Undergraduate GPA Prediction using Python-based Data mining and Machine Learning methods

Vaishali Sharma, Gary M. Weiss
Dept. of Computer & Info. Science
Fordham University
Bronx, New York, USA
{vsharma20, gaweiss}@fordham.edu

***Abstract* - Education data mining has gained a lot of attention as the universities are more concerned about the academic performance, learning capabilities and progress of their students. They need effective measures or methods to improve the academic standing of students, reduce the student drop-out or failure rate. This study aims to predict the GPA of undergraduate students at a large university using various data mining and machine learning methods and approaches. This early GPA prediction information can be used by the instructors to mitigate potentially negative outcomes for a student such as failing or dropping out a course, low academic performances. Python based data mining methods such as Linear regression (LR), naïve bayes (NB), support vector regression (SVR) and K-nearest neighbor (KNN) are used to predict the GPA of students. Based on the accuracy models, it was found that naïve bayes model showed best prediction performance as compared to the KNN model. Term -II (which is usually the second year in an undergraduate degree program) has been found to be the best term for making course-specific grade predictions for future terms. The study identifies the best model and best term (i.e., year) for predicting future grades or academic performance of a student. The work is very useful for educational institutions, interested in improving the success rate or graduation rate of their students by mitigating early warning threats such as course failing, low performance and dropping out risks.**

Keywords—GPA prediction, machine learning, data mining, python, academic performance

I. Introduction

Educational institutions are more concerned about the academic standing, learning capabilities and progress of their admitted students. Predicting academic performance of a student at an early stage or at the graduation time could be helpful in detecting students-at-risk, reducing failure rates and in overcoming several educational challenges. Education data mining (EDM) is a fast-growing scientific field offering the potential to analyze a variety of student data and to discover valuable knowledge and information from the student database [1]. Institutions of higher learning use this discovered information or knowledge in several educational purposes such as improving the academic performance of students, estimating student drop-out or failure rate, or recommending the best major to study for the future terms [2]. Data mining methods such as clustering, association rule mining, and classification and regression methods including neural network (NN), support vector regression (SVR), K-means clustering may be used to extract the useful information from the education data. The study aims to implement several classification and regression data mining methods to assist educational institutions with predicting their student' grade point average (GPA) and their performances. If the predicted GPA is low, then extra efforts are needed to improve the academic standings of a student. Among the various prediction goals, early stage and graduation time GPA prediction is of tremendous importance because it could help in predicting the outcome of a course or degree chosen by a student. This early GPA prediction information can enable instructors to identify students at-risk and take necessary steps to mitigate negative outcomes for students. This prediction could significantly reduce the student's failure rate and drop-out risk. The scope of the present research is to predict the grade point average (GPA) of student at graduation and using that information to predict the GPA of future terms. Firstly, the grade point average (GPA) of undergraduate students using all course term GPA will be predicted and then a relationship between first two terms and final GPA using the python-based data mining and machine learning algorithms will be developed to predict the GPA of future terms. Python-based data mining models such as Linear Regression, Support Vector regressions, K-Nearest Neighbor and Naïve Bayes are used in this study to predict academic performances of the students. These performance evaluation of each model has also been done to identify or suggest the best model for undergraduate GPA prediction.

II. Research Questions

Several research questions noted below, need to addressed in order to predict the undergraduate GPA or academic performance of a student: (1) What is the strategy to impute the missing data from the GPA data set? (2) What approaches could be used to predict the final GPA of students? (3) How do you identify what terms to use for future terms GPA prediction? (4) What are the key evaluation metrics that can be used to identify the performance of the model?

III. Prior Work / Background

Numerous data mining and machine learning approaches have been used by academic researchers to predict the academic performance of a student. Table A1 summarizes the review of the prior work done in this field. Different state-of-the-art

machine learning techniques for university grade prediction have been used. Bayesian deep learning models, linear regression, decision trees, naïve bayes, random forest, bayes network, neural network, support vector regression, and collaborative filtering are the most commonly used models for predicting student's grades and their academic performances [4-9]. Restricted Boltzmann Machines (RBM) is also one method that can predict students' grades more accurately and precisely as the technique visualizes uncertainty on student learning and can be used to enable instructors to identify students at risk [3]. Course-specific Bayesian deep leaning models namely multilayer perception (MLP) and long short term memory network were proposed by [4], which can identify the influential courses result in a student's failure of a course. Researchers also evaluated the model's capability of catching at-risk-students. The key features used for the study were number of students, number of courses and grades.

