Case Study On Predicting Student Performance by Applied Machine Learning Algorithms

Bui Thi Bich Ngoc   
ICT Department, FPT UniversityHanoi, Vietnam  
Ngocbtbhe140996@fpt.edu.vn

Ha Long Duy   
ICT Department, FPT UniversityHanoi, Vietnam  
Duyhlhe141012@fpt.edu.vn Ngo Tung Son   
ICT Department, FPT UniversityHanoi, Vietnam  
Sonnt69@fe.edu.vn

Le Bat Bao   
ICT Department, FPT UniversityHanoi, Vietnam  
Baolbhe140311@fpt.edu.vn Le Duc Anh   
ICT Department, FPT UniversityHanoi, Vietnam  
Anhldhe140935@fpt.edu.vn

Nguyen Tien Dat  
ICT Department, FPT UniversityHanoi, Vietnam  
Datnthe140556@fpt.edu.vn

*Abstract*

During the history, the art of predicting student grade is mostly based on the teacher’s instinct. This method although is trustworthy yet outdated and tedious to be applied to real work. In response, we had devised a way to make the machine do this for us. But in it lies more problems and challenges: (1) Students individually have a personal interest that tremendously differs from each other and vary over time; (2) Courses and teachers could change by semester. In this paper, we present a novel machine learning method for predicting student performance in the FPT University – Hoa Lac campus. The proposed method utilizes multiple features. One of them is the recommender system, which is applied to accommodate for their relative interest with the classes that they take. The second one is Bayesian linear regression to account for the second issue.

Keywords

Student Performance Prediction, Recommender System, Bayesian Linear Regression, Rating matrix

# **Introduction**

# First, students’ performance can differ tremendously in terms of sets of goals as well as their chosen areas (major, specializations), resulting in different selected courses and the order said courses. On the other hand, a student could wish to change their subject and major. This lead to the fact that one course could be taken by a student with different sets of records. Since predicting student academic performance relies on their past grades, a key challenge for training an effective predictor is the way we handle heterogeneous student data due to the aforementioned problem. In contrast, solving problems in ITSs often follow routine steps which are the same for all student [1]. Identically, predictions of student’s performance in courses are often based on in-course assessments which are designed to be the same for all students [2].

# Second, courses and teacher quality in our school can vary depending on the semester that you enrolled in. A deeper look into our data we can clearly see a low distribution in grades, telling that in the first few years our school still struggles to teach and passes on information to the students. On the other hand, after a few years, those scores started to gain traction and slowly going up.

# This work proposes a common approach which applies numerous of method for predicting student performance. Because of that we use data provided to us by our school, we do not focus on competing but rather getting the real educational data from this event and employing these methods in e-learning. Our main contributions are:

# Applying recommender system techniques such as rating matrix[3] in the educational context, especially for predicting student performance

# Using Bayesian linear regression[4]in the same vein as applying the recommender system to address the latter issue

# The rest of the paper is organized as follows. Section 2 introduces some related works; section 3 formulates the student performance prediction using recommender system approach; section 4 briefly introduces our applied methods and their techniques which are used in this study; section 5 describes data sets and the proposed methods including a method for mapping educational data to recommender systems and to traditional regression problem; section 6 shows the experimental results; and finally, section 7 for conclusion.

# **Related Researches**

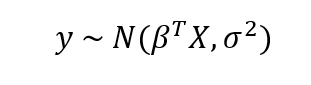
## Rating Matrix

An effective collaborative filtering approach is rating matrix (Koren et al. 2009). By factorizing the user-item interactions matrix, rating matrix can map both users and items to a joint latent factor space. Therefore, user-item interactions are modeled as inner products in that space. Formally, rating matrix decomposes the original rating matrix R into two low-rank matrices U and V consisting of the user and item latent factor vectors respectively, such that R ≈ UV. Given the latent factor vectors for users and items, a user’s rating for a movie is predicted by the inner product of those vectors. The objective function of rating matrix can be written as:

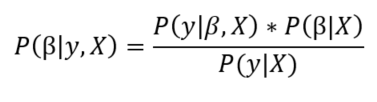
where L(·, ·) is the loss function that measures the distance between two matrices with the same size, the last two terms are the regularizations used to avoid overfitting and || · ||F denotes the Frobenius norm. By specifying different L(·, ·), many rating matrix models have been proposed, for example, non-negative rating matrix, probabilistic rating matrix, max-margin matrix factor, etc. When side information is available, some rating matrix models generate a rating from the product of latent factor vectors which contain additional information about users or items. Various models show that additional side information can act as a useful informative prior that can significantly improve results (Porteous, Asuncion, and Welling 2010; Singh and Gordon 2008).

