades.

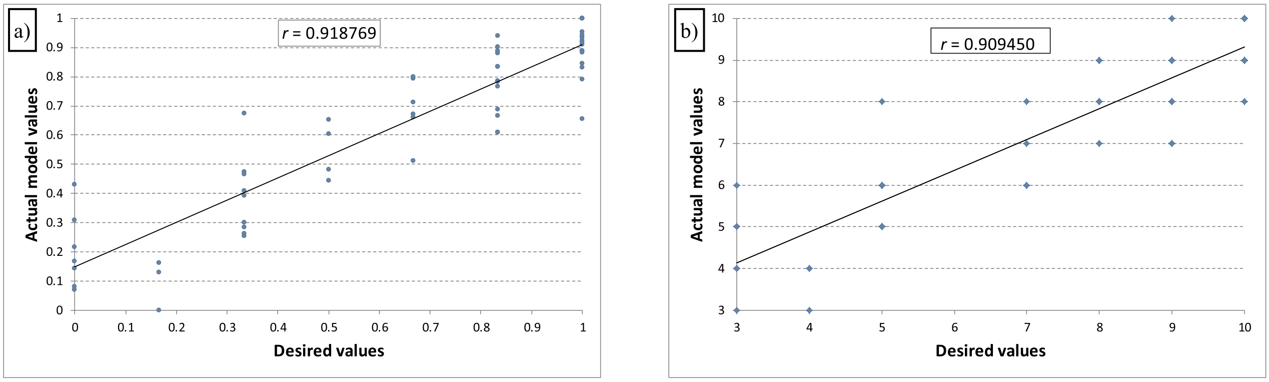

FIGURE 5. Comparison between the predictions of the model and the expected data for Year 4: a) the normalized data processed by the algorithm; b) the data transformed into integer grades.

The LMNNR algorithm performs perfectly (i.e., $r = 1$) on data that belong to the training set, unless there are different instances in the training set with the same inputs but different output values. Therefore, the results presented here are those provided by 10-fold cross-validation, which is a *de facto* standard of comparing the performance of different classification or regression methods. Thus, the dataset is divided into ten equally sized groups (or ''bins''), and in one fold, a model is built on the union of nine bins and tested on the tenth bin. The procedure is repeated ten times, with the test bin iterated. The final results are calculated for the ten combined test bins, which actually represent the entire dataset. Thus, the algorithms are only evaluated for their performance on test datasets.

We also included the corresponding charts (on the right side of the figures) when the normalized results are converted back to their initial domain, to integer grades between 3 and 10, where 10 is the best grade. The correlation coefficients in the two situations are very close, yet one can see that by rounding, some information is lost and r is slightly lower in the right side charts compared to the left side ones.

Finally, we try to create a model for the combined data of all six years. In the first variant, whose predictions are presented in Fig. 8, the six datasets were directly merged, and then normalization was performed attribute-wise on the entire resulting dataset, as required by the LMNNR algorithm. In the second variant, whose predictions are presented in Fig. 9, each of the six datasets was separately normalized, and then the six normalized datasets were merged and processed with the LMNNR algorithm. In both scenarios the correlation coefficient is lower than the values obtained for individual years. This shows that each year has specific characteristics caused by the particular dynamics of the students. However, there are still common characteristics of the six years, otherwise the correlation coefficient would be much lower. This is an interesting result showing that the behavior of students that belong to different years of study presents a combination of stable and variable elements.

Figures 10 and 12 show the errors of the models as actual differences between the predicted and desired grade values. The shape of the graph is a slightly skewed Gaussian distribution, which is mainly caused by the number of the training instances, which is not so large.

More importantly, this analysis shows that, on average, 85% of predictions are within only 1 point of the actual grade (Fig. 11). This emphasizes the fact that the model is capable of good approximation for our problem.

