SN Computer Science (2020) 1:323
https://doi.org/10.1007/s42979-020-00332-7

ORIGINAL RESEARCH

From Transcripts to Insights for Recommending the Curriculum to University Students

Thong Le Mai[1,2] · Minh Thanh Chung[3] · Van Thanh Le[1,2] · Nam Thoai[1,2]

Received: 29 May 2020 / Accepted: 14 September 2020 / Published online: 8 October 2020

Abstract

Student data play an important role in evaluating the effectiveness of educational programs in the universities. All data are aggregated to calculate the education criteria by year, region, or organization. Remarkably, recent studies showed the data impacts when making exploration to predict student performance objectives. Many methods in terms of data mining were proposed to be suitable to extract useful information in regards to data characteristics. However, the reconciliation between applied methods and data characteristics still exists some challenges. Our paper will demonstrate the analysis of this relationship for a specific dataset in practice. The paper describes a distributed framework based on Spark for extracting information from raw data. Then, we integrate machine learning techniques to train the prediction model. The experiments results are analyzed through different scenarios to show the harmony between the influencing factors and applied techniques.

Keywords Educational data mining · Prediction · Student performance · Machine learning · Distributed system · Spark

Introduction

Education data mining (EDM) is a research field which concerns data-mining techniques to analyze patterns from data in educational context [33]. Online learning systems such as Learning Management Systems (LMS) [6], Massive Open Online Courses (MOOCs) [25] have become more and more popular in higher education institutions with the advance of current technology. Sometimes, it is a requirement every student needs to participate as a course rule, an external factor such as a global pandemic where all universities are forced to close. As a result, educational data gathered from these systems expand more quickly in this day and age. Student's performance and their behaviors can be better understand when these data are thoroughly examined. Therefore, these findings can help in identifying students' risks to timely intervene, discover their hidden potentials, predict student's performance in the next semester, etc. Based on the literature review in 2013 [8], we consider two main groups: "Student Modeling" and "Decision Support Systems" in terms of EDM. Some widely used methods are regression and classification for predicting [9, 38], but other methods have also been used such as clustering and feature selection for exploring patterns or emphasizing the interesting features [10, 24].

Prediction has been one of the most attractive fields of EDM since 1995 [3]. Related studies usually exploit potential factors from the university's data [4] to build a prediction model such as GPA or student's performance. Various Machine Learning algorithms are used to solve these problems including Decision Tree, Random Forest, Regression, and Neural Network [9, 34, 42]. Some other techniques based on Recommendation System (e.g., Collaborative Filtering and Matrix Factorization) also found a lot of successes [17, 27–29, 36]. Furthermore, some studies focus on different types of predictor variables rather than the explicit ratings such as ages, sex, online time, and response efficiency in improving the accuracy [9, 13]. However, the

This article is part of the topical collection "SoftwareTechnology and Its Enabling Computing Platforms" guest edited byLam-Son Lê and Michel Toulouse.

✉ Minh Thanh Chung
minh.thanh.chung@ifi.lmu.de

Nam Thoai
namthoai@hcmut.edu.vn

[1] Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam

[2] Vietnam National University Ho Chi Minh City, Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam

[3] MNM-Team, Ludwig-Maximilians-Universitaet (LMU), Oettingenstraße 67, 80538 Munich, Germany

problems in collecting educational data are the scale of prediction models and the characteristic aware of each dataset applied to prediction methods.

In previous work [21, 22], we focused on the development and evaluation of our distributed framework based on Spark [39] to predict the performance of undergraduate students. The dataset was collected at Ho Chi Minh City University of Technology (HCMUT). Our previous work showed how to adapt prediction model to the data features. In this paper, we focus on evaluating the harmony between data and prediction techniques affecting to the accuracy. Simultaneously, the technical architecture of the framework is also introduced in detail.

The rest of this paper is organized as follows. The section "Related Work" shows related work about methods for predicting student performance. We describe specifically the dataset provided by Ho Chi Minh City University of Technology and problem definition in the section "Student Dataset and Problem Definition". The section "Implementation" describes the architecture of the proposed framework. The section "Experiment" presents the experimental scenarios, results, and then highlights the conclusions as well as future work in the section "Conclusion".

Related Work

In terms of Educational Data Mining (EDM), one of the most common tasks is to filter out information that can be used to predict the student's performance [2, 3, 33]. Generally, many studies have been conducted to predict student's grades as well as identify risky students using efficiently machine learning algorithms. Some of them are based on Recommendation System models [30] that can handle effectively with sparse data.

Romero et al. have applied classification algorithms such as Decision Tree, Rule Induction, and Neural Network to predict students' final marks by labeling their final mark as four categories: FAIL, PASS, GOOD, and EXCELLENT. The prediction model is built based on information extracted from an e-learning system [31, 34] such as the number of completed assignments, quizzes, and forum posts. The random forest method was employed in [41, 42] to examine the statistical relationship between students' graduate-level performance and undergraduate achievements. García et al. have applied the association rule mining to discover interesting information through students' usage data in the form of IF–THEN recommendation rules. The work's objective is to build a system helping teachers to continuously improve and maintain the adaptive and non-adaptive e-learning courses [15]. In other research, Nurjanah et al. proposed an approach for recommending a learning system that combines content-based filtering and collaborative filtering [29, 36]. In detail, content-based filtering is first applied to filter out relevant materials. Then, collaborative filtering is used to select good students. This technique aims to reduce the drawbacks of classic collaborative filtering which recommends materials based on the similarity between students and not take into account students' competence. The resulting model achieved an MAE [37] score of 0.96 for a scale of 1–10 and 0.73 for a scale of 1–5. In 2017, Iqbal et al. have applied and evaluated Collaborative Filtering, Matrix Factorization, and Restricted Boltzmann Machine (RBM) [20, 26] to predict student grades in a dataset which consists of 225 students and 24 courses with 1736 available grades and 3664 missing grades (grades are given in scale 0–4). They concluded that the RBM model gave the best result with 0.3 of RMSE, while the one of the Non-negative Matrix Factorization was 0.57 [17]. On the other hand, Conijn et al. analyzed 17 blended courses with 4,989 students using 23 predictor variables extracted from Moodle Learning Management Systems (LMS) [9], They found that there was a significant improvement in prediction when those grades are unavailable in the case of in-between assessment grades are available. Thus, the LMS data in this dataset are substantially smaller predictive values compared to the midterm grades.

Concerning another approach, Nguyen et al. proposed matrix factorization to predict student performance on the Knowledge Discovery and Data Mining Challenge 2010 dataset. They showed that matrix factorization could improve prediction results compared to the traditional regression methods such as logistic/linear regression [28]. Furthermore, in their follow-up paper [27], they extended the research using tensor-based factorization to take the temporal effect into account when predicting student performance. Feng et al. [13] improved the prediction accuracy for some traditional models which only use the correctness of the test questions. They have taken into account the advantage of the student–system interaction information that is not normally available in the traditional practice tests such as the time students take to answer questions and the time they take to correct an answer which they got wrong. For this reason, the models are shown to make better predictions than their traditional counterparts. Elbadrawy et al. attempted to use Personalized Multiregression and Matrix Factorization to forecast students' grades on in-class assessments [11, 12]. The results revealed that these methods could achieve a lower error rate than the traditional methods.

Furthermore, a lot of studies focus reviews on the existing types of educational systems and methods applied in EDM. Web mining is considered a prominent group of EDM [32], because many methods revolve around the analysis of logs of student–computer interaction. [33] examined three hundred published papers until 2009 grouped by task/category such as recommendation, predicting performance, detecting behavior, analysis, visualization, etc. [3] investigated what

some of the major trends are in EDM research. They found that in 43% papers which was examined in [32] published between 1995 and 2005 centered around relationship mining methods. However, in 2008 and 2009, relationship mining slipped to fifth place with only 9% papers. On the other hand, prediction, which was in second place between 1995 and 2005, moved to the dominant position in 2008–2009.

Student Dataset and Problem Definition

Student Data of Ho Chi Minh City University of Technology

The student database is usually recorded as the transcripts of each student annually. Traditionally, this kind of data would be used to evaluate student performance or efficiency of the education program year by year. Depending on each university, the student data can be organized in different ways. Our training and testing data set are extracted from one part of the whole university database. The data were collected from Ho Chi Minh City University of Technology (HCMUT) in Vietnam (from 2006 until 2017). We divided the dataset into groups of 14 faculties. There are in total 61271 undergraduate students with 2389 courses, as shown in Table 1. Each record includes 35 fields with detailed information about students such as grades in particular courses, but we focus on 4 main fields: student identification, name of faculty, course identification, and grades for the corresponding courses, as shown in Table 2.