## Bayesian Linear Regression

## Bayesian Linear Regression[4] assumes data is sampled from a probability distribution such as the [normal (Gaussian) distribution](http://www.statisticshowto.com/probability-and-statistics/normal-distributions/):



The mean of the Gaussian is the product of the parameters, β and the inputs, X, and the standard deviation is σ. In Bayesian Models, not only is the response assumed to be sampled from a distribution but so are the parameters. The objective is to determine the [posterior probability distribution](http://www.statisticshowto.com/posterior-distribution-probability/) for the model parameters given the inputs, X, and outputs, y:



The end result of Bayesian Linear Modeling is not a single estimate for the model parameters, but a distribution that we can use to make inferences about new observations. This distribution allows us to demonstrate our uncertainty in the model and is one of the [benefits of Bayesian Modeling methods](http://twiecki.github.io/blog/2013/08/12/bayesian-glms-1/). As the number of data points increases, the uncertainty should decrease, showing a higher level of certainty in our estimates.

## Support Vector Machine[5]

## Extra Trees & Random Forest Classifiers[6]

## Gradient Boosted[7]

# **Exploring The Data**

## Data Cleaning and Preprocessing

We implemented our models in Python using Anaconda and Google Collaboratory. The data which contains 229K marks, for 462 subjects by 10k students was collected from Vietnam FPT University therefore, in order to apply it in Python, we had to reformat the data into their English equivalents. We then split the data into training and test set in 80-20 ratio. We also remove all null variable as well as encode all the students’ marks and subjects to fit the algorithm’s inputs.

## Loss Function

In analogy to MF approaches, we define the loss function as:

where if the mark m which is rated by student n, denotes the Frobenius norm, s is all the mark we have. These are not the only loss functions possible. For instance, by using a quantile loss we can prioritize estimates based on the confidence with which we would like to recommend events. Other possible loss functions include the Huber loss and the hinge loss function, which can be useful in the case of implicit taste information.

## Model Evaluation

We evaluated the prediction results and the established prediction model by Root Mean Square Error (RMSE). The smaller these metrics are, the closer the predicted value is to the actual value. The formulas for the error metrics are shown below:

where stands for predicted values, is the set of actual values and n is the number of observations.

## Feature selection for Bayesian Linear Regression[4]

The input data qualified for Bayesian Linear Regression implementation consists of 3478 students whose had completed at least 2 semesters. We can use a simple measure called the [Correlation Coefficient](https://onlinecourses.science.psu.edu/stat501/node/256) to determine the most useful variables for predicting a grade. This is a value between -1 and +1 that measures the direction and strength of a linear relationship between two variables. We can find these correlations of numerical variables by using pandas library:

Fail -0.400638

overAllAvgMath 0.267007

overAllAvgIt 0.314816

CurrentTerm 0.359561

overAllAvgOther 0.443207

overAllAvg 0.526690

percentile 0.935045

For categorical variables, we one-hot encode and then calculate the correlation coefficient:

Gender\_Male -0.174527

Major\_SE -0.171446

Major\_Software Engineering -0.018890

Major\_MC -0.015836

Major\_JS -0.015159

.

.