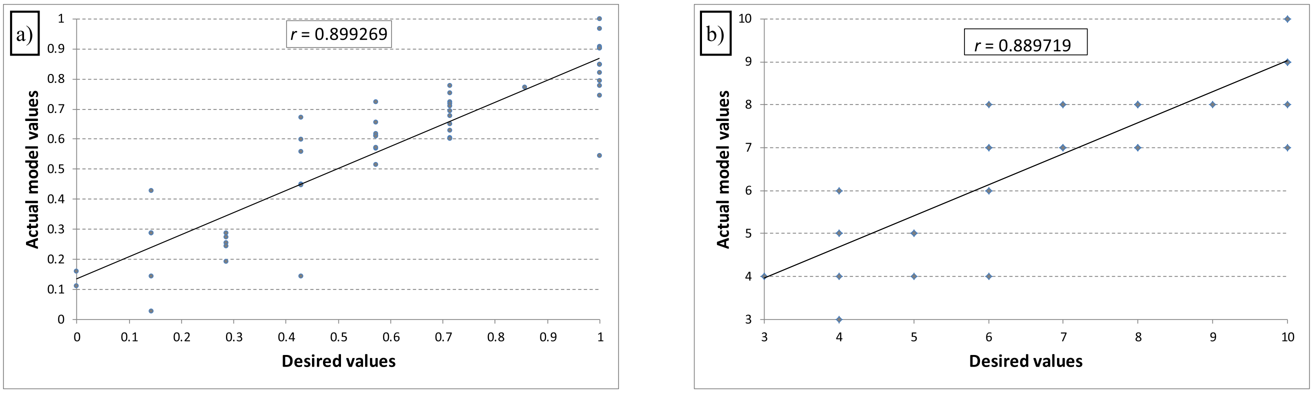

FIGURE 6. Comparison between the predictions of the model and the expected data for Year 5: a) the normalized data processed by the algorithm; b) the data transformed into integer grades.

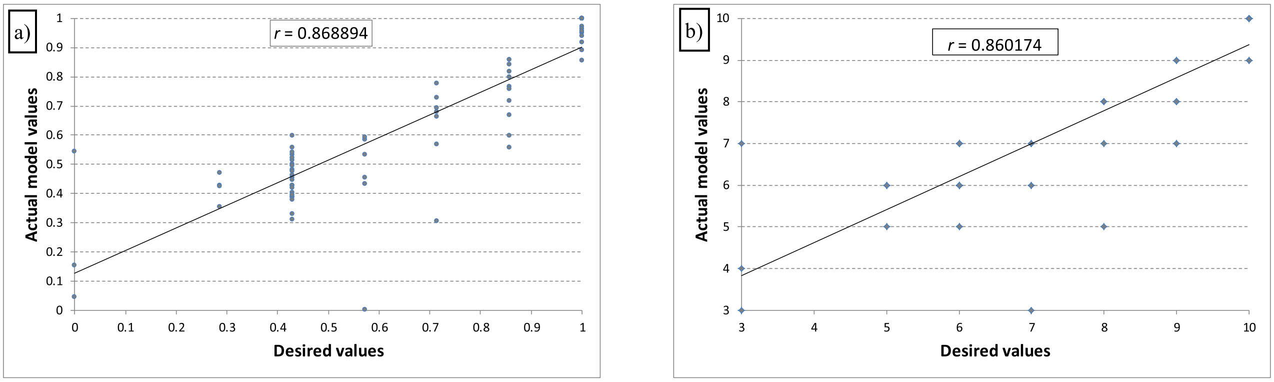

FIGURE 7. Comparison between the predictions of the model and the expected data for Year 6: a) the normalized data processed by the algorithm; b) the data transformed into integer grades.