In this context, we focus on predicting the final grade for uncompleted courses of students by observing the relationship between completed courses and uncompleted ones. Each record contains information about the grades of students for the corresponding courses. The grades are scaled from 0 to 10 by the *double* type. The grade distribution is shown in Fig. 1. The popular range of undergraduate grades is from 5 to 8.5, and the sparsity of the dataset is 0.9845. The sparsity is calculated by the following formula (1):

$$S = 1 - \frac{G}{N \cdot C}, \tag{1}$$

where, N, G, and C are the total number of student's grades, students, and courses, respectively. The more the value of S closes to 1, the data are sparser.

In detail, the information of almost faculties including the number of courses, students, or grades is shown in Table 3. Because the educational program will be updated every 4 years, thus choosing the period of training set also need to be considered. This helps to keep data extracted from similar program cycles.

Figure 2 shows the distribution of student's grades in only 2012 and 2014. Different from the overall distribution in Fig. 1, almost all grades are larger than 4.0. This opens some questions about how to use the data, which factors are really influential.

Problem Definition

Basically, universities often register student data according to the specific fields. However, not all fields could be used

Table 1 The statistics of the dataset

Number of faculties	14
Number of courses	2389
Number of students	61271
Number of student's grades	2270045
Sparsity	0.9845

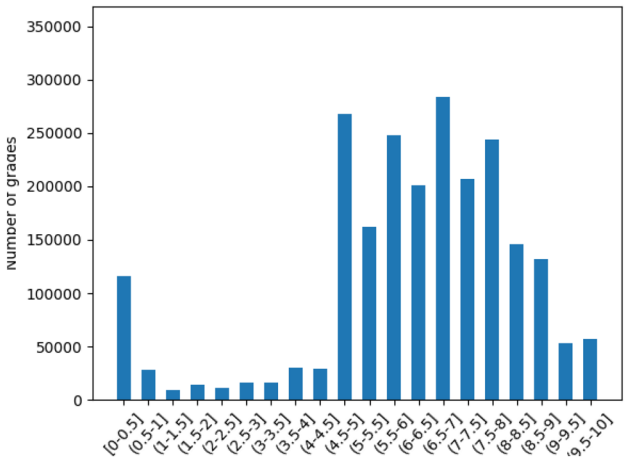


Fig. 1 Overall distribution of student's grades

Table 2 Educational dataset of a given student

Student	Faculty	Course	Grade
1228909	Applied Sciences	Solar energy	8.0
2973512	Electrical and Electronic Engineering	Computer Networks	4.5
2365234	Computer Science and Engineering	Operating Systems	8.0
3281723	Civil Engineering	Calculus	9.5
...	...	...	...

Table 3 The detail statistics of each faculty

Faculty	Notation	# courses	# students	# student's grades	Sparsity
Computer Science and Engineering	MT	168	5158	15,5574	0.8205
Industrial Maintenance	BD	116	1958	54,976	0.7580
Mechanical Engineering	CK	435	9233	351,539	0.9125
Geology & Petroleum Engineering	DC	207	2476	93,516	0.8175
Electrical and Electronic Engineering	DD	325	9391	360,546	0.8819
Transportation Engineering	GT	230	2323	88,510	0.8343
Chemical Engineering	HC	322	6117	222,478	0.8870
Environment and Natural Resources	MO	177	2401	90633	0.7867
Energy Engineering	PD	89	565	16,912	0.6637
Industrial Management	QL	137	3577	104,514	0.7867
Applied Sciences	UD	192	2099	78,986	0.8040
Materials Technology	VL	183	2910	109,612	0.7942
Training Program of Excellent Engineers in Vietnam (PFIEV)	VP	309	1515	92,040	0.8039
Civil Engineering	XD	445	11,691	450,209	0.9135

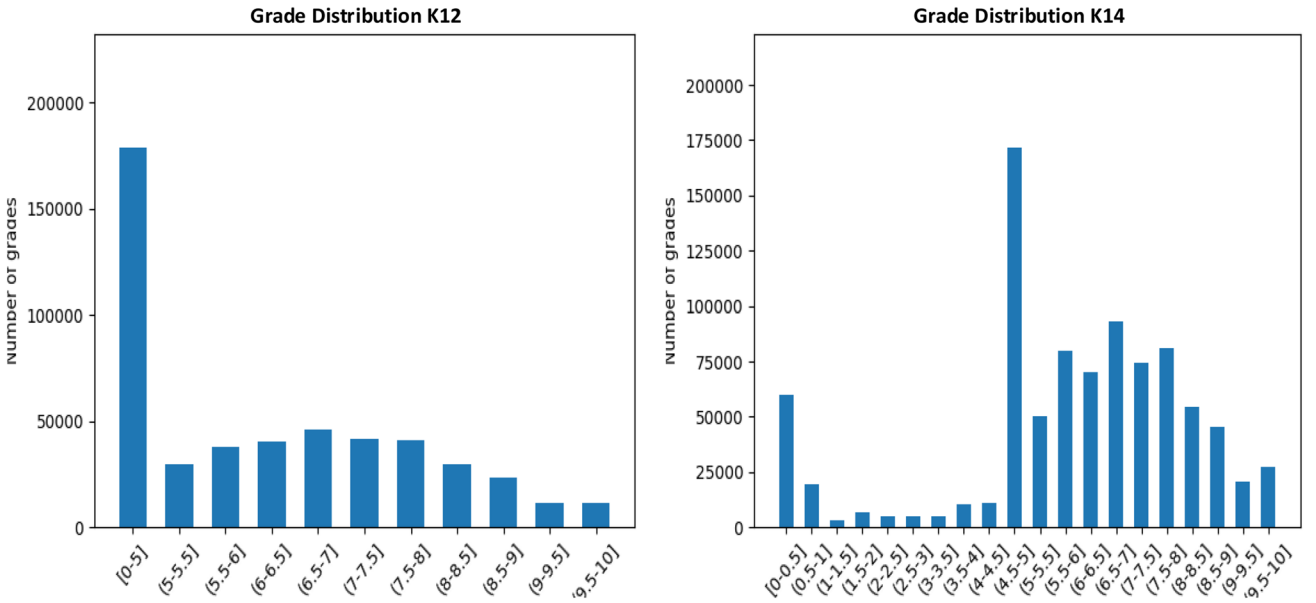


Fig. 2 Distribution of student's grades from 2012 and 2014

for analysis while the database consists of many different fields. In this section, we discuss the important factors that are drawn as the inputs of the training model. Specifically, these factors include:

- Student ID: this is the value to determine which student is being considered. Each student has 1 unique ID.
- Faculty: indicates the faculty that students are studying. This field affects the specific curriculum of each student. Because each department will have specific programs for teaching.
- Year: indicates the year that students are taking courses, also related to the semester in a school year. Normally, each faculty's training program will be changed or improved every 4 years.
- Number of courses (*Num_Courses*) and Subject: indicate the number of courses that a student has taken or must take. And which subjects (defined by subject code) students are enrolled in. Subjects are divided into 4 groups during the training period as Fig. 4 shows.

Finally, with these factors, a predictive model is expected to rely on certain inputs in producing predictive results. In Fig. 3, the predicted objective is student performance and the recommendation for upcoming specialized subjects. This model will be based on certain inputs including finished courses along with grades. For example, student A has studied subjects 1, 2, 3, ... together with the corresponding number of grades. According to the training program of the faculty, the prediction will recommend student A for the next subjects and the predicted grades which A can achieve. To this end, to achieve an accurate result in the model for predicting university-student results, the following questions will be evoked.

- Which Machine Learning algorithm should apply?
- Should the training model be divided by Year, Faculty, or a group of other related students in the same specialized program?
- What is the main factor affecting the accuracy of prediction models?

Our paper shows the level of influence and specific experiments from pre-processing influence factors to the accuracy of the predicted models (in the section "Experiment").

