A Comparative Study of Student Performance Prediction using Pre-Course Data

Budor Alharbi[a,b,c], Fatima Assiri[b], and Basma Alharbi[a]

[a]Department of Computer Science and Artificial Intelligence, College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia

[b]Department of Software Engineering, College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia

[c]Department of Computer Science and Artificial Intelligence, College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia BALHARBI0266.stu@uj.edu.sa,fyassiri@uj.edu.sa,bmalharbi@uj.edu.sa

KEYWORD

Student performance prediction; machine learning; matrix factorization

ABSTRACT

Students at Saudi universities face difficulty registering for the right course since there is no support offered to students that uniquely consider each situation. Machine learning techniques could be applied to fill this gap by predicting grades of new courses for each student based on their historical data. This paper experiments with nine different prediction algorithms to predict course grades for public university students'. The data-set includes grades for 215 students and 180 various courses. The models utilize grades obtained in semesters between the 2015 and 2018 academic years and evaluated on grades obtained in the 2019 academic year. Our result shows that the K-nearest neighbor with ZScore model outperforms the remaining models with respect to the Percentage of Tick Accuracy (PTA), which is the difference difference between two consecutive letter grades for the predicted letter grade and the observed letter grade. Our work achieved an 84% accuracy score in PTA2, where the difference between predicted letter grade and the actual letter grade is less than or equal to two consecutive letter grades.

1. Introduction

Many students have suffered from completing courses in several higher education institutes since there is no devoted assistance for those who need exceptional help identifying the appropriate course for the next semester by avoiding courses that may delay their graduation. Thus, this study investigates the possibility of using machine learning in solving this problem that hinders students from getting high grades.

Machine learning plays a vital role in improving the education sector by providing different smart solutions to predict learner performance, which helps academic management and students to make right decisions efficiently. Early assessment can guide decision-making at different levels (e.g., ministry, regional academies, provincial directorates, and institutions) to plan their budgets, capacities, staff hiring, etc. For universities, student grade

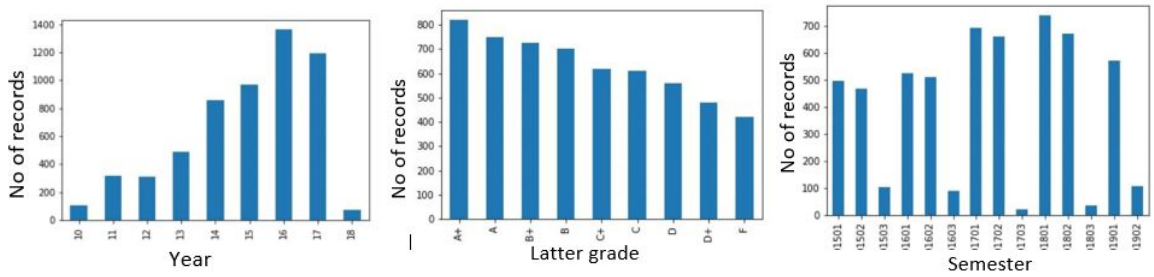
©Ediciones Universidad de Salamanca / cc by-nc-nd

prediction techniques can help plan the upcoming courses and select them beforehand for the forthcoming semesters. Furthermore, predicting academic results in advance can help monitor students' progress and avoid the risk of student's failure to continue their education.
There is some existing research investigated student's performance in the education sector. Recommended system that depends on the representation of grade and the combination of course and grade predictions were proposed (Morsy and Karypis, 2019). On the other hand, supervised learning algorithms were used to predict student's academic statas as Fail or Pass (Buenaño-Fernández et al., 2019). In other study, the performance of different machine learning algorithms were compared, and they found that Collaborative Filtering (UBCF algorithm), Matrix Factorization Singular Value Decomposition, and Non-negative Matrix Factorization, and Restricted Boltzmann Machines (RBM) is more accurate one (Iqbal et al., 2017). Machine learning was also utilized to build a recommend system that used matrix factorization and linear regression model for grade prediction (Polyzou and Karypis, 2016). Additive latent affect (ALE) along with matrix factorization (MF) were used to build a student grade prediction model (Ren et al., 2018). In (Acharya and Sinha, 2014), authors have introduced students' predictions of students' performance using machine learning techniques by studying a set of attributes. Furthermore, Genetic programming algorithms was also introduced to predict if they will fail or pass a particular course (Zafra and Ventura, 2009). However, there is limited research on investigating the impact of machine learning on students' grades in Saudi Arabia. This study aims to further investigate applying different prediction algorithms to predict students' grades for public University in jeddah. We will apply the most popular prediction algorithms utilized in the literature, which are Singular Value Decomposition(SVD) and Non-negative Matrix Factorization (NMF). In the purpose of produce a comparative study, we included seven prediction algorithms in addition to SVD and NMF, those prediction algorithms are Singular Value Decomposition with implicit grades(SVDpp), Slope One, BaselineOnly, NormalPredictor, K-Nearest Neighbor (KNN), K-Nearest Neighbor with ZScore (KNNWithZScore), and CoClustering algorithm.