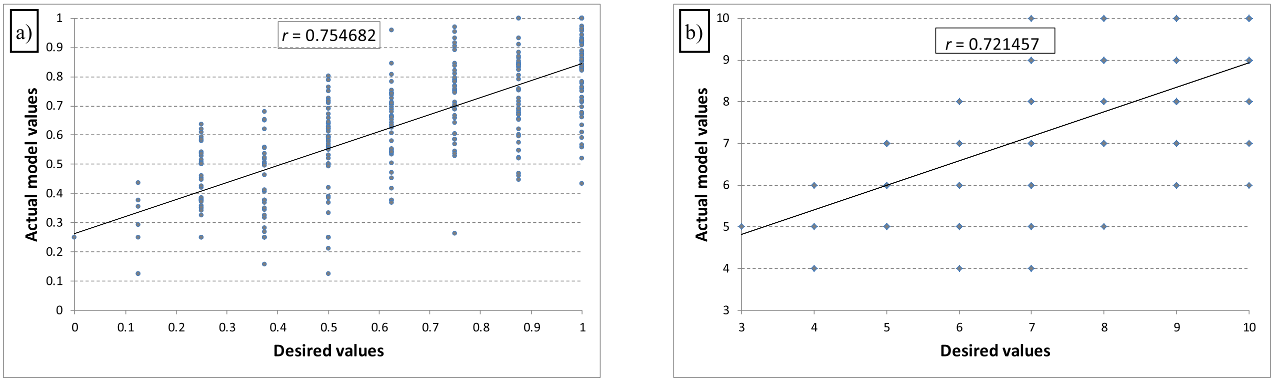

FIGURE 8. Comparison between the predictions of the model and the expected data for all years directly combined: a) the normalized data processed by the algorithm; b) the data transformed into integer grades.

In case of the combined years (Fig. 12), the errors are slightly higher, as the previous results have already shown. Here, 79% of predictions are within 1 point of the actual grade and 96% of predictions are within 2 points of the actual grade. One may also notice that the number of errors for the difference of −4 is 0, and for the differences of −3 and 4 it is only 1. In Fig. 10, the corresponding values for the individual years are a little higher. This is because by combining the data from all years, a new dataset resulted, and the algorithm learned this model independently.

One of the main motivations for analyzing learning data is the ability to predict student behavior early in the course, therefore we also assessed how useful the trained model is for predicting student behavior in future years. Thus, we trained the algorithm on data from earlier years and tested its prediction performance on data from later years. The results are presented in Table 2.

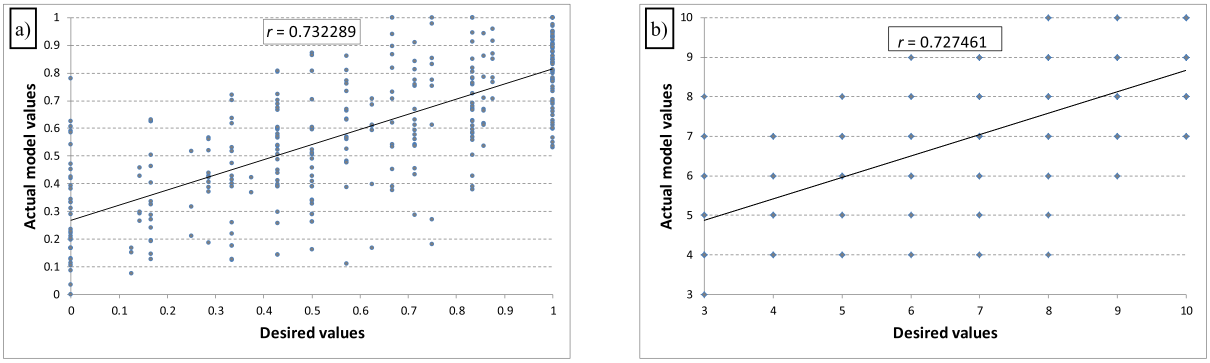

FIGURE 9. Comparison between the predictions of the model and the expected data normalized by year and then combined: a) the normalized data processed by the algorithm; b) the data transformed into integer grades.

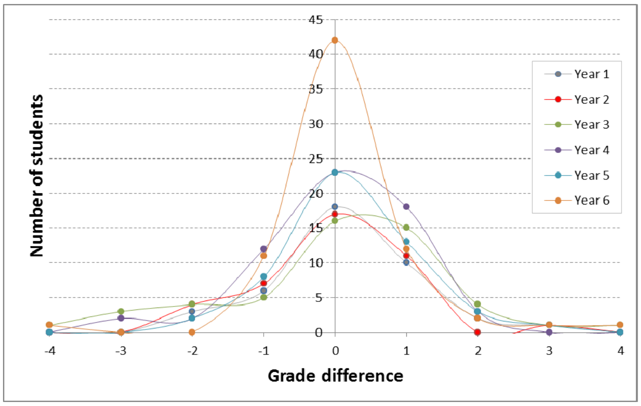

FIGURE 10. The differences between the predictions of the model and the expected data transformed into integer grades (Year 1 – Year 6).