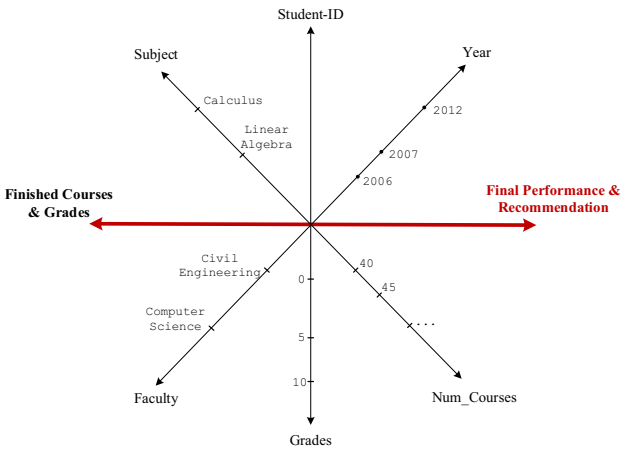


Fig. 3 Methodology with influential factors from the student data

Implementation

Framework Architecture

For handling big data analysis, the proposed framework includes 2 main blocks: offline and online as can be seen from Fig. 5. The offline block consists of modules as follows:

1. The raw data stored and summarized at the International Office would be updated year by year. The pre-processing module will extract and divide them into two datasets: Test-Set and Train-Set.
2. After that, the training module uses the Training-Set to build the prediction model. The validation module is integrated to evaluate and improve the model.

The result of the offline block is considered as the backend service of the online block. After training the module, the generated predictor cooperates with the online interaction. For instance, on the online-interaction module (web-based interface), students can create requests for predicting their future result with inputs such as the scores of given subjects. The inputs are sent to Predictor, and the predicted values would be displayed on the web interface and simultaneously transferred to the Recommender Module. This module uses association rules to guide students on what kind of subjects which they should choose.

Regarding the practical architecture based on Spark, Fig. 5 shows all modules from processing to Predictor and Recommender that are implemented as plugins of Spark [1]. Except for the web-based interface, it is used to interact with students to get the request. Similar to the operation model of Spark on a distributed system, the computation modules in training are assigned to workers. The master plays a role in controlling and scheduling tasks into workers. This helps our framework to process efficiently large-scale dataset in adequate time, because the framework can scale up by increasing the number of workers running in parallel. However, the data-transfer communication also needs to be considered.

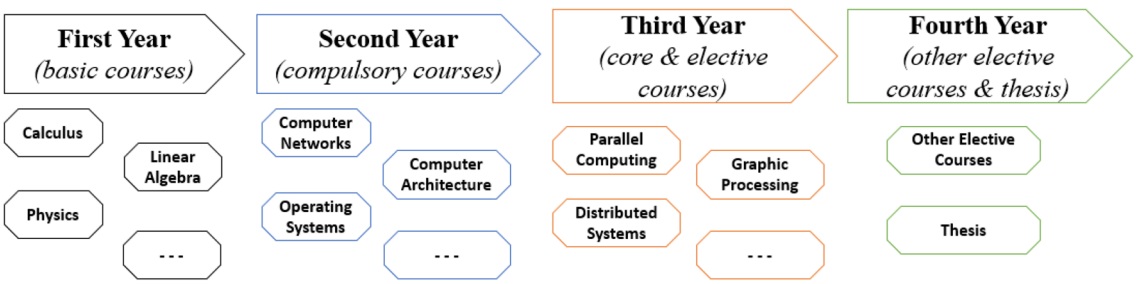


Fig. 4 Group of subjects by year in the university training system

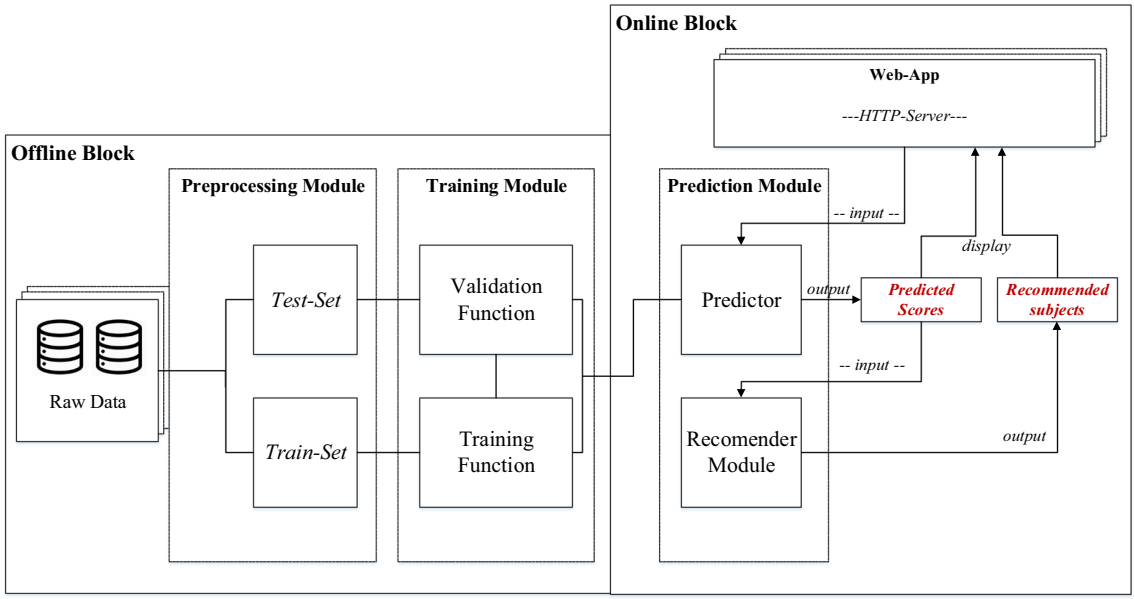


Fig. 5 The architecture and operation model of prediction framework

Underlying Machine Learning Techniques and Prediction Model

Collaborative Filtering

Collaborative Filtering is commonly used in recommendation systems [18]. It focuses on suggesting the set of items for users based on their history behaviors and the relationship between users to determine the user's rating for each item. In this paper, the users are students, and the items are courses associated with the user's ratings being grades. There are two kinds of Collaborative Filtering: User-Based Collaborative Filtering (UBCF) [40] and Item-Based Collaborative Filtering (IBCF) [35].

User-based Collaborative Filtering is performed by selecting and aggregating the grades of other students. There is a list of n students $S = \{S_1, S_2, ..., S_n\}$ and a list of m courses $C = \{C_1, C_2, ..., C_m\}$. Each student has a list of courses which represents student GPA. To predict the student's grades:

1. The UBCF algorithm calculates the similarity matrix to determine how similar each student in the database to the examined student is.
2. Then, the algorithm selects the most similar students by using k-nearest neighbors [14].
3. The prediction results are generated by aggregating the GPAs of the most similar students. In the simple case, the aggregation can be mean or weighted average by taking similarity between students into account.

Item-based Collaborative Filtering is used in the case that the courses have been rarely changed. This algorithm predicts the student's grade by identifying similar courses which have learned by the examined student. Instead of identifying the most similar students in UBCF, the IBCF algorithm determines the most similar courses from the set of courses that the current student have learned. The predictions are made by selecting and aggregating the grades of other courses.

1. The IBCF algorithm calculates the similarity matrix between the courses to determine how similar each course in the database to the course that needs to be predicted.
2. Then, the algorithm will select the most similar courses which are learned by the examined student based on using k-nearest neighbors.
3. Similar to UBCF, the prediction result is made by aggregating the GPAs of the most similar courses.

Matrix Factorization

Matrix Factorization [19] is the basis for some of the most successful realizations in the latent factor model which tries to characterize students and courses on k factors to explain the grades patterns. For courses, these factors can correspond to the amount of math, difficulty, and number of equations. For student, these factors correspond to the student affinity toward those latent factor. Matrix Factorization is a method that decomposes a matrix. In this case, it is the

utility matrix which represents all student's grades into two or more matrix. There are some variations and advancements for Matrix Factorization as follows.