Other study used linear regression and matrix factorization on the large dataset of George Mason University to predict the students' grades [5]. Some case studies involve identifying the best method or model for predicting student's performance [6]. Researchers used methods like Decision trees, Naïve bayes, Random forest, Bayes network to analyze the dataset of 412 post-graduate students. It was found that Random forest algorithm shows the best result as compared to other algorithms. The correctly classified instance with Random Forest algorithm is 61.40% with a very good recall value equal to 1. Some other authors used Support vector regression and collaborative filtering to construct one potential value predictor i.e., an estimated grade that a student will receive for a given course [7].

In addition to grade prediction, some recommendation system could also be developed to help students identify the best major / course based on their interests and past performances [10]. A student must graduate with good grades and credentials and this can only be achieved by choosing the right major to study. Selecting a best major to study is a challenging decision for a student. A robust recommendation system [as shown in 10] could be helpful in suggesting best major to the students and their academic performances can be improved. So, major prediction could also be very useful in improving the academic progress of a student [10-11].

IV. EXPERIMENT METHODOLOGY

A. Dataset Description

The dataset contains records of undergraduate students who were admitted to a large university in a 4-year degree program. Only core-courses data has been used for this study. In the first year, a student takes intensive language, reasoning and philosophy courses. Sophomore year are more inclined towards science and history based courses. Junior year and senior year are dedicated to philosophy and interdisciplinary project based courses. Key features such as student ID, program name, core programs, code, level of study and GPA of a student were used for this study. The grading scheme and sample student dataset is shown in Table I and Table II.

TABLE I. GRADING SCHEME

Grade	A	A-	B+	B	B-	C+	C-	D	F
Points	4.0	3.67	3.33	3.00	2.67	2.33	2.00	1.0	0

TABLE II. SAMPLE STUDENT DATASET

ID	**Programs**	**Code**	**Core Programs**	**Level**	**Score**
SD01	English	1102	Composition I	1st Term	0.00
SD01	Mathematics	1206	Calculus I	1st Term	1.33
SD01	Philosophy	1000	Phil of Human Nature	1st Term	1.33
SD01	Theology	1000	Faith & Critical Reason	1st Term	2.13
SD01	English	1102	Composition II	1st Term	2.53

B. Data pre-processing

Data pre-processing is an integral part of any analytics task and it involves cleaning the data to make it more meaningful before performing any task. Handling missing values, redundant fields, outliers are some of the key data preprocessing approaches. In this study, the student dataset contains some missing values, which have been imputed by taking the average of all grade points and combining the core courses together as one single course. The modified sample data set is shown in Table III. After imputing the missing value, all the four terms were used to determine the final GPA or grad_GPA (Table IV) of a student.

TABLE III. IMPUTING MISSING VALUES (SAMPLE DATASET)

ID	**1st Term**	**2nd Term**	**3rd Term**	**4th Term**
SD01	1.687	2.386	1.571	2.532
SD02	1.814	2.469	1.825	2.737
SD03	1.344	3.217	3.518	0.566
SD04	1.669	2.916	2.472	3.388
SD05	1.595	1.741	3.071	1.820

TABLE IV. FINAL GPA PREDICTION (SAMPLE DATASET)

ID	**1st Term**	**2nd Term**	**3rd Term**	**4th Term**	**grad_GPA**
SD01	1.687	2.386	1.571	2.532	2.044
SD02	1.814	2.469	1.825	2.737	2.211
SD03	1.344	3.217	3.518	0.566	2.161
SD04	1.669	2.916	2.472	3.388	2.611
SD05	1.595	1.741	3.071	1.820	2.057

C. Categorizing History and Prediction GPA Window Data

The whole dataset is divided into history and prediction window data. Historical data will include GPA of first & second terms and grad_GPA values, which will be used to predict the GPA of third and fourth terms using various python based data mining and machine learning models. This process will help us in identifying the best term and best model for predicting the student's GPA or academic performance.