Major\_FB 0.036612

Major\_Truyền thông đa phương tiện 0.041669

Major\_IB 0.043791

Major\_BA 0.050588

Major\_BBA 0.052517

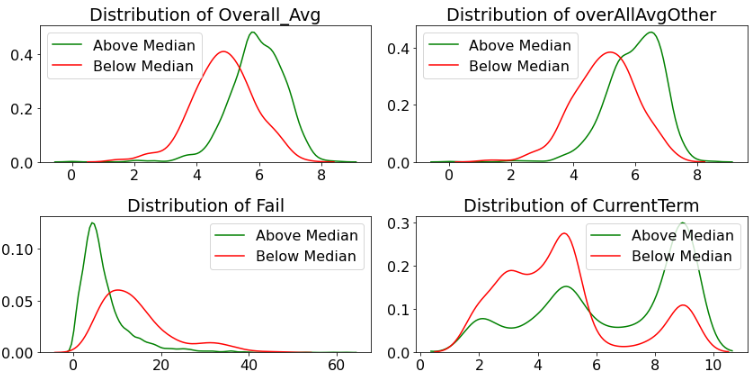
Major\_BIB 0.061226

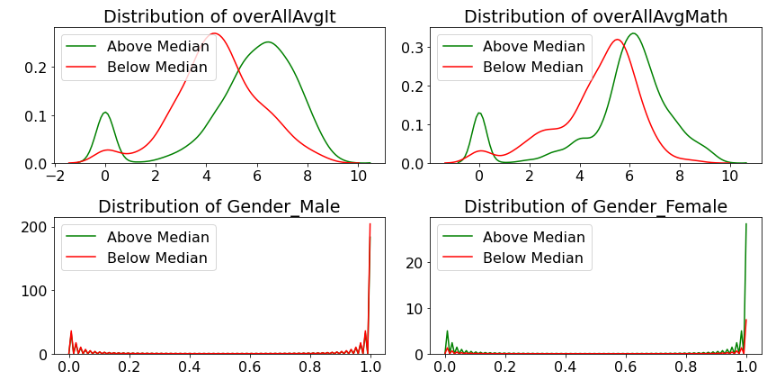
Major\_BSE 0.067911

Gender\_Female 0.174527

In this problem we will use these results to perform feature selection by retaining only the 8 variables that are most highly correlated with the final grade.

*Figure 1. Distribution of features in correspondence to median grade*





While we are performing feature selection, we also split the data into a training and testing set in 80-20 ratio using a  [**Scikit-learn**](http://scikit-learn.org/) function. This is necessary because we need to have a hold-out test set to evaluate our model and make sure it is not overfitting to the testing data.

This leaves us with 2781 training observations and 696 testing data points.

## 3.5. Feature normalization

In this study, the extracted features were initially at different scales; therefore, the data were normalized by dividing the values in each session matrix column by the mean of thatcolumn. Thus, each column value is located around the mean.

# **Result And Discussion**

## Bayesian Linear Regression

* + 1. *Overall Analysis*

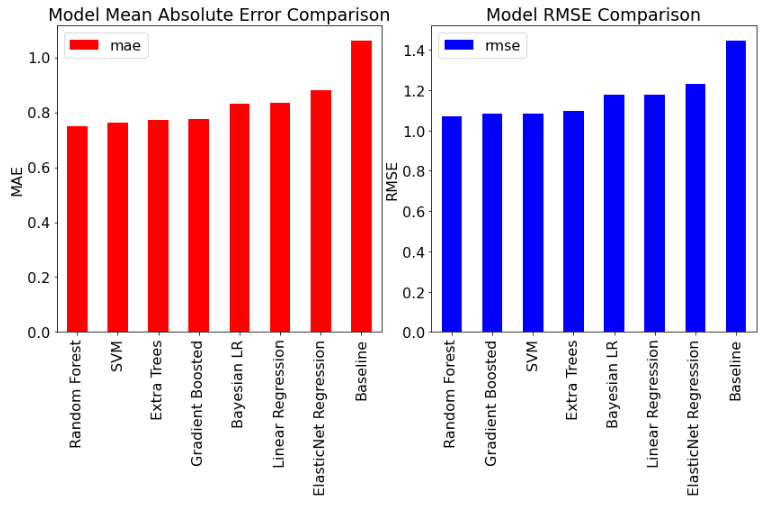
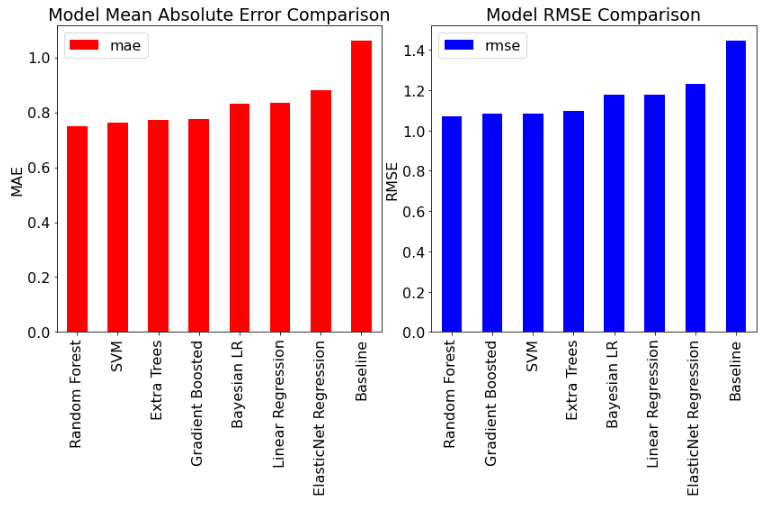
Indicated from Table 3.1.1, Bayesian Linear Regression when being applied to input data returned RMSE of 1.1939 (which is 19.46% more accurate than those of the naive baseline metrics). Meanwhile, the model that achieved the best accuracy is the Support Vector Machine with RMSE of 28.7036% better than the baseline. Among models, ElasticNet Regression was the least accurate model which produced the largest RMSE score: 1.28233 (only 13.567% better than the baseline’s). However, the overall RMSEs of models weren’t satisfactory results to the applicable prediction of values due to overfitting problems and inefficient data preprocessing stage.