Singular-Value Decomposition (SVD) tries to decompose the utility matrix G into two matrix, U and V [16]:

$$G \approx U \times V, \tag{2}$$

where U is a $m \times r$ matrix, where m is the number of students and r is the number of latent factors. Each student u is associated with vector p_u of the length r. Each element in this vector corresponds to the affinity of student u for the corresponding latent factor. Vector p_u can be viewed as a row in matrix U where U_{uk} represents the affinity of student u for the latent factor k. V is a $r \times n$ matrix, where n is the number of courses. V_{ik} represents the affinity of course i for the latent factor k. Each course i is associated with a vector q_i of the length r. Each element in this vector corresponds to the affinity of course i for the corresponding latent factor. Vector q_i can be viewed as a row in matrix V where V_{ik} represents the affinity of course i for the latent factor k. The dot product $p_u q_T^i$ will be the estimated grade $\hat{g}_{ui}$ of student u in course i:

$$\hat{g}_{ui} = p_u q_T^i. \tag{3}$$

To learn matrix U and V, we will minimize the cost function:

$$\sum_{u,i \in H} \left(r_{ui} - p_u q_i^T\right) + \lambda\left(p_u^2 + q_i^2\right), \tag{4}$$

where H is the set of (u, i) pair where g_{ui} is in the training set; λ is the regularization parameter. Using gradient descent, for each given value of g_{ui} in the training set, to update vector p_u and q_i:

$$\begin{aligned} p_u &= p_u + \gamma\left(\left(r_{ui} - p_u q_i^T\right) \cdot q_i - \lambda p_u\right) \\ q_i &= q_i + \gamma\left(\left(r_{ui} - p_u q_i^T\right) \cdot p_u - \lambda q_i\right), \end{aligned} \tag{5}$$

where γ is the learning rate.

Alternative Least Square (ALS) [5] is one of the optimization for the Singular-Value Decomposition (SVD) method. Recall Eq. (4) where both p_u and q_i are unknown and tied with each other in a multiplication operation which makes this non-convex. The idea of ALS is: when we fix one of the unknown variables which is either p_u or q_i, the cost function becomes a quadratic problem. In each iteration, ALS first fixes U (all vectors p_u) and solves for V, then it fixes V (all vectors p_i) and solves for U. The process is repeated until there is a convergence. In ALS, each p_u is independent with other $p_{u' != u}$, and each q_i is independent with other $q_{i' != i}$. This algorithm can be massively parallelized.

Non-negative Matrix Factorization is another type of matrix factorization, where the non-negative constraint is added. A given non-negative matrix G contains all observed grades, then we need to find the non-negative matrix factors, W and H [23]:

$$G^{(n)} \approx W \times H, \tag{6}$$

where, W is a non-negative $m \times r$ matrix and H is a non-negative $r \times n$ matrix. With normal matrix factorization, we can obtain negative affinity between a student u and a latent factor k which can be hard to interpret (e.g., the difficulty of latent factors can be negative). Non-negative matrix factorization (NMF) can give us a better representation of the latent factors by guaranteeing non-negative value. Especially in our problem, the course's grades are always larger or equal to 0. Moreover, NMF is better than matrix factorization in processing missing value or sparse data that our task must handle for student datasets.

Experiment

Testing Environment

Our experiments are performed on the cluster named SuperNode-XP which is a heterogeneous cluster with 24 compute nodes. There are 2 CPU sockets—Intel Xeon E5-2680 v3 @ 2.70 GHz, 2 Intel Xeon Phi 7120P (Knight Corners) cards, and 128GB RAM per node. The Spark cluster in this paper is built on 4 nodes: 1 master and 3 workers. In addition, the software stack is described as follows, Table 4.

Evaluation Scenarios

The dataset is divided separately into groups for training and testing the prediction model. Our work evaluates five different experiments conducted on the dataset:

1. **Baseline experiment**: Only use the data of a specific faculty to train and test the prediction for that faculty. This experiment is the base experiment to find out the problem when the data are not carefully chosen. The experiment is discussed clearly in the following comparison between Local & Global Locality analytics.
2. **Most-recent data experiment**: Use data of students enrolled at HCMUT from the academic year 2012. Because of the hypothesis, the most-recent data will

Table 4 Software specification on the testing environment

No.	Software	Description
1	Operating System	Red Hat Enterprise Linux 7.2
2	Spark	Apache Spark ver 2.4.0
3	Python	Version 2.7.15

affect the knowledge of the training model. As a result, the data of earlier years may have very different characteristics compared to recent years. This experiment aims to determine whether the change in time has a large impact on the performance of prediction models.

3. **Locality & Global**: For this context, the dataset is divided separately into groups for training and testing the prediction model. First is Locality Case (LC) which implies that each faculty has a separate prediction model that is trained by the local dataset. Second is Global Case (GC) which indicates a prediction model trained by the whole dataset. Then, it is used to predict scores for all faculties.
4. **Remove all zero grade experiment**: Remove all zero grade from the dataset. There are many reasons why students have zero grade as follows:

- Being absent in final exams.
- Being banned from taking the final exam.
- Missing too many course lectures.

Thus, the value of 0 may not reflect the true performance of these students in other different courses. Therefore, all zero grade will be removed in this experiment.

5. **Normalize failed grades and drop soft-skill/physical-education courses experiment**: In HCMUT, the students fail the course when their final grade is less than 5. We may label these fail grades as "Fail" (which evaluate to number 4 for calculating errors). Some soft-skill/physical training courses do not take into account the number of earn credits/GPA, but students must successfully achieve these courses before graduating. Students enrolled these courses just need to achieve the minimum grade of 5 to pass these courses and these courses have no impact on the final GPA of students. This experiment drops these course's grades, because they may not reflect the true technical skills and commitment of the student. Furthermore, we also only take data of students who enrolled in HCMUT from the year 2014 and above to further localize the data to observed whether carefully chosen data can greatly improve the accuracy of the predictions model.

Data Splitting and Prediction Algorithms

Students in the dataset will be split into 3 groups:

- **Train students**: occupy 60% number of students in the dataset. All data of these students will be used for the training models.
- **Validate students**: 20% students of the dataset. These student's data will be used for choosing the best parameter of prediction models.
  - Validation run: 50% data of the validate students will be used to train the model in the validation run. The remaining 50% will be used to evaluate the errors of prediction models.
  - Test run: All validation data will be used in the training phase.
- **Test students**: 20% students of the dataset. These student's data will only be used in the test run. 50% is used in training data of the prediction model and 50% is used to calculate the final errors of prediction models.

We proposed prediction models using the train sets, and then, we calculate Root Mean Square Error (RMSE), Mean Square Error (MSE), and Mean Absolute Error (MAE) [7, 37] on the test set. RMSE and MSE score will show us if there are any unusually high errors in some predictions. Meanwhile, MAE score gives a more average error in the experiment.

First, we will use grid search to obtain the best configuration for all models in each faculty. The search will use train-student data and validate student data. After that, each method is run and tested 10 times using the best configuration obtained from the previous grid search. Then, we get the average for the final results. In short, data usage is as follows:

- Grid search train data: (100% Train students data) + (50% Validate students data).
- Grid search test data: (50% Remaining Validate students data).
- Evaluation train data: (100% Train students data) + (100% Validate students data) + (50% Test students data).
- Evaluation test data: (50% Remaining Test student data).

We named seven different algorithms based on the two underlying methods, Collaborative Filtering and Matrix Factorization, for building the prediction model, as shown in Table 5. In this paper, the data in 2 faculties (MT and MO) are used to conducts the experiments. Information about these 2 datasets can be seen in Table 3.

Results

Baseline Experiment

The result of the baseline experiment is shown in Fig. 6. We can see that prediction errors in MT faculty are greater

Table 5 The detail information of proposed algorithms

Name	Algorithms
Baseline	Taking average of all visible grades on each student
IBCF	Item-based Collaborative Filtering
UBCF	User-based Collaborative Filtering
ALS	Alternative Least Square
ALS_NN	Alternative Least Square with non-negative constraint
ALS_NN_IBCF	Item-based Collaborative Filtering on Non-negative Alternative Least Square's Course Factor Matrix
ALS_IBCF	Item-based Collaborative Filtering on Alternative Least Square's Course Factor Matrix

than those of MO faculty even with the baseline model. For example, the best model in MT faculty (which is *ALS_NN* model) achieves an RMSE score of 1.69 much higher than *ALS_NN* model in MO faculty which has a RMSE score of 1.23. This behavior also persists across all other experiments. The predictions in MO faculty are more accurate than in MT faculty. Therefore, there are differences in the characteristics of the dataset between faculties.