D. Python-based Data Mining & Machine Learning Methods

Python-based Education data mining models namely Linear Regression (LR), Support vector regression (SVR), K-nearest

neighbor (K-NN) and Naïve Bayes (NB) were developed using the entire student dataset. Mean Square Error (MSE) and Accuracy were used as performance metric or evaluation metric of these models.

Linear Regression is a supervised machine learning algorithm where the predicted outcome is continuous and has a constant slope, which is used to predict a continuous values rather than classifying them into categories. Support vector regression (SVR) uses support vector machines to predict a continuous variable and it tries to fit the best line within a predefined or threshold error value. SVR has been proved to be an effective tool in real-value function estimation.

K-nearest neighbor is a simple algorithm that stores all available features and predict the numerical value based on the similarity measures. Whereas a Naïve bayes classifier is a probabilistic machine learning model that is used for classification task. In this paper, Gaussian Naïve bayes model is used, which means continuous values associated with each feature are assumed to be distributed according to a Gaussian distribution i.e., normal distribution.

The models use first two terms or years and grad_GPA as feature values and predict the last two terms GPA based on the previous or historical performances of a student. Important questions such as 1) What will be the predicted third term and fourth term GPA (Fig. 1) ? 2) What will be the MSE or Accuracy of models used? 3) Which model and term are best to work on prediction window? will be answered using these python-based models.

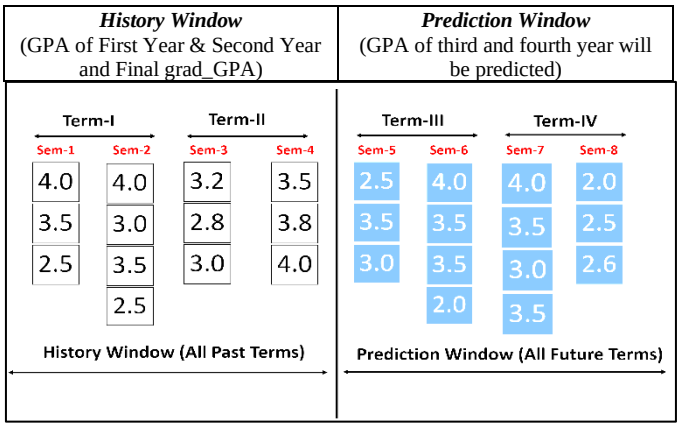

Fig.1 History and Prediction Window (All past terms and All future terms)

E. Evaluation Metrics

Mean Square error (MSE) of an estimator (unobserved quantity) measures the average of the squares of the errors shown in (1). The smaller the mean square error, the closer we are to find the line of best fit.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2 \qquad (1)$$

Another performance metric is *Accuracy*, which is the ratio of correctly predicted samples to the total number of input features or predictions made, and is expressed as (2).

$$\textbf{Accuracy} = \frac{\text{Correctly Predicted Class}}{\text{Total Testing Class}} \times 100\% \qquad (2)$$

V. RESULTS

All the experiments were done on Python and the most common python libraries used for this project are scikit-learn, numpy, pandas, matplotlib, joblib, cython, contextlib, certifi, pillow, scipy etc. Linear regression model and support vector regression model are used to predict the future GPA of students. Table V and Table VI summarizes the term-wise results by LR and SVR models. Historical window uses GPA of 1st and 2nd terms and grad_GPA to make predictions for prediction window (Fig. 2). It has been observed that the mean square error of LR model is less than MSE of SVR model. The smaller the mean square error, the closer we are to find the line of best fit. It may be possible that LR model shows the line of best fit as compared to SVR model because of the low MSE of LR model.

TABLE V. LINEAR REGRESSION RESULTS (SAMPLE DATASET)

IDs	Historical Window (Past terms)			Prediction Window (Future GPA Predictions)			
				Using 1st Term		*Using 2nd Term*	
	1st Term	*2nd Term*	*grad_ GPA*	*3rd Term*	*4th Term*	*3rd Term*	*4th Term*
SD01	1.68	2.38	2.04	1.93	2.07	1.93	1.93
SD02	1.81	2.46	2.21	1.96	2.10	1.97	1.97
SD03	1.34	3.21	2.16	2.21	1.78	2.22	2.22
SD04	1.66	2.91	2.61	2.06	2.19	2.06	2.06
SD05	1.59	1.74	2.05	2.14	1.96	2.15	2.15
SD06	0.33	1.46	1.47	1.94	2.05	1.94	1.94
SD07	1.52	1.83	1.96	2.06	1.99	2.06	2.06
SD08	1.88	1.67	1.84	1.95	2.00	1.96	1.96