Our dataset is collected from public University in jeddah for 215 students and 180 different courses in the semesters of 2015 and 2018 academic years. The prediction algorithms have been evaluated to predict the students' grades for the academic year of 2019. The obtained results found that that the K-nearest neighbor with ZScore (KNNWithZScore) model outperformed the remaining models in term of Percentage of Tick Accuracy (PTA), and achieved an 84% accuracy score in terms of PTA2.

©Ediciones Universidad de Salamanca / cc by-nc-nd

2. Background



(a) Distribution of records per entry year. *(b) Distribution of records per letter grade* *(c) Distribution of records per semester.*

Figure 1: Visualization of different distributions in the dataset.

2.1 Data Description

	All (Before data cleaning)	All (after data cleaning)	Training (%)	Testing (%)
No. of Records	7770	5683	5006 (88%)	677 (12%)
No. of Students	215	213	213	136
No. of Courses	180	51	51	27
Years covered	2015-2019	2015-2019	2015-2018	2019

Table 1: Dataset Summary Statistics.

We apply our proposed methodology, detailed in Section 4 on a real dataset that was collected from a public university in Saudi Arabia. The collected data spanned a period of five years, starting from 2015 to 2019. The dataset contains records from one bachelor program only, namely industrial engineering. Each record in the dataset describes one student-course enrollment. A record contains details such as the student ID, course ID, semester number, teacher ID, and course grade. Table 1 shows summary statistics of the collected dataset.

Figure 1 visualizes different distributions from the data. Figure 1-a shows the number of records per entry year. From the figure, it can be noted that the distribution of records per entry year does not follow a uniform distribution. Recall, our dataset collected records from 2015 to 2019. The per entry year distribution shows that our collected dataset has few records for students who joined the university on 2010, and similarly for those

©Ediciones Universidad de Salamanca / cc by-nc-nd

who joined on 2018. This is reasonable as students of 2010, would have few courses left to finish their 5-year undergraduate program. Similarly, students of 2018 have few records since enrollment.

Figure 1-b shows the distribution of records per letter grade. As shown in the figure, the records are uniformly distributed in grades B and above. Yet, the number of records per letter grade F is significantly less than A+. Lastly, Figure 1-c shows the distribution of records per semester. This figure shows that the number of records in the third semester (summer semester) in each academic year represents only 5% of the data.

2.2 Problem Definition

The raw dataset, described in the previous section, is mapped into an $m \times n$ (m=213 ,n=51) matrix R, where m is the number of students after cleaning and n is the number of courses after cleaning. In this matrix, rows represent students, and columns represent courses. Table 2 provides a summary of the used notations.

We formalize student grade prediction as a regression problem, where both the input and output are numerical values. The input for the regression algorithms will be R with grades reported before the first semester in 2019. Figure 2 illustrates the input and output of our problem. Let Matrix 1 in Figure 2 represents the reported grades for each student in the registered course. The cell r_{ij} represents the grade reported for student i in course j. The cells with zeros such as $r_{2,1}$ shows that the student (student ID 2) did not study the course number 101. The orange cells represent the grades reported on or after the first semester in 2019. The blue cells represent the grades reported before the first semester in 2019.