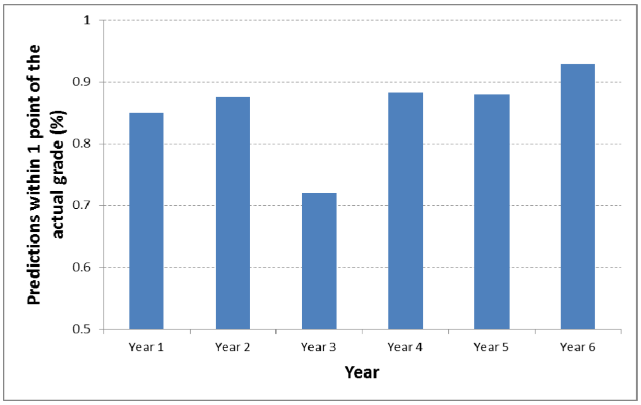

FIGURE 11. Percentage of predictions within only 1 point of the actual grade.

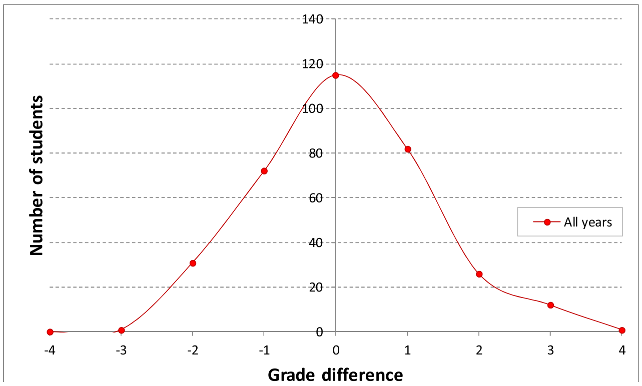

FIGURE 12. The differences between the predictions of the model and the expected data transformed into integer grades (all years combined).

TABLE 2. Evaluation of the ability to predict future trends.

Training → *Testing ↓*	*Year 1*	*Year 2*	*Year 3*	*Year 4*	*Year 5*
Year 2	0.7418				
Year 3	0.6161	0.6056			
Year 4	0.7101	0.7052	0.7726		
Year 5	0.7696	0.8004	0.7217	0.7518	
Year 6	0.7671	0.6478	0.5997	0.7492	0.7165

The obtained correlations are comparable to those found for all years combined. Nevertheless, there is a difference: in the previous cases, we used cross-validation for the same dataset, and in one fold 90% of the data was used for training and 10% for testing. Here, the whole dataset of a year is used for training and the whole dataset of another year is used for testing. Thus, the problem is more difficult.

Table 2 shows that even if different years have different characteristics, they also have a large degree of consistency, thus the existing model can be used to anticipate and correct or enhance certain student trends as they are identified.

VI. CONCLUSION

The study has shown that students' actions on social media tools are good predictors of academic performance. The innovative LMNNR algorithm proved very suitable for our prediction problem, outperforming classic regression algorithms. Very good correlation coefficients were obtained and 85% of predictions were within only 1 point of the actual grade.

From a pedagogical perspective, the results indicate that, as a general rule, a higher engagement with social media tools correlates with a higher final grade. This is in line with several previous studies, which found that online participation is a strong indicator of student performance [2] and improves learning effectiveness [20], [33], [54]. Nevertheless, there are also contradictory studies, which concluded that students learned equally well regardless of their level of online participation [1], [32]. At the same time, the body of literature specifically focused on students' active participation on social media is scarce, hence the novelty and added value of our study.