Most-Recent Data Experiment

Similarly, we can see the result in Fig. 7. In this context, we only use data of students enrolled at HCMUT from 2012 and above. There is a significant drop in performance across all models compare to the previous experiment. *RMSE* errors of the baseline model in this experiment for MT and MO are 2.11 and 1.86, respectively. These errors are higher than their corresponding errors in the first experiments around 15%. From this experiment, the change of grade/subject characteristics in time affects the accuracy of prediction models. In terms of thickness and variety for the training data, we want to try getting data as much as possible, but the experiment shows that using the dataset from 2012 cannot bring better efficiency, because it covers 4.5-year programs and 4-year programs of our promotions.

Locality and Globality

Figure 8 shows the evaluation when we attempt to divide the dataset into local-set and global-set for training the prediction models. These two experiments (**LC** and **GC**) also show that using a big training dataset with all of the faculties does not affect much when comparing to the prediction models which are trained by a dataset from specific faculties. Figure 8-MT highlights that the predictor using the local-set with ALS_IBCF algorithm can reduce the error score from 2.10 to 1.83. Similar to other faculties, shrinking the dataset for training models combines with the used algorithm could reduce the error score as well as increase the accuracy, but this does not affect all cases. Faculty of Chemical Engineering (Fig. 8-HC) works well with the ALS algorithm and the trained

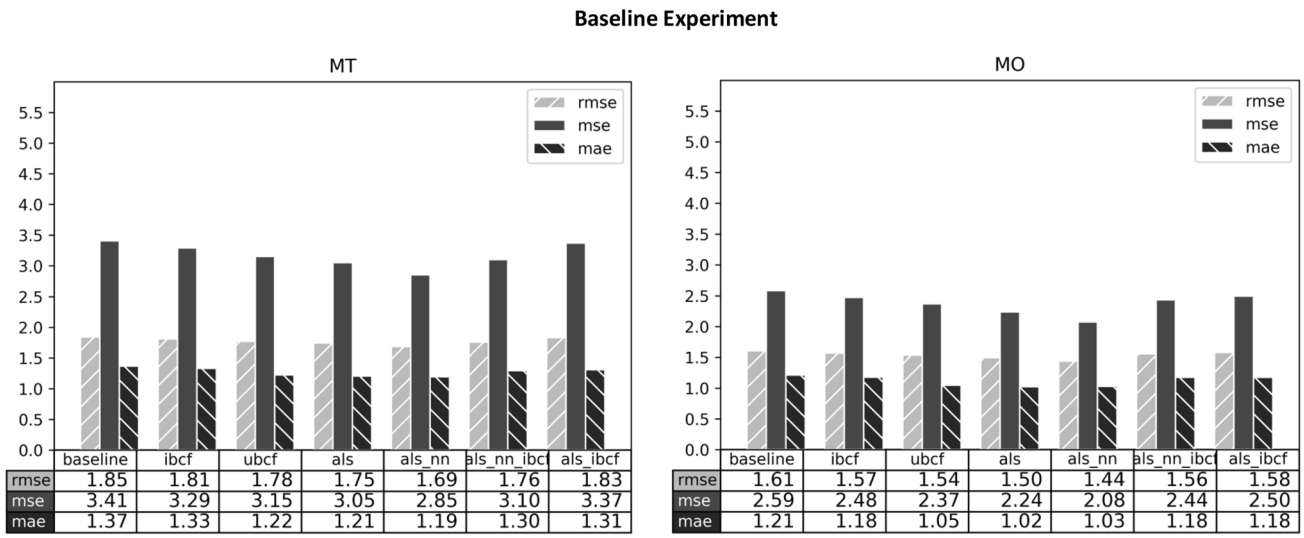
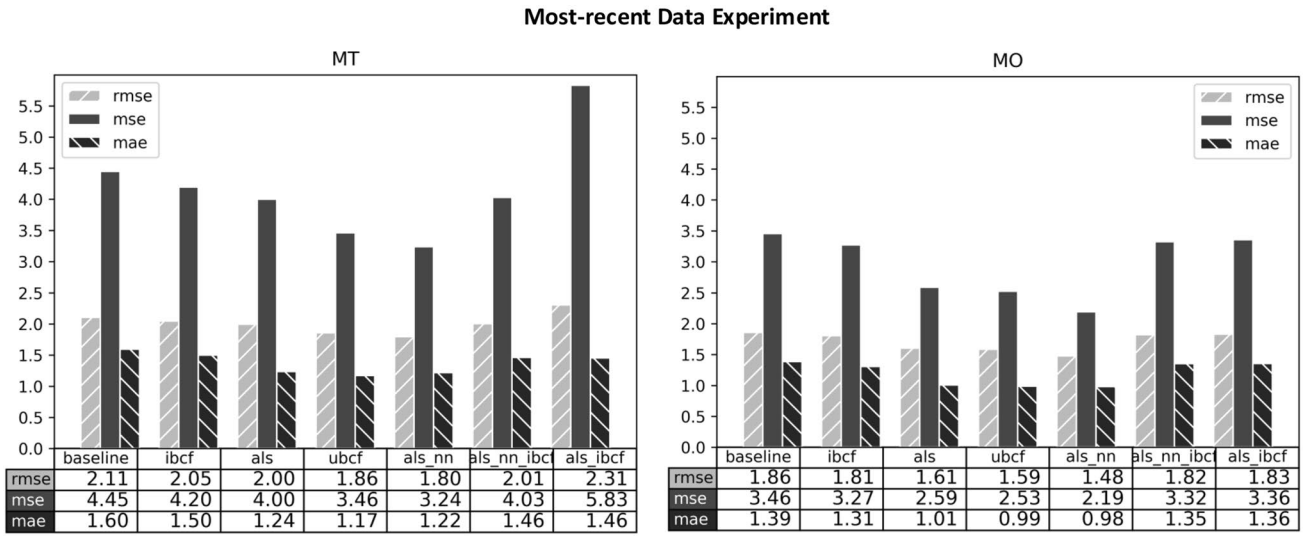


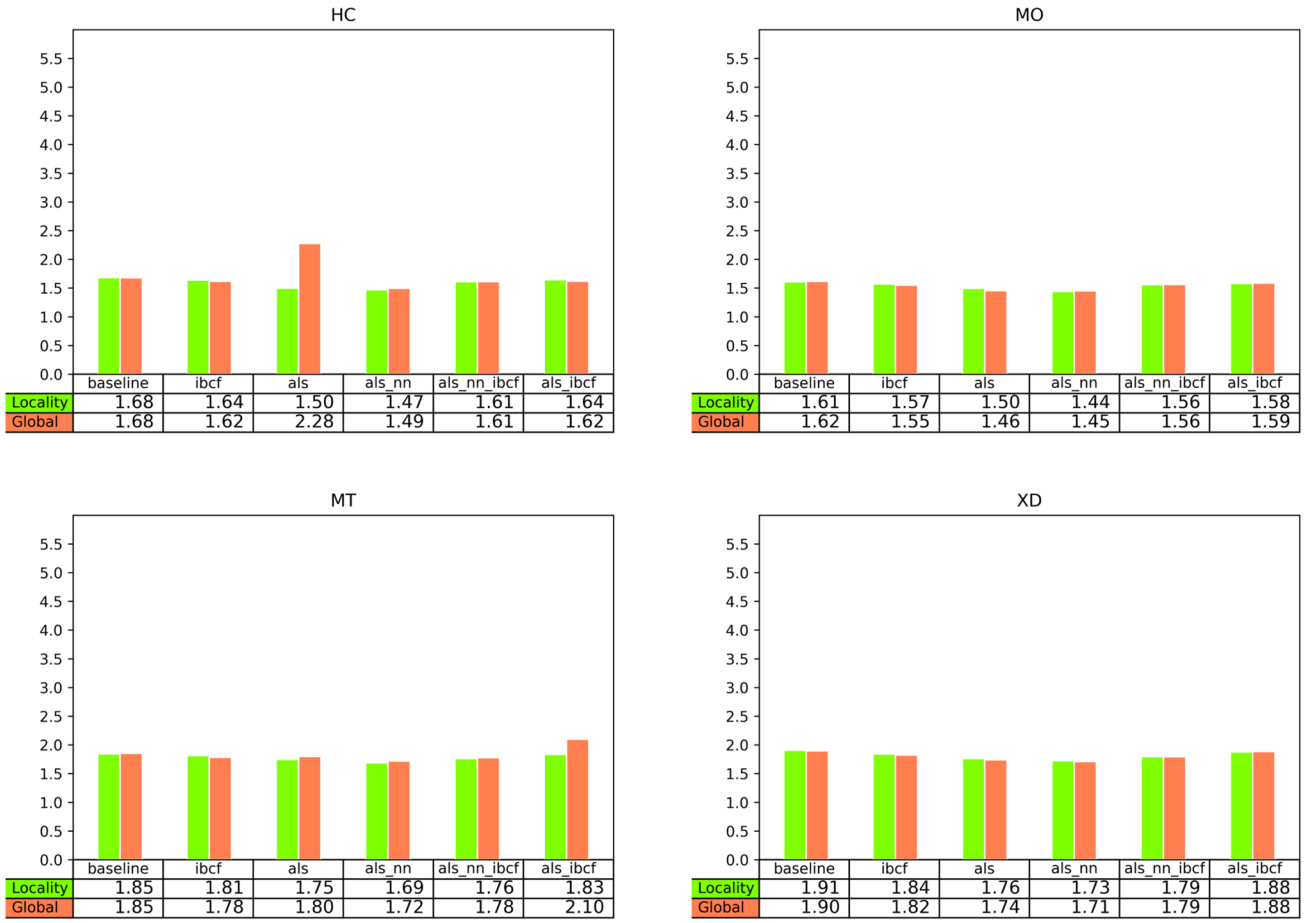
Fig. 6 Errors in basic experiment with MT and MO faculty's data