In matrix 2, we replace the grades in orange cells in matrix 1 with zeros to be utilized in the evaluation stage (Error table). Matrix 2 represents the regression algorithm's input. The input matrix (matrix 2) will be pass to several regression algorithms to predict the student grades for all courses. Matrix 3 represents the regression algorithm's output, which predicts grades for all cells. For example, $\hat{r}_{1,1}$ represents the predicted grade which is 92 for student id 1 and course number 1. However the actual grade for student id 1 in course number 1 is 87 as shown in matrix 1. In the evaluation stage, different evaluation metrics will be calculated based on an Error table that calculates the difference between actual grades (from matrix 1) and predicted grades (from matrix 3) for grades reported on or after the first semester in 2019.

©Ediciones Universidad de Salamanca / cc by-nc-nd

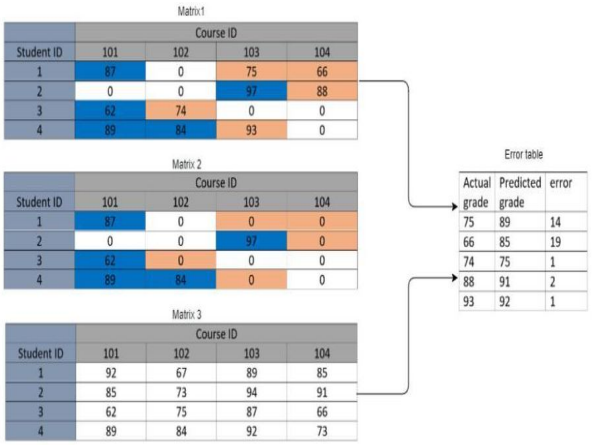


Figure 2: Example of input and output matrices for our experiment

R	is an $m \times n$ student by course matrix
m	Number of students
n	Number of courses
r_{ij}	is the predicted grade for i student and j course
$\hat{r}_{ij}$	is the predicted grade for i student and j course

Table 2: List of Notations

3. Related Work

Several studies have applied machine learning and deep learning in the education domain to predict grades of undergraduate students. Grade predictions can assist during course selection process to ensure that the student will be able to successfully complete the degree requirements.
(Morsy and Karypis, 2019) proposed a recommendation system using two different approaches, which are grade-aware representation learning approaches and combining course recommendation with grade prediction. Dataset has been collected from the University of Minnesota, which covers 16 years (Fall 2002 to Summer 2017) and includes students' data from 23 different majors. The first approach combined two methods; the first method is one-to-one relationship between previous and subsequent courses that applied Singular Value Decomposition

©Ediciones Universidad de Salamanca / cc by-nc-nd

(SVD) to create a co-occurrence frequency matrix that differentiates between good and bad . The second method the second method is based on Course2vec, which is considered as many-to-one relationship. On the other hand, the second combined the predicted grades to improve the rankings produced by the recommendation methods, wich combined the Cumulative Knowledge-based Regression Models (CKRM)(Morsy and Karypis, 2017). Their results showed that grade-aware course recommendation approach outperformed grade-unaware recommendation approaches by recommending courses that increase the students' GPA.
(Buenaño-Fernández et al., 2019) utilized several supervised learning algorithms to predict students' academic status as fail or pass. Their dataset was collected between the first semester in the year of 2016 and the second semester in the year of 2018 from a university in Ecuador. The results showed that the final grade prediction does not improve the accuracy of the the recommender system.
(Iqbal et al., 2017) compared the performance of different machine learning algorithms, which are Collaborative Filtering (UBCF algorithm), Matrix Factorization (SVD and NMF), and Restricted Boltzmann Machines (RBM) for students' grade prediction. Used data covered grades of 24 different courses of 225 students for three years (2013, 2014, 2015). Their study found that RBM outperformed both CF and MF.
(Polyzou and Karypis, 2016) used two approaches to predict student's grades: course-specific matrix factorization (CSMF) and linear regression. They generated a matrix factorization (MF) model for each course.The dataset was used in this study from the University of Minnesota, which includes grades for Computer Science and Engineering (CS & E), and Electrical and Computer Engineering (ECE) students. The dataset collected for the period between fall 2002 to spring 2014 containing the grades for 76748 students related to 2556 different courses and 2949 students. Their results showed that both proposed approaches outperformed existing traditional methods, and course-recommendations based on regression achieved the best results compared to (CSMF) and linear regression.
(Ren et al., 2018) proposed student grade prediction model based on the additive latent effect (ALE) within the framework of matrix factorization (MF) that focused on outsourced factors rather than data associated with courses and students. The dataset was obtained from George Mason University and covered the period of Fall 2009 to Spring 2016. The proposed model followed the method that was developed by (Morsy and Karypis, 2019).It created matrix factorization (MF) for each student jointly with the course's grades. Moreover, their model utilized additional data such as course instructor and student academic level data. (Ren et al., 2018) applied the Percentage of Tick Accuracy (PTA) as a performance measure. Their study results found that ALE method outperformed existing grade prediction methods in terms of PTA0, PTA1,where PTA0 that is the predicted letter grade and the actual letter grade are the same and PTA1 is the difference between predicted letter grade and the actual letter grade is less than or equal to one consecutive letter grades.
Our study will compare several grade recommendation algorithms. Some have been included in the literature work, such as non-negative matrix factorization (NMF) and singular value decomposition (SVD). Other algorithms, including K-Nearest Neighbors (KNN), co-clustering, Baseline Only, Normal Predictor, and SlopeOne will be applied in our study. We used different machine learning techniques, such as ensemble methods and data standardization, to improve the accuracy of the grade prediction. Table 2 summarizes related work in terms of methods, number of students, number of courses, number of majors, and number of batches that have been used in the corresponding study.