It is worth mentioning that the performance of the generalized predictive model (from all six years combined) is slightly lower than the performance of each individual year model. This is in line with the findings in [19], which addresses the issue of aggregating trace data from different courses for creating one generalized model for academic success prediction. The differences in instructional conditions and technology use, even in the context of the same discipline, may influence the prediction of academic success; in addition, the individual differences of the students involved in the studies (e.g., metacognitive and motivational factors) may have an impact on the learning analytics results.

Hence, the findings obtained in this paper need to be interpreted in the context of our specific instructional scenario; this is the case for most of the studies on academic performance prediction, which rely on data collected from one or few courses in the same discipline [19]. Nevertheless, the model offers interesting insights into the learning process, in particular in the context of a PBL scenario supported by social media tools. We also showed that the model may be used with good results from one year to the other, although the specificities of each year may lead to slightly different patterns of feature influence.

Any attempt at generalizability needs to carefully consider the pedagogical and disciplinary context of the predictive model. Therefore, further studies are needed for comparing courses with different internal and external conditions [19]. Hence, investigating the LMNNR algorithm performance on student data collected from different courses and instructional scenarios is an interesting research direction. Furthermore, combining the predictive analytics approach proposed here with our previous work on social network analytics [7] and discourse analytics [14], [39] could lead to a more comprehensive perspective on the social learning process and environment.

REFERENCES

[1] M. Abdous, W. He, and C.-J. Yen, "Using data mining for predicting relationships between online question theme and final grade," *Edu. Technol. Soc.*, vol. 15, no. 3, pp. 77–88, 2012.

[2] J. W. Alstete and N. J. Beutell, "Performance indicators in online distance learning courses: a study of management education," *Quality Assurance Educ.*, vol. 12, no. 1, pp. 6–14, 2004.

[3] O. Ardaiz-Villanueva, X. Nicuesa-Chacón, O. Brene-Artazcoz, M. L. S. de Acedo Lizarraga, and M. T. S. de Acedo Baquedano, "Evaluation of computer tools for idea generation and team formation in project-based learning," *Comput. Educ.*, vol. 56, no. 3, pp. 700–711, 2011.

[4] K. C. Assi, H. Labelle, and F. Cheriet, "Modified large margin nearest neighbor metric learning for regression," *IEEE Signal Process. Lett.*, vol. 21, no. 3, pp. 292–296, Mar. 2014.

[5] R. S. Baker and P. S. Inventado, "Educational data mining and learning analytics," in *Learning Analytics*, J. A. Larusson and B. White, Eds. New York, NY, USA: Springer-Verlag, 2014, pp. 61–75.

[6] R. Barber and M. Sharkey, "Course correction: Using analytics to predict course success," in *Proc. 2nd Int. Conf. Learn. Anal. Knowl. (LAK)*, 2012, pp. 259–262.

[7] A. Becheru and E. Popescu, "Using social network analysis to investigate students' collaboration patterns in eMUSE platform," in *Proc. ICSTCC*, Oct. 2017, pp. 266–271.

[8] M. Bienkowski, M. Feng, and B. Means, "Enhancing teaching and learning through educational data mining and learning analytics: An issue brief," US Dept. Educ., Office Educ. Technol., Washington, DC, USA, 2012.

[9] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[10] C. Brooks and C. Thompson, "Predictive modelling in teaching and learning," in *Handbook of Learning Analytics*, C. Lang, G. Siemens, A. Wise, and D. Gašević, Eds. SOLAR, 2017, pp. 61–68. [Online]. Available: https://solaresearch.org/hla-17, doi: 10.18608/hla17.

[11] M. Brown. (2012). Learning analytics: Moving from Concept to Practice. EDUCAUSE Learning Initiative. [Online]. Available: https://library.educause.edu/~/media/files/library/2012/7/elib1203-pdf.pdf

[12] M. D. Calvo-Flores, E. G. Galindo, M. P. Jiménez, and O. P. Piñeiro, "Predicting students' marks from Moodle logs using neural network models," in *Current Developments in Technology-Assisted Education: General Issues, Pedagogical Issues*, vol. 1. Badajoz, Spain: Formatex, pp. 586–590, 2006. [Online]. Available: http://www.formatex.org/micte2006/book1.htm

[13] M. A. Chatti, A. L. Dyckhoff, U. Schroeder, and H. Thüs, "A reference model for learning analytics," *Int. J. Technol. Enhanced Learn.*, vol. 4, no. 5, pp. 318–331, 2012.