MT	baseline	ibcf	als	ubcf	als_nn	als_nn_ibcf	als_ibcf
rmse	2.11	2.05	2.00	1.86	1.80	2.01	2.31
mse	4.45	4.20	4.00	3.46	3.24	4.03	5.83
mae	1.60	1.50	1.24	1.17	1.22	1.46	1.46

MO	baseline	ibcf	als	ubcf	als_nn	als_nn_ibcf	als_ibcf
rmse	1.86	1.81	1.61	1.59	1.48	1.82	1.83
mse	3.46	3.27	2.59	2.53	2.19	3.32	3.36
mae	1.39	1.31	1.01	0.99	0.98	1.35	1.36

Fig. 7 Errors in most-recent data experiment with MT & MO faculty's data



HC	baseline	ibcf	als	als_nn	als_nn_ibcf	als_ibcf
Locality	1.68	1.64	1.50	1.47	1.61	1.64
Global	1.68	1.62	2.28	1.49	1.61	1.62

MO	baseline	ibcf	als	als_nn	als_nn_ibcf	als_ibcf
Locality	1.61	1.57	1.50	1.44	1.56	1.58
Global	1.62	1.55	1.46	1.45	1.56	1.59

MT	baseline	ibcf	als	als_nn	als_nn_ibcf	als_ibcf
Locality	1.85	1.81	1.75	1.69	1.76	1.83
Global	1.85	1.78	1.80	1.72	1.78	2.10

XD	baseline	ibcf	als	als_nn	als_nn_ibcf	als_ibcf
Locality	1.91	1.84	1.76	1.73	1.79	1.88
Global	1.90	1.82	1.74	1.71	1.79	1.88

Fig. 8 The evaluation in the influence of dataset locality to the predictions

local-set when the error score could be reduced from 2.28 to 1.5. Overall, this scenario emphasizes that the large dataset does not affect much on the accuracy of prediction models, especially in undergraduate student data (as Fig. 8-MO & XD).

Table 6 shows the experiment data in detail. Faculty MT and XD have a much higher based error than MO and HC faculties although HC has far higher sparsity (89%) and the number of courses (322) compared to MT which has sparsity of 82% and 168 courses. When running the experiment—GC of models without non-negative constraint, sometimes *RMSE* and *MSE* scores—would be much higher abnormally even though *MAE* scores are very consistent among all running trials. For example, the *ALS* model in the faculty of HC has a pretty bad *RMSE* score (2.28)—worse than the *RMSE* of the baseline model. However, with the *MAE* score, it gets a better *MAE* compared to the baseline model. This behavior is also seen in *IBCF* and *ALS* models when running the dataset of MT faculty. The MAE score is very close to *IBCF* but *RMSE* and *MSE* errors are very high. This experiment emphasizes that if we consider the local locality of student data, the training model can reduce dataset size and increase accuracy.

Remove All-Zero Grade Experiment

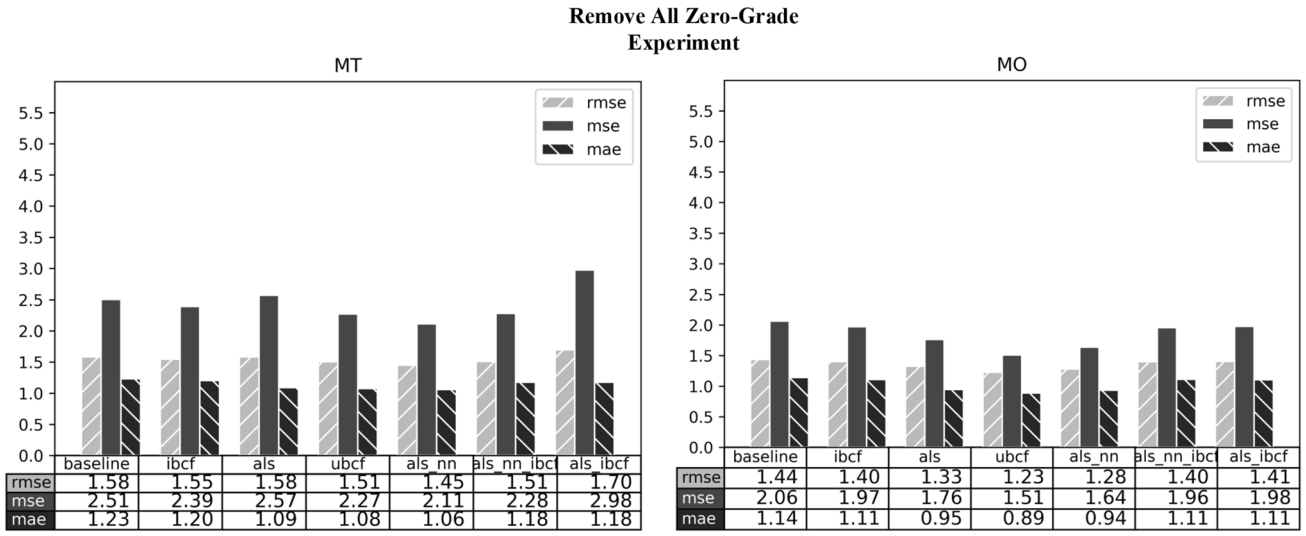
In the fourth experiment, the results are shown in Fig. 9, errors across all prediction models are greatly decreased in comparison with the first experiment. Another interesting observation is the result of the *UBCF* model with the MO dataset. In the first experiment, the *UBCF* model with MO's data only achieves the third-best RMSE score of 1.54. However, in this experiment, the result of the *UBCF* model is drastically improved and becomes the best prediction model for MO faculty with the RMSE score of 1.23. As a result, we can see that removing noise from the dataset brings the higher accuracy of all prediction models with each model has a different degree of improvement.

Normalize Failed Grades and Drop Soft-Skill/ Physical-Education Course Experiment

Finally, the last experiment shows the best result of all experiments. In this experiment, we will not use *ALS* and *ALS_IBCF*, because from the previous three experiments, these methods always performed worse than their non-negative counterpart—which is *ALS_NN* and *ALS_NN_IBCF* respectively. Figure 10 shows that *ALS_NN* models perform the best with RMSE of 1.22 in MT faculty data and 1.11 in