©Ediciones Universidad de Salamanca / cc by-nc-nd

Ref.	Methods	No. Students	No. Courses	No. Majors	No. Batches
(Morsy and Karypis, 2019)	Singular Value Decomposition (SVD),Course2vec, Cumulative Knowledge-based Regression Models(CKRM)	33,896	Na	23	(Fall 2002 to Summer 2017)
(Buenaño-Fernández et al., 2019)	Decision Tree (DT) algorithm	335	68	10	2016-1 semester to 2-2018 semester
(Iqbal et al., 2017)	Collaborative Filtering (CF), Matrix Factorization (MF), and Restricted Boltzmann Machines (RBM) techniques	225	24	1	(2013, 2014, 2015)
(Polyzou and Karypis, 2016)	Course-specific regression (CSR), Student-Specific Regression (SSR), Methods based on Matrix Factorization	2,949	2,556	2	Fall of 2002 to Spring of 2014
(Ren et al., 2018)	Additive Latent Effect (ALE) models within the framework of MF	43.099	4,654	151	Fall 2009 to Spring 2016

Table 3: Summary of related work

4. Methodology

Figure 3 presents a conceptual diagram showing the proposed system for student grade prediction. The methodology contains three main components: data pre-processing, data modelling, and model evaluation. The following sub-sections discuss each main stage in detail.

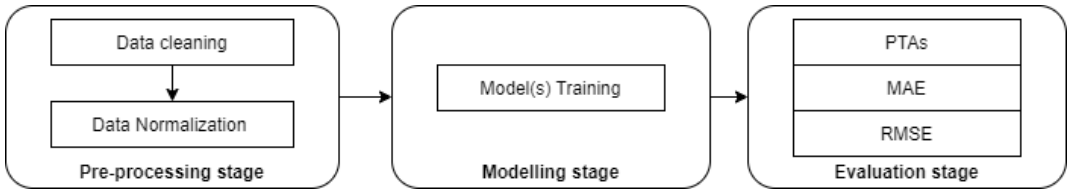


Figure 3: High-level visualization of the proposed methodology

4.1 Pre-processing Stage

In the first stage, we prepare the data set to pass it to the prediction algorithm, which is an important step in machine learning. We applied several preprocessing tasks:

©Ediciones Universidad de Salamanca / cc by-nc-nd

- Data cleaning step
  Collected data contains students records with grades equal to 0 or -1. Those grades represent entry error or courses withdrawal. In this step those records are removed and considered as noisy data because it does not represent the actual student academic situation. In addition, the collected data contains courses enrolled by a small number of students over a period of 5 years. Those courses may represent old courses that have been removed from the student's study plan in the specialization. This step will also remove all courses taken by less than 50 students because these courses could not be taken by new students. Therefore, there is no need to add them to the matrix.