[14] M. Dascalu, E. Popescu, A. Becheru, S. Crossley, and S. Trausan-Matu, "Predicting academic performance based on students' blog and microblog posts," in *Adaptive and Adaptable Learning. EC-TEL* (Lecture Notes in Computer Science), vol. 9891. Cham, Switzerland: Springer, 2016, pp. 370–376.

[15] A. L. Dyckhoff, V. Lukarov, A. Muslim, M. A. Chatti, and U. Schroeder, "Supporting action research with learning analytics," in *Proc. Int. Conf. Learn. Anal. Knowl. (LAK)*, 2013, pp. 220–229.

[16] C. Ellis, "Broadening the scope and increasing the usefulness of learning analytics: The case for assessment analytics," *Brit. J. Educ. Technol.*, vol. 44, no. 4, pp. 662–664, 2013.

[17] R. Ferguson, "The state of learning analytics in 2012: A review and future challenges," Knowl. Media Inst., Open Univ., Milton Keynes, U.K., Tech. Rep. KMI-12-01, 2012.

[18] R. Ferguson *et al.*, "Research evidence on the use of learning analytics: Implications for education policy," R. Vuorikari and J. C. Muñoz, Eds., Joint Research Centre Science for Policy Report, EUR 28294 EN. [Online]. Available: https://ec.europa.eu/jrc/en/publication/eur-scientific-and-technical-research-reports/research-evidence-use-learning-analytics-implications-education-policy, doi: 10.2791/955210.

[19] D. Gaševic, S. Dawson, T. Rogers, and D. Gasevic, "Learning analytics should not promote one size fits all: The effects of instructional conditions in predicting academic success," *Internet Higher Educ.*, vol. 28, pp. 68–84, Jan. 2016.

[20] B. Giesbers, B. Rienties, D. Tempelaar, and W. Gijselaers, "Investigating the relations between motivation, tool use, participation, and performance in an e-learning course using Web-videoconferencing," *Comput. Hum. Behav.*, vol. 29, no. 1, pp. 285–292, 2013.

[21] W. Greller and H. Drachsler, "Translating learning into numbers: A generic framework for learning analytics," *Edu. Technol. Soc.*, vol. 15, no. 3, pp. 42–57, Jul. 2012.

[22] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *ACM SIGKDD Explorations Newslett.*, vol. 11, no. 1, pp. 10–18, 2009.

[23] W. Hämäläinen and M. Vinni, "Comparison of machine learning methods for intelligent tutoring systems," in *Proc. Int. Conf. Intell. Tutoring Syst.* Berlin, Germany: Springer-Verlag, 2006, pp. 525–534.

[24] R. Junco and C. Clem, "Predicting course outcomes with digital textbook usage data," *Internet Higher Educ.*, vol. 27, pp. 54–63, Oct. 2015.

[25] R. Junco, G. Heiberger, and E. Loken, "The effect of Twitter on college student engagement and grades," *J. Comput. Assist. Learn.*, vol. 27, no. 2, pp. 119–132, 2011.

[26] P. Kim, J.-S. Hong, C. Bonk, and G. Lim, "Effects of group reflection variations in project-based learning integrated in a Web 2.0 learning space," *Interact. Learn. Environ.*, vol. 19, no. 4, pp. 333–349, 2011.

[27] J. H. L. Koh, S. C. Herring, and K. F. Hew, "Project-based learning and student knowledge construction during asynchronous online discussion," *Internet Higher Educ.*, vol. 13, pp. 284–291, Dec. 2010.

[28] C. Lang, G. Siemens, A. Wise, and D. Gašević, Eds., *Handbook of Learning Analytics*. SOLAR, 2017. [Online]. Available: https://solaresearch.org/hla-17, doi: 10.18608/hla17.

[29] F. Leon and S. Curteanu, "Evolutionary algorithm for large margin nearest neighbour regression," in *Proc. 7th Int. Conf. Comput. Collective Intell. Technol. Appl.*, 2015, pp. 305–315.