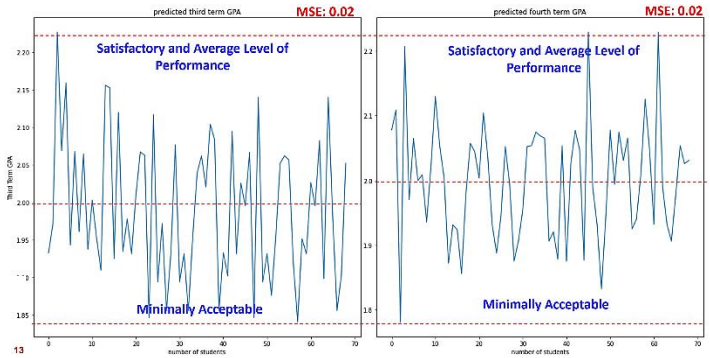

Fig.2 LR Predicted GPA of Third and Fourth term (selected student dataset)

TABLE VI. SV REGRESSION RESULTS (SAMPLE DATASET)

ID	Historical Window (Past terms)			Prediction Window (Future GPA Predictions)			
				Using 1st Term		*Using 2nd Term*	
	1st Term	*2nd Term*	*grad_ GPA*	3rd Term	*4th Term*	*3rd Term*	*4th Term*
SD01	1.68	2.38	2.04	1.92	2.12	1.92	2.12
SD02	1.81	2.46	2.21	1.91	2.19	1.91	2.19
SD03	1.34	3.21	2.16	2.15	1.98	2.15	1.98
SD04	1.66	2.91	2.61	2.09	2.21	2.09	2.21
SD05	1.59	1.74	2.05	2.27	1.91	2.27	1.91
SD06	0.33	1.46	1.47	1.92	2.08	1.92	2.08
SD07	1.52	1.83	1.96	2.09	1.93	2.09	1.93
SD08	1.88	1.67	1.84	1.91	1.95	1.91	1.95

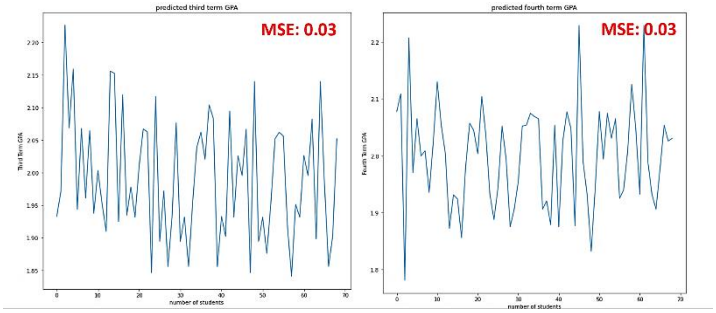

Fig.3 SVR Predicted GPA of Third and Fourth term (selected student dataset)

For the Naïve Bayes and K-Nearest neighbor machine learning models, repeated 3-fold stratified cross-validation is used. The accuracy of each term is calculated by using K-nearest neighbor and naïve bayes algorithms. The number of correct predictions made by the machine learning model is defined by Accuracy metric. Higher the accuracy, higher will be the chances that correct predictions are made or model is trained correctly. Accuracy results of KNN and NB are shown in Table VII. It has been found that accuracy results for Naïve bayes (i.e., 1st term - 78% and 2nd Term - 79%) are higher than the KNN results, so NB could be considered as a best model for predicting grade or academic performance of a student. Also, the accuracy results for Term 2 (year-2, i.e., sophomore year) are better than other terms, so Term 2 could be the appropriate term for making future grade predictions of a student.

TABLE VII. ACCURACY RESULTS OF KNN AND NB

Method	1st term	2nd term	Cumulative terms
KNN	0.42	0.78	0.57
NB	**0.78**	**0.79**	**0.65**

Precision, Recall and f1 score are some of the other metrics that are used to evaluate the model. Precision helps when the cost of false positives is high whereas, recall helps when the cost of false negative is high. f1 score is an overall measure of a model's accuracy that combines precision and recall (shown in (3)).