- Applied Z-score Normalization
  We applied Z-score normalization to rescale the value of our data to a common scale without modifying the difference in the range of value. Such techniques are useful in classification tasks.
  Normalization with Z score technique is calculated based on the mean and standard deviation of the student grades column (Cheadle et al., 2003). For student grade G_i in grade column, normalized value N_i is given as follows:

$$N_i = \frac{G_i - Avg(Grade)}{std(Grade)} \tag{1}$$

4.2 Modelling Stage

In this stage, we utilize several prediction algorithms that can be grouped as either supervised machine learning methods or matrix factorization techniques. Specifically, this study will conduct experiments that include the following prediction algorithms:

A. Singular Value Decomposition (SVD) SVD is a very popular matrix factorization technique that decomposes student's courses matrix R into as follows(Mnih and Salakhutdinov, 2008; Iqbal et al., 2017):

$$R = U \sum V^T (2)$$

U is an $m \times k$ orthogonal matrix, where m represents number of students and k represents the rank of the matrix R in this case. $\sum$ is an $k \times k$ diagonal matrix with singular values along the main diagonal entries and zero everywhere else, V^T is the transpose matrix of V, where V is an $k \times n$ orthogonal matrix where n represents the number of courses.

Singular Value Decomposition with implicit grades (SVD_{PP}).
The SVD_{PP} algorithm is an extension of SVD considering implicit grading. The gradings of a given student, called an evaluation, is represented as an incomplete array i, where ij is the predicted grading of this student i. The prediction $\hat{r}_{ij}$ is set (Ricci et al., 2011; Koren, 2008):

$$\hat{r}_{ij} = \ + \ b_i + \ b_j + \ {q_j}^T \left(p_i + \ |J_i \ |^{-\frac{1}{2}} \sum_{j \in J_i} y_j \right), \tag{3}$$

©Ediciones Universidad de Salamanca / cc by-nc-nd

where the y_j terms are a new set of item factors that capture implicit grading. Here, an implicit grading describes the fact that a student i enrolled a course j, regardless of the grading value. If student i is unknown (new student), then the bias b_i and the factors p_i are assumed to be zero. The same applies for item j with b_j, $q_j$and y_j (Hug, 2020).

Slope One Slop One algorithm is simple collaborative filtering algorithm. This is a straightforward implementation of the SlopeOne algorithm developed by (Lemire and Maclachlan, 2005). The prediction $\hat{r}_{ij}$ is set as:

$$\hat{r}_{ij} = I_i + \frac{1}{|R_u\ (i)|} \sum_{j \in R_u(i)} dev\,(u,j)\,, \tag{4}$$

where $R_u\,(i)$ is the set of relevant courses, i.e. the set of courses j graded by student i that also have at least one common student with u. $dev\,(u,j)$ is defined as the average difference between the grading of u and those of j (Hug, 2020):

$$dev\,(i,j) = \frac{1}{|I_{uj}\ |} \sum_{i \in I_{uj}} r_{iu} - r_{ij}$$

BaselineOnly Algorithm predicting the baseline grade estimate for given student and course(Koren, 2010).

$$\hat{r}_{ij} = b_{ij} = i + b_i + b_j \tag{5}$$

If student i is unknown, then the bias b_i is assumed to be zero. The same applies for item j with b_j.

Non-negative Matrix Factorization (NMF) NMF algorithm is a matrix factorization technique. It is similar to the SVD algorithm with slight modification in predicting $\hat{r}_{ij}$ where p_i , q_j are the student and course bias terms respectively . The prediction $\hat{r}_{ij}$ is set as follows(Luo et al., 2014; Zhang et al., 2006):

$$\hat{r}_{ij} = q_j^T + p_i, \tag{6}$$

where student and course factors are kept positive.

NormalPredictor Algorithm predicts a random grading based on the distribution of the training set, which is assumed to be normal. The prediction $\hat{r}_{ij}$ is generated from a normal distribution $\mathcal{N}(\widehat{\mu}, \widehat{\sigma}^2)$, where $\widehat{\mu}$ and $\widehat{\sigma}$ are estimated from the training data using Maximum Likelihood Estimation:

$$\widehat{\mu} = \frac{1}{|\mathcal{R}_{train}|} \sum_{r_{ij} \varepsilon \mathcal{R}_{train}} r_{ij} \tag{7}$$

$$\tag{8}$$

$$\widehat{\sigma} = \sqrt{\sum_{r_{ij} \varepsilon \mathcal{R}_{train}} \frac{(r_{ij} - \widehat{\mu}\ )^2}{|\mathcal{R}_{train}|}} \tag{9}$$