[30] F. Leon and S. Curteanu, "Large margin nearest neighbour regression using different optimization techniques," *J. Intell. Fuzzy Syst.*, vol. 32, no. 2, pp. 1321–1332, 2017.

[31] F. Leon and E. Popescu, "Using large margin nearest neighbor regression algorithm to predict student grades based on social media traces," in *Proc. Methodologies and Intelligent Systems for Technology Enhanced Learning* (Advances in Intelligent Systems and Computing), vol. 617. Springer, 2017, pp. 12–19.

[32] J. Lu, C.-S. Yu, and C. Liu, "Learning style, learning patterns, and learning performance in a WebCT-based MIS course," *Inf. Manage.*, vol. 40, no. 6, pp. 497–507, 2003.

[33] L. P. Macfadyen and S. Dawson, "Mining LMS data to develop an 'early warning system' for educators: A proof of concept," *Comput. Educ.*, vol. 54, no. 2, pp. 588–599, 2010.

[34] Z. Papamitsiou and A. Economides, "Learning analytics and educational data mining in practice: A systematic literature review of empirical evidence," *Edu. Technol. Soc.*, vol. 17, no. 4, pp. 49–64, 2014.

[35] Z. Pardos, N. Heffernan, C. Ruiz, and J. Beck, "The composition effect: Conjuntive or compensatory? An analysis of multi-skill math questions in ITS," in *Proc. 1st Int. Conf. Educ. Data Mining*, 2008, pp. 147–156.

[36] Z. A. Pardos, N. T. Heffernan, B. Anderson, and C. L. Heffernan, "The effect of model granularity on student performance prediction using Bayesian networks," in *Proc. 11th Int. Conf. User Modeling*. Springer, 2007, pp. 435–439.

[37] A. Peña-Ayala, "Educational data mining: A survey and a data mining-based analysis of recent works," *Expert Syst. Appl.*, vol. 41, pp. 1432–1462, Mar. 2014.

[38] E. Popescu, "Providing collaborative learning support with social media in an integrated environment," *World Wide Web*, vol. 17, no. 2, pp. 199–212, 2014.

[39] E. Popescu and G. Badea, "Using CollAnnotator to analyze a community of inquiry supported by educational blogs-preliminary results," in *Data Driven Approaches in Digital Education. EC-TEL* (Lecture Notes in Computer Science), vol. 10474. Cham, Switzerland: Springer, 2017, pp. 580–583.

[40] D. Roberge, A. Rojas, and R. Baker, "Does the length of time off-task matter?" in *Proc. 2nd Int. Conf. Learn. Anal. Knowl. (LAK)*, 2012, pp. 234–237.

[41] C. Romero, P. G. Espejo, A. Zafra, J. R. Romero, and S. Ventura, "Web usage mining for predicting final marks of students that use Moodle courses," *Comput. Appl. Eng. Educ.*, vol. 21, no. 1, pp. 135–146, 2013.

[42] C. Romero, M. I. López, J.-M. Luna, and S. Ventura, "Predicting students' final performance from participation in on-line discussion forums," *Comput. Educ.*, vol. 68, pp. 458–472, Oct. 2013.

[43] C. Romero and S. Ventura, "Educational data mining: A review of the state of the art," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 40, no. 6, pp. 601–618, Nov. 2010.

[44] C. Romero, S. Ventura, P. G. Espejo, and C. Hervás, "Data mining algorithms to classify students," in *Proc. 1st Int. Conf. Educ. Data Mining*, 2008, pp. 8–17.

[45] M. Saqr, U. Fors, and M. Tedre, "How learning analytics can early predict under-achieving students in a blended medical education course," *Med. Teacher*, vol. 39, no. 7, pp. 757–767, 2017.