Table 6 Detail of experiments results

Faculty	Metric	Baseline	IBCF	UBCF	ALS	ALS_NN	ALS_NN_IBCF	ALS_IBCF
MT(LC)	RMSE	1.85	1.81	1.78	1.75	**1.69**	1.76	1.83
	MSE	3.40	3.29	3.15	3.05	**2.85**	3.10	3.37
	MAE	1.37	1.33	1.22	1.21	**1.19**	1.31	1.30
MO(LC)	RMSE	1.61	1.57	1.54	1.50	**1.44**	1.56	1.58
	MSE	2.59	2.48	2.37	2.24	**2.08**	2.44	2.50
	MAE	1.21	1.18	1.05	**1.02**	1.03	1.18	1.18
HC(LC)	RMSE	1.68	1.64	1.55	1.50	**1.47**	1.61	1.64
	MSE	2.82	2.69	2.41	2.25	**2.17**	2.60	2.71
	MAE	1.24	1.20	1.02	1.05	**1.01**	1.19	1.18
XD(LC)	RMSE	1.91	1.84	**1.73**	1.76	1.73	1.79	1.88
	MSE	3.64	3.39	2.99	3.11	**2.98**	3.22	3.54
	MAE	1.41	1.35	**1.20**	1.26	1.25	1.33	1.33
MT(GC)	RMSE	1.85	1.78	_	1.80	**1.72**	1.78	2.10
	MSE	3.44	3.18	_	3.24	**2.95**	3.16	4.64
	MAE	1.37	1.30	_	1.26	**1.22**	1.31	1.31
MO(GC)	RMSE	1.62	1.55	_	1.46	**1.45**	1.56	1.59
	MSE	2.61	2.41	_	2.12	**2.11**	2.44	2.53
	MAE	1.22	1.15	_	1.04	**1.04**	1.17	1.16
HC(GC)	RMSE	1.68	1.62	_	2.28	**1.49**	1.61	1.62
	MSE	2.82	2.61	_	7.36	**2.23**	2.60	2.63
	MAE	1.24	1.19	_	1.21	**1.05**	1.19	1.18
XD(GC)	RMSE	1.90	1.82	_	1.74	**1.71**	1.79	1.88
	MSE	3.60	3.32	_	3.03	**2.93**	3.21	3.56
	MAE	1.41	1.34	_	1.24	**1.24**	1.33	1.33



	baseline	ibcf	als	ubcf	als_nn	als_nn_ibcf	als_ibcf
rmse	1.58	1.55	1.58	1.51	1.45	1.51	1.70
mse	2.51	2.39	2.57	2.27	2.11	2.28	2.98
mae	1.23	1.20	1.09	1.08	1.06	1.18	1.18

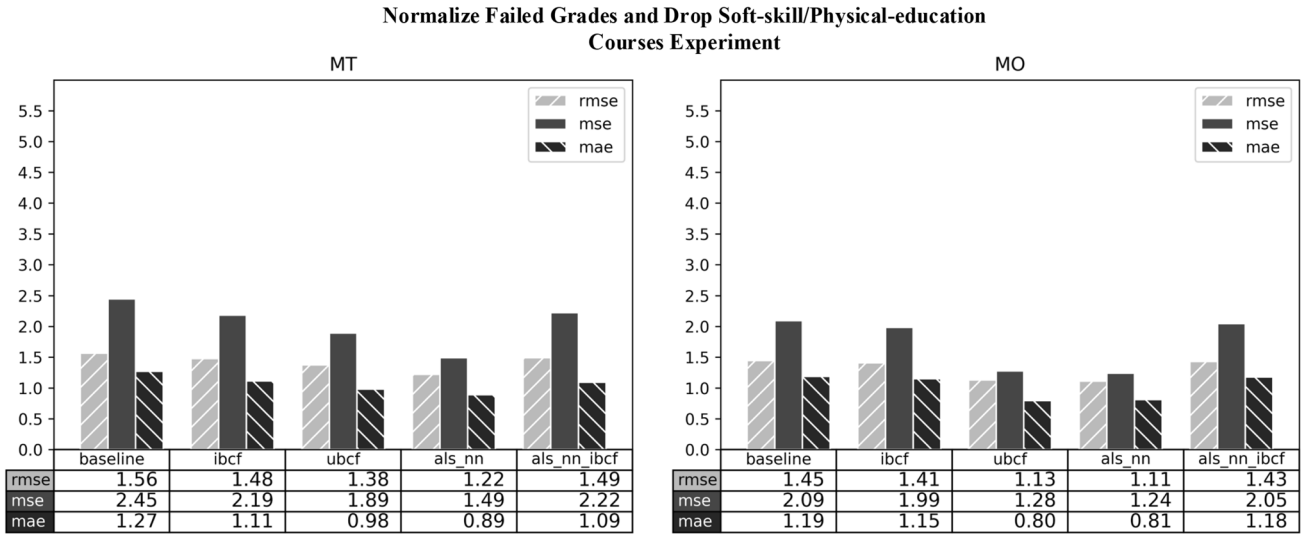
	baseline	ibcf	als	ubcf	als_nn	als_nn_ibcf	als_ibcf
rmse	1.44	1.40	1.33	1.23	1.28	1.40	1.41
mse	2.06	1.97	1.76	1.51	1.64	1.96	1.98
mae	1.14	1.11	0.95	0.89	0.94	1.11	1.11

Fig. 9 Errors in removing all zero-grades with MT and MO faculty's data

MO faculty data. These show the improvement of ≈ 38% in MT faculty and ≈ 30% in MO faculty. This experiment highlights that the courses of soft-skill or physical-education affect much on the accuracy of the prediction models and they are not necessary for the training process.

In summary, we draw the results of all experiments with ALS-NN method to demonstrate the relationship of dataset characteristic and the accuracy of student performance prediction models. As can be seen from Fig. 11, it is clear to note that when removing such noisy data as 0 or ungraded marks of soft skill, physical courses, the prediction models give the better results as RMSE of fourth and fifth experiments are significantly lower than those of other ones. Regarding the local and global locality, we will break down our dataset into smaller pieces presenting for distinct educational programs. That would help to increase the accuracy of the prediction models.

All experiments also illustrate that ICBF and ALS-ICBF are not much efficient as expected, but it is reasonable to have this results because ICBF based on course similarity. A educational program organize many courses to ensure that students achieve student outcomes after graduating, these courses may have the relationship, but are not much similar.



	baseline	ibcf	ubcf	als_nn	als_nn_ibcf
rmse	1.56	1.48	1.38	1.22	1.49
mse	2.45	2.19	1.89	1.49	2.22
mae	1.27	1.11	0.98	0.89	1.09

	baseline	ibcf	ubcf	als_nn	als_nn_ibcf
rmse	1.45	1.41	1.13	1.11	1.43
mse	2.09	1.99	1.28	1.24	2.05
mae	1.19	1.15	0.80	0.81	1.18

Fig. 10 Errors in normalize failed grades and drop soft-skill or physical-education courses experiment with MT and MO faculty's data

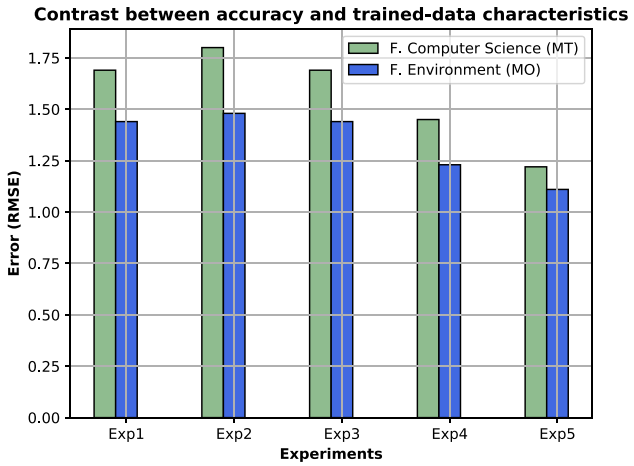


Fig. 11 Comparing the accuracy of 5 experiments with the results of the algorithm yielded the lowest error (ALS-NN), respectively

Conclusion

The paper shows aspects that affect the problem of predicting students' grades based on both application techniques and data characteristics. This is analyzed as a story from transcripts to insights for predicting student performance. Specifically, traditional university data are stored in various schools with depository purposes. We extract from the data set important parameters to prepare the training and testing dataset for specific experiments. Thereby, we build a framework for automatically analyzing and training models to predict student performance presented by predicted values of the recommended subjects in the future. In terms of practical applications, this is a framework that brings benefits in training orientation for students. Technically, we have built a framework based on Spark to be able to process large-scale dataset problem. By evaluating 5 experiments, our paper shows that finding influential factors or aspects plays an important role in the accuracy of prediction problems. In the scope of this paper, with the data set from HCMUT, we highlight that eliminating noise grades and unnecessary subjects could improve the efficiency of the framework.

Acknowledgements This research was conducted within the "*Studying Tools to Support Applications Running on Powerful Clusters & Big Data Analytic (HPDA phase I 2018–2020)*" funded by Ho Chi Minh City Department of Science and Technology (under grant number 46/2018/HD-QKHCN). We acknowledge the support of time and facilities from Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for this study.