©Ediciones Universidad de Salamanca / cc by-nc-nd

K-Nearest Neighbor (KNN) KNN is a basic collaborative filtering algorithm that follows a basic nearest neighbors' approach. The prediction $\hat{r}_{ij}$ is set as:

$$\hat{r}_{ij} = \frac{\sum_{v \varepsilon N_j^k(i)} sim(i,v) \cdot r_{vj}}{\sum_{v \varepsilon N_i^k(i)} sim(i,v)} \tag{10}$$

where sim(i,v) represents the similarity between student i and student v

K-Nearest Neighbor with ZScore (KNNWithZScore) KNNWithZScore algorithm is similar to KNN algorithm with slight modifying in prediction. KNNWithZScore assigns the nearest neighbors by calculating the z-score normalization of each student(Koren, 2010). The prediction $\hat{r}_{ij}$ is set as follows:

$$\hat{r}_{ij} = \frac{\sum_{v \varepsilon N_j^k(i)} sim(i,v) \cdot (r_{v,j} - i_v)/\sigma_v}{\sum_{v \varepsilon N_j^k(i)} sim(i,v)} \tag{11}$$

CoClustering CoClustering is collaborative filtering algorithm where students and courses are assigned some clusters C_i, C_j and some co-clusters C_{ij}. The prediction $\hat{r}_{ij}$ is set as follows(George and Merugu, 2005):

$$\hat{r}_{ij} = \overline{C_{ij}} + (\mu_i - \overline{C_i}) + (\mu_i - \overline{C_j}), \tag{12}$$

where $\overline{C_{ij}}$ is the average grading of co-cluster C_{ij}, $\overline{C_i}$ is the average grading of i's cluster, and $\overline{C_j}$ is the average grading of i's cluster. If the student is unknown, the prediction is $\hat{r}_{ij} = \mu_j$. If the course is unknown, the prediction is $\hat{r}_{ij} = \mu_i$. If both the student and the course are unknown, the prediction is $\hat{r}_{ij} = \mu$.

4.3 Evaluation Stage

Evaluation of prediction systems is typically conducted experimentally, rather than analytically. We have utilized two distinct groups of evaluation metrics, regression metrics and classification metrics. The first group of metrics evaluate the actual predicted grades (numerical output), while the second group of metrics evaluate the output after converting it to letter grades. Both groups of evaluation metrics are usually employed in evaluating student performance prediction models (Polyzou and Karypis, 2016; Ren et al., 2018). Namely, the selected metric falling in the first group are: Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). The selected metric in the second group is the Percentage of Tick Accuracy (PTA's). Next, we will define the selected evaluation metrics.

A. Regression Metrics. In this group of metrics, we utilized two measures, namely: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

Given I samples, the prediction error for each sample is calculated as follows:

$$e_i = r_i - \hat{r}_i \quad \text{for} \quad (i = 1,2,3,\ldots,I) \tag{13}$$

where r_i is the observed course grade for student i in the testing dataset, and $\hat{r}_i$ is the predicted grade for the same course for student i in the testing dataset. The MAE and the RMSE are calculated for the test dataset as follows

©Ediciones Universidad de Salamanca / cc by-nc-nd


(Chai and Draxler, 2014):

$$\text{MAE} = \frac{1}{I}\sum_{i=1}^{I} \mid e_i \mid \tag{14}$$

$$\text{RMSE} = \sqrt{\frac{1}{I}\sum_{i=1}^{I} e_i^2} \tag{15}$$

In both metrics, smaller values (closer to 0) indicate lower error in prediction and thus better model performance. A perfect model will result in a value of 0, indicating that there is no error in predicting the student grade. A model with $MAE = 8$ indicates that this model has, on average, an absolute error of 8, which means that the error in predicted grade is, on average, 8 points less or more than the actual score.