[46] T. Stafford, H. Elgueta, and H. Cameron, "Students' engagement with a collaborative Wiki tool predicts enhanced written exam performance," *Res. Learn. Technol.*, vol. 22, p. 22797, 2014. [Online]. Available: ://journal.alt.ac.uk/index.php/rlt/article/view/1491/html

[47] C. Steiner, M. Kickmeier-Rust, and M. A. Türker. (2014). *Review Article About LA and EDM Approaches*. [Online]. Available: http://css-kmi.tugraz.at/mkrwww/leas-box/downloads/D3.1.pdf

[48] M. Van Harmelen and D. Workman, "Analytics for learning and teaching," JISC, CETIS Analytics Series, Bolton, U.K., 2012, vol. 1, no. 3. [Online]. Available: http://publications.cetis.org.uk/2012/516

[49] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Feb. 2009.

[50] A. Wolff, Z. Zdrahal, A. Nikolov, and M. Pantucek, "Improving retention: Predicting at-risk students by analysing clicking behaviour in a virtual learning environment," in *Proc. 3rd Int. Conf. Learn. Anal. Knowl. (LAK)*, 2013, pp. 145–149.

[51] W. Xing, R. Guo, E. Petakovic, and S. Goggins, "Participation-based student final performance prediction model through interpretable genetic programming: Integrating learning analytics, educational data mining and theory," *Comput. Hum. Behav.*, vol. 47, pp. 168–181, Jun. 2015.

[52] J. Yoo and J. Kim, "Can online discussion participation predict group project performance? Investigating the roles of linguistic features and participation patterns," *Int. J. Artif. Intell. Educ.*, vol. 24, pp. 8–32, Jan. 2014.

[53] A. Zafra and S. Ventura, "Predicting student grades in learning management systems with multiple instance genetic programming," in *Proc. 2nd Int. Conf. Educ. Data Mining*, 2009, pp. 309–319.

[54] D. Zhang, L. Zhou, R. O. Briggs, and J. F. Nunamaker, Jr., "Instructional video in e-learning: Assessing the impact of interactive video on learning effectiveness," *Inf. Manage.*, vol. 43, pp. 15–27, Jan. 2006.

ELVIRA POPESCU (M'06) received the Ph.D. degree (double degree) in information and systems technologies from the University of Technology of Compiègne, France, and the University of Craiova, Romania, in 2009.

She has been a Faculty Member at the Computers and Information Technology Department, University of Craiova, since 2005, where she became a Full Professor in 2018. She has authored and co-authored over 100 publications, including two books, journal articles, book chapters, and conference papers. In addition, she has co-edited four journal special issues, as well as 11 international conference proceedings published by Springer and one published by the IEEE. She has participated in over 15 national and international research projects, three of which as a principal investigator (grant director). Her research interests include technology-enhanced learning, adaptation and personalization in Web-based systems, learner modeling, computer-supported collaborative learning, learning analytics, and intelligent and distributed computing.

Dr. Popescu also serves as the Vice Chair for the *IEEE Women in Engineering* Romania Section Affinity Group and is an Executive Board Member for the IEEE Technical Committee on Learning Technology and the International Association of Smart Learning Environments. She received several scholarships and awards, including four best paper distinctions (IEEE EUROCON 2007, IEEE ICALT 2013, ICWL 2015, and ICWL 2018). She is actively involved in the research community by participating in six journal editorial boards, organizing a series of international workshops in the area of social and personal computing for e-learning (SPeL 2008–2018), serving as a conference chair, program committee chair, and track chair for 12 conferences.

FLORIN LEON received the Ph.D. degree in computer science from the Gheorghe Asachi Technical University of Iasi, Romania, in 2005, followed by a Post-Doctoral Fellowship completed in 2007. In 2015, he defended his habilitation thesis.

He has been a Faculty Member at the Department of Computer Science and Engineering, Gheorghe Asachi Technical University of Iasi, since 2005, where he became a Full Professor in 2015. He has authored and co-authored over 140 journal articles, book chapters and conference papers, and 12 books. He was a member in the guest editorial boards for three journal special issues, and he participated in 28 national and international research projects, two of which as a principal investigator. His research interests include artificial intelligence, machine learning, multiagent systems, and software design.

Dr. Leon was a member of the organizing committee or program committee chair of five conferences. He is currently a member of the IEEE Systems, Man, and Cybernetics Society, Computational Collective Intelligence Technical Community.