1. Comparisons Of Mae And Rmse Evaluation Of Distinctive Models

|  |  |  |
| --- | --- | --- |
|  | **MAE** | **RMSE** |
| Linear Regression | 0.843136 | 1.19481 |
| ElasticNet Regression | 0.923633 | 1.28233 |
| Random Forest | 0.765144 | 1.09378 |
| Extra Trees | 0.754432 | 1.05966 |
| SVM | 0.742788 | 1.05776 |
| Gradient Boosted | 0.783338 | 1.10901 |
| Baseline | 1.09572 | 1.48361 |
| Bayesian LR | 0.843174 | 1.1949 |

It is better recommended to use SVM, Random Forest, Extra Trees and Gradient Boosted Models for input data types and features alike.

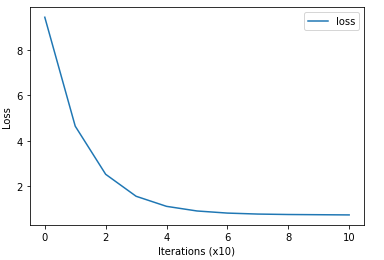
*Figure 2.Visualized MAE and RMSE evaluation of distinctive models*



## Rating matrix

* + 1. *Overall Analysis*

In Fig. 3, We show the empirical convergence of our algorithm on the data we have. One can see that our algorithm decreases the cost function monotonically.



*Figure 3: Empirical Convergence plot*

## Discussion

Within the Bayesian Linear Regression implementation, we foremostly build a formula relating the features to the target and divide on a prior distribution for the data likelihood.

Throughout the feature selection process, features with the highest correlation to last semester’s average are specified in Table II:

* Overall average of other subject groups (besides IT-related and Math-related subject groups) has the highest negative weight
* Overall average has the highest positive weight
* Female student has a positive weight

1. Mean weights of selected features to the prediction of last semester average

|  |  |  |
| --- | --- | --- |
|  | **Mean** | **SD** |
| Intercept | 3.491 | 0.256 |
| Overall\_Avg | 2.254 | 0.151 |
| overAllAvgOther | -1.489 | 0.119 |
| Fail | -0.036 | 0.006 |
| CurrentTerm | 0.202 | 0.013 |
| overAllAvgIt | -0.316 | 0.027 |
| overAllAvgMath | -0.070 | 0.018 |
| Gender\_Female | 0.331 | 0.070 |
| sd | 1.171 | 0.016 |

The intercept, 3.491, represents our guess if every variable of a student is 0 (not correlated to the last semester average). There is also a large standard deviation (SD) for the data likelihood, indicating large uncertainty in the targets. Overall, we see considerable uncertainty in the model because we are dealing with a small number of samples. With only about a few thousands of students, we do not have enough data to pin down the model parameters precisely.

As mentioned, Bayesian Linear Regression returns a normal distribution of estimated outputs by multiplying the model parameters by our data point to find the mean and using the standard deviation from our model. This can be visualized when we test with a few examples from the testing set (with training-testing distribution of 80-20):

# **Conclusion**

This paper proposes an approach to predicting FPT student performance by past performance. Using the given dataset along with Rating matrix and Bayesian Linear Regression approach to account for the non-heterogenous cases, an architecture for predicting student performance was generated and built to incorporate student and facility evolving nature. These data-driven methods could be of use when put in combination with other pedagogical method to evaluate student academic performance. Future works will include extending to recommend courses to students.

# **References**

1. KDD Cup, “Educational data minding challenge,” https://pslcdatashop: web:cmu:edu/KDDCup/, 2010.
2. Y. Meier, J. Xu, O. Atan, and M. van der Schaar, “Personalized grade prediction: A data mining approach,” in Data Mining (ICDM), 2015 IEEE International Conference on. IEEE, 2015, pp. 907–912
3. Will Koehrsen, “Bayesian Linear Regression in Python: Using Machine Learning to Predict Student Grades Part 2”, Towards Data Science, Medium, April 21st 2018
4. Will Koehrsen, “Bayesian Linear Regression in Python: Using Machine Learning to Predict Student Grades Part 1”, Towards Data Science, Medium, April 20th 2018
5. Rohith Gandhi, “Support Vector Machine — Introduction to Machine Learning Algorithms”, Towards Data Science, Medium, June 7th 2018
6. Frank Ceballos, “An Intuitive Explanation of Random Forest and Extra Trees Classifiers”, Towards Data Science, Medium, June 14th 2019
7. Prince Grover, “Understanding Gradient Boosting Machines”, ML Review, Medium, Dec 9th 2017