B. Classification Metrics. In this group of metrics, we utilize three variants of the Percentage of Tick Accuracy (PTA's) measure. The dataset was collected from a public university that applies letter grade systems. This means that the final grade that will be recorded for a student is a letter ($A+$, A, $B+$, B, $C+$, C, $D+$, D, and F) , which is based on the numerical grade achieved in the specific course. Table 4 shows the letter grades and their associated numerical grades.

letter grade	numerical grade range
A+	100–95
A	94.5–90
B+	89.5–85
B	84.5–80
C+	79.5–75
C	74.5–70
D+	69.5–65
D	64.5–60
F	59.5–0

Table 4: Letter grades and their associated numerical grades

To calculate the Percentage of Tick Accuracy (PTA) we convert the r_i and $\hat{r}_i$ for each student and course pair from a numerical grading format to letter grading format by following the ranges in Table 4 for both the training set and testing set. After that we calculated the tick as the difference between two consecutive letter grades for the predicted letter grade and the observed letter grade. In this study we applied three levels of PTA as follows:

$$\text{PTA}_0 = \frac{TP_0}{I_{ts}} \tag{16}$$

©Ediciones Universidad de Salamanca / cc by-nc-nd

Where TP_0 is the number of records in testing set that predicted letter is equal to the observed letter,and I_{ts} is the number of records in the testing set.

$$\text{PTA}_1 = \frac{TP_1}{I_{ts}} \tag{17}$$

where TP_1 is the number of records in the testing set that achieved the following condition: The difference between the predicted letter and the observed letter is less than or equal to one consecutive letter grade, and I_{ts} is the number of records in the testing set.

$$\text{PTA}_2 = \frac{TP_2}{I_{ts}} \tag{18}$$

where TP_2 is the number of records in the testing set that achieved the following condition: The difference between the predicted letter and the observed letter is less than or equal to two consecutive letter grades, and I_{ts} is the number of records in the testing set.

In all variants of PTA, higher values (close to 1) indicate a better prediction model. A perfect prediction model is a model with $\text{PTA}_0 = 1$, which indicates that all student (letter) grades were correctly classified in the testing set.

5. Results and Analysis

Experimental Settings. For a valid evaluation of our proposed approach, we split the data set into two separate folds; training and testing. The training fold is used to train the prediction models. The trained models are then evaluated on an unseen testing fold. We split our data based on timeline. In other words the training set contains all students' grades reported before the first semester in 2019. The testing set contains student records for the 2019 academic year. The number of records included in the training dataset is 5006 records (88%), and the number of records included in the testing dataset is 677 records (12%). The training data records are related to 213 students; and the test dataset has 136 students. The training set includes 51 courses and the testing set includes 27 different courses. Table 1 provides summary statistics of the training and testing sets, in comparison with the complete dataset.

Parameter Tuning. To optimize the performance of the prediction model in terms of accuracy, the grid search method has been applied to find the best hyper-parameters set for each applied algorithm. In cases of SVD and SVDpp the grid search applied to the following parameters and values n_epochs (3,5) lr_all (0.005,0.006) reg_all (0.2,0.4). The best parameters are reported with SVD and SVDpp are { 'n_epochs': 5, 'lr_all': 0.005, 'reg_all': 0.4 }. On the other hand, in cases of KNN and KNNWith-ZScore, we applied the grid search to find the best number of k in the range of (3, 7, 40 *default*), the k $\bar{3}$ achieved the best result.

The code and cleaned data to regenerate the results are available here. https://www.kaggle.com/budoralharbi/grade-prediction

Experimental Results. Table 4 shows the evaluation results on five metrics, namely: RMSE, MAE, PTA_0, PTA_1, and PTA_2, as described in section 4.3. The metrics evaluated the selected prediction algorithms described in section 4.2. Namely, the selected algorithms are: SVD, SVDpp, SlopeOne, BaselineOnly, NMF, Normal Predictors, KNNBasic, KNNwithZscore, and CoClusting. Considering the first metric, RMSE, SVDpp

©Ediciones Universidad de Salamanca / cc by-nc-nd


achieved the best and lowest score compared to the compared algorithms. Considering all other remaining metrics, KNNwithZscore consistently outperformed all other algorithms. In term of MAE, the KNNWithZScore outperformed the compared algorithms with the lowest MAE score. Additionally, the KNNWithZScore algorithm achieved the highest score in terms of PTA with 29%, 62%, and 84% respectively. For some evaluation metrics, such as PTA_2, the SVD algorithm achieved similar performance to that of KNNWithZScore algorithms. Finally, MF and NormalPredictor algorithms achieved the worst performance in all performance measurements applied in this study.The reported results in this study can not be compared with previous studies due to the difference in the applied datasets; however, it can be observed that the obtains results in this study are reasonable when compared with (Polyzou and Karypis, 2016) and (Ren et al., 2018) when considering the number of records and the number of features.