Funding Open Access funding enabled and organized by Projekt DEAL.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Funding This study was funded by Ho Chi Minh City Department of Science and Technology (Under Grant Number 46/2018/HD-QKHCN).

References

1. Armbrust M, Xin RS, Lian C, Huai Y, Liu D, Bradley JK, Meng X, Kaftan T, Franklin MJ, Ghodsi A et al. Spark sql: relational data processing in spark. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, pages 1383–1394, Melbourne, Victoria, Australia, May 31–June 4 2015. ACM.
2. Asif R, Merceron A, Ali SA, Haider NG. Analyzing undergraduate students' performance using educational data mining. Comput Educ. 2017;113:177–194 .
3. Baker RSJD, Yacef K. The state of educational data mining in 2009: a review and future visions. JEDM J Educ Data Mining. 2009;1(1):3–17.
4. Benkwitz A, Parkes S, Bardy H, Myler K, Peters J, Akhtar A, Keeling P, Preece R, Smith T. Using student data: student-staff collaborative development of compassionate pedagogic interventions based on learning analytics and mentoring. J Hosp Leisure Sport Tour Educ. 2019;25:100202.
5. Bokde D, Girase S, Mukhopadhyay D. Matrix factorization model in collaborative filtering algorithms: a survey. Procedia Comput Sci. 2015;49:136–46.
6. Cavus N, Zabadi T. A comparison of open source learning management systems. Procedia Soc Behav Sci. 2014;143:521–6.
7. Chai T, Draxler RR. Root mean square error (rmse) or mean absolute error (mae)? Arguments against avoiding rmse in the literature. Geosci Model Dev. 2014;7(3):1247–50.
8. Chrysafiadi K, Virvou M. Student modeling approaches: a literature review for the last decade. Expert Syst Appl. 2013;40(11):4715–29.
9. Conijn R, Chris SA, Kleingeld UM. Predicting student performance from lms data: a comparison of 17 blended courses using moodle lms. IEEE Trans Learn Technol. 2016;10(1):17–29.
10. De Morais AM, Araujo JMFR, Costa EB. Monitoring student performance using data clustering and predictive modelling. In Proceedings of The 2014 IEEE Frontiers in Education Conference (FIE) Proceedings, pages 1–8, Madrid, Spain, October 2014. IEEE.

11. Elbadrawy A, Polyzou A, Ren Z, Sweeney M, Karypis G, Rangwala H. Predicting student performance using personalized analytics. Computer. 2016;49(4):61–9.
12. Elbadrawy A, Studham RS, Karypis G. Collaborative multi-regression models for predicting students' performance in course activities. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, LAK '15, page 103–107, Poughkeepsie, New York, March 2015. ACM.
13. Feng M, Heffernan N, Koedinger K. Addressing the assessment challenge with an online system that tutors as it assesses. User Model User-Adap Inter. 2009;19(3):243–66.
14. Fukunaga K, Narendra PM. A branch and bound algorithm for computing k-nearest neighbors. IEEE Trans Comput. 1975;100(7):750–3.
15. García E, Romero C, Ventura S, De Castro C. An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering. User Model User-Adap Inter. 2009;19(1–2):99–132.
16. Golub GH, Reinsch C. Singular value decomposition and least squares solutions. In Linear Algebra, 1971;volume 2, pages 134–151. Springer.
17. Iqbal Z, Qadir J, Adnan NM, Faisal K. Machine learning based student grade prediction: a case study; 2017.
18. Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 426–434, Las Vegas, Nevada, USA, August 2008. ACM.
19. Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. Computer. 2009;42(8):30–7.
20. Larochelle H, Mandel M, Pascanu R, Bengio Y. Learning algorithms for the classification restricted boltzmann machine. J Mach Learn Res. 2012;13(1):643–69.
21. Mai TL, Do Phat T, Chung MT, Thoai N. An apache spark-based platform for predicting the performance of undergraduate students. In Proceedings of the 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pages 191–199, Zhangjiajie, China, August 2019. IEEE.
22. Mai TL, Do Phat T, Chung MT, Thoai N, et al. Adapting the score prediction to characteristics of undergraduate student data. In Proceedings of The 2019 International Conference on Advanced Computing and Applications (ACOMP), pages 70–77, Nha Trang, Vietnam, November 2019. IEEE.
23. Lee DD, Seung HS. Algorithms for non-negative matrix factorization. In Proceedings of the 13th International Conference on Neural Information Processing Systems, NIPS'00, pages 535–541, Denver, CO, January 2000; MIT Press.
24. Li Z, Shang C, Shen Q. Fuzzy-clustering embedded regression for predicting student academic performance. In Proceedings of The 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pages 344–351, Vancouver, BC, Canada, July 2016; IEEE.
25. Margaryan A, Bianco M, Littlejohn A. Instructional quality of massive open online courses (moocs). Comput Educ. 2015;80:77–83.
26. Marlin Benjamin, Swersky Kevin, Chen Bo, Freitas Nando. Inductive principles for restricted boltzmann machine learning. In *Proceedings of The Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 509–516, Sardinia, Italy, May 2010; JMLR Workshop and Conference Proceedings.
27. Nguyen T-N, Drumond L, Horváth T, Nanopoulos A, Schmidt-Thieme L. Matrix and tensor factorization for predicting student performance. In Proceedings of The 3rd International Conference on Computer Supported Education (CSEDU), pages 69–78, Noordwijkerhout, Netherlands, May 2011; SCITEPRESS.
28. Nguyen T-N, Drumond L, Krohn-Grimberghe A, Schmidt-Thieme L. Recommender system for predicting student performance. Procedia Comput Sci. 2010;1(2):2811–9.
29. Nurjanah D. Good and similar learners' recommendation in adaptive learning systems. In Proceedings of the 8th International Conference on Computer Supported Education (CSEDU), pages 434–440, Rome, Italy, April 2016; SCITEPRESS.
30. Resnick P, Varian HR. Recommender systems. Commun ACM. 1997;40(3):56–8.
31. Romero C, López M-I, Luna J-M, Ventura S. Predicting students' final performance from participation in on-line discussion forums. Comput Educ. 2013;68:458–72.
32. Romero C, Ventura S. Educational data mining: a survey from 1995 to 2005. Expert Syst Appl. 2007;33(1):135–46.
33. Romero C, Ventura S. Educational data mining: a review of the state of the art. IEEE Trans Syst Man Cybern Part C (Appl Rev). 2010;40(6):601–18.
34. Romero C, Ventura S, Espejo PG, Hervás C. Data mining algorithms to classify students. In Proceedings of The 1st International Conference on Educational Data Mining, pages 8–17, Montreal, Quebec, Canada, June 2008. The International Educational Data Mining Society.
35. Sarwar B, Karypis G, Konstan J, Riedl J. Item-based collaborative filtering recommendation algorithms. In Proceedings of The 10th international conference on World Wide Web, pages 285–295, Hong Kong, May 2001. ACM.
36. Turnip R, Nurjanah D, Kusumo DS. Hybrid recommender system for learning material using content-based filtering and collaborative filtering with good learners' rating. In Proceedings of the 2017 IEEE Conference on e-Learning, e-Management and e-Services (IC3e), pages 61–66, Sarawak, Malaysia, November 2017. IEEE.
37. Willmott CJ, Matsuura K. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. Clim Res. 2005;30(1):79–82.
38. Xu J, Moon KH, Van Der Schaar M. A machine learning approach for tracking and predicting student performance in degree programs. IEEE J Select Top Signal Process. 2017;11(5):742–53.
39. Zaharia M, Chowdhury M, Franklin MJ, Shenker S, Stoica I. Spark: Cluster computing with working sets. In Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing, HotCloud'10, page 10, Boston, MA, USA, June 2010. USENIX Association.
40. Zhao Z-D, Shang M-S. User-based collaborative-filtering recommendation algorithms on hadoop. In Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, pages 478–481, Phuket, Thailand, January 2010; IEEE.
41. Zimmermann J, Brodersen KH, Heinimann HR, Buhmann JM. A model-based approach to predicting graduate-level performance using indicators of undergraduate-level performance. J Educ Data Mining. 2015;7(3):151–76.
42. Zimmermann J, Brodersen KH, Pellet J-P, August E, Buhmann JM. Predicting graduate-level performance from undergraduate achievements. In Proceedings of the 4th International Conference on Educational Data Mining, pages 357–358, Eindhoven, Netherlands, July 2011. Eindhoven University of Technology.