Method	**RMSE**	**MAE**	**PTA_0**	**PTA_1**	**PTA_2**
SVD	12.24	8.71	0.24	0.58	0.84
SVD_{PP}	**12.11**	8.77	0.26	0.61	0.81
SlopeOne	12.71	8.96	0.25	0.58	0.82
BaselineOnly	12.39	8.86	0.21	0.58	0.83
NMF	26.55	19.18	0.09	0.26	0.45
Normal Predictor	23.74	18.33	0.12	0.31	0.45
KNNBasic	12.98	9.38	0.22	0.56	0.77
KNNWith ZScore	12.18	**8.68**	**0.29**	**0.62**	**0.84**
CoClustering	12.88	9.09	0.22	0.58	0.80

Table 5: Experimental results

6. Conclusion

The main contribution of this paper is to investigate a variety of grade prediction techniques on a dataset provided by a public university in Saudi Arabia. The prediction techniques applied in this paper have been widely used by many researchers in different domains including student perfromance prediction. The results of this paper show that KNNWithZScore algorithm achieved the highest prediction performance, where 84% of the student grades were correctly in terms of PTA_2. Future work will consider collecting data that cover more students, courses, and majors to improve the prediction model's performance.

7. References

Acharya, A. and Sinha, D., 2014. Early prediction of students performance using machine learning techniques. *International Journal of Computer Applications*, 107(1).

Buenaño-Fernández, D., Gil, D., and Luján-Mora, S., 2019. Application of machine learning in predicting performance for computer engineering students: A case study. *Sustainability*, 11(10):2833.

©Ediciones Universidad de Salamanca / cc by-nc-nd

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

Cheadle, C., Vawter, M. P., Freed, W. J., and Becker, K. G., 2003. Analysis of microarray data using Z score transformation. *The Journal of molecular diagnostics*, 5(2):73–81.

George, T. and Merugu, S., 2005. A scalable collaborative filtering framework based on co-clustering. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 4–pp. IEEE.

Hug, N., 2020. Surprise: A Python library for recommender systems. *Journal of Open Source Software*, 5(52):2174.

Iqbal, Z., Qadir, J., Mian, A. N., and Kamiran, F., 2017. Machine learning based student grade prediction: A case study. *arXiv preprint arXiv:1708.08744*.

Koren, Y., 2008. 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.

Koren, Y., 2010. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 4(1):1–24.

Lemire, D. and Maclachlan, A., 2005. Slope one predictors for online rating-based collaborative filtering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 471–475. SIAM.

Luo, X., Zhou, M., Xia, Y., and Zhu, Q., 2014. An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics*, 10(2):1273–1284.

Mnih, A. and Salakhutdinov, R. R., 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264.

Morsy, S. and Karypis, G., 2017. Cumulative knowledge-based regression models for next-term grade prediction. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 552–560. SIAM.

Morsy, S. and Karypis, G., 2019. Will this Course Increase or Decrease Your GPA? Towards Grade-aware Course Recommendation. *arXiv preprint arXiv:1904.11798*.

Polyzou, A. and Karypis, G., 2016. Grade prediction with course and student specific models. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 89–101. Springer.

Ren, Z., Ning, X., and Rangwala, H., 2018. Ale: Additive latent effect models for grade prediction. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 477–485. SIAM.

Ricci, F., Rokach, L., and Shapira, B., 2011. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer.

Zafra, A. and Ventura, S., 2009. Predicting Student Grades in Learning Management Systems with Multiple Instance Genetic Programming. *International working group on educational data mining*.

Zhang, S., Wang, W., Ford, J., and Makedon, F., 2006. Learning from incomplete ratings using non-negative matrix factorization. In *Proceedings of the 2006 SIAM international conference on data mining*, pages 549–553. SIAM.

©Ediciones Universidad de Salamanca / cc by-nc-nd