A good f1 score is an indication of low false positive and low false negative, which means threats are correctly identified and results are not affected by false alarms or noises.

$$f1 = 2 \times \frac{\text{Precision x recall}}{\text{Precision + recall}} \quad (3)$$

The evaluation metric results of Naïve Bayes model are shown in Table VIII. It certainly shows good f1-score for 2nd term (i.e., 82%), meaning low false positive and false negatives are observed. Precision and recall also shows promising values for 2nd term. So, it could be suggested to use 2nd term GPA for future grade predictions of a student. Cumulative GPA could also show promising results.

TABLE VIII. OTHER EVALUATION METRIC RESULTS OF NAÏVE BAYES

Class	Precision		Recall		f1-score	
	1st term	*2nd term*	*1st term*	*2nd term*	*1st term*	*2nd term*
Minimally Accepted	1.00	0.88	0.67	0.78	0.80	0.82
Satisfactory	0.62	0.67	1.00	0.80	0.77	0.73

It is always better to do the predictions by the end of Term 2. Following this, 3rd and 4th term GPA is predicted using the historical GPA values of 2nd term and grad_GPA. Table IX shows the prediction values of representative student dataset only. If the predicted grade is less than 2.00, it is labeled as minimally acceptable and if it is in the range 2.00 < predicted grade < 3.00, it is labelled as satisfactory. Higher grades i.e., in the range of 3.00 to 4.00 are labelled as very good to excellent performance. In the representative dataset (as shown in Table IX), only minimally accepted and satisfactory student performances are observed. This way, we can predict the academic performance of a student in the future terms.

TABLE IX. USE NAÏVE BAYES TO PREDICT ACADEMIC PERFORMANCE OF STUDENTS USING SECOND TERM GPA ONLY

Ids	Historical Window (Past terms)			Prediction Window Student Performance in 3rd Term	Prediction Window Student Performance in 4th term
	1st Term	2nd Term	grad_ GPA		
SD01	1.68	2.38	2.04	Minimally Accepted	Satisfactory
SD02	1.81	2.46	2.21	Minimally Accepted	Satisfactory
SD03	1.34	3.21	2.16	Satisfactory	Minimally Accepted
SD04	1.66	2.91	2.61	Satisfactory	Satisfactory
SD05	1.59	1.74	2.05	Minimally Accepted	Minimally Accepted
SD06	0.33	1.46	1.47	Satisfactory	Satisfactory
SD07	1.52	1.83	1.96	Minimally Accepted	Minimally Accepted
SD08	1.88	1.67	1.84	Satisfactory	Minimally Accepted

VI. CONCLUSIONS

The present study aims to predict the GPA of undergraduate students at a large university using various data mining and machine learning approaches. The early GPA prediction information can be used by any instructor to take necessary steps to mitigate potentially negative outcomes for a student such as failing or dropping out a course. Python based data mining methods such as Linear regression, naïve bayes, support vector regression and K-nearest neighbor are used to address the important research questions of this project. At first, the data was imputed using the mean values of the GPA data set. Then final graduation GPA was predicted based on the course-specific GPAs of past terms. Linear regression model and SVR models were used to predict the third and fourth term student GPA. It was found that the MSE of LR is less than MSE of SVR model, which indicates that the LR could be used for grade prediction of future terms.

Next important question is, what would be the best term to make GPA prediction or academic performance prediction of a student? Naïve Bayes and KNN algorithms were used to make the initial grade prediction and their performances were evaluated based on some performance metrics such as Accuracy, Precision, Recall, f1-score. The correctly classified instance with Naïve Bayes is 79% with a very good recall value equal to 80%. Accuracy of NB is higher than KNN, so this suggests that NB algorithm may be the best model for future prediction. The same is confirmed by f1-score and other metric values. It has also been observed that predictions using Term 2nd could result in higher accuracy results with better recall values and f1-score. So, it would be advantageous to use Term 2nd for making future grade predictions of a student. Lastly, the academic performances of all the students were predicted using NB model and 2nd Term course specific grade data.

The study identifies the best model and best term (i.e., year) for predicting academic performance of a student. The work is very useful for educational institutions, interested in improving the success rate or graduation rate of their students by mitigating early warning threats such as course failing, low performance and dropping out risks.

In addition to grade prediction, some recommendation system could also be developed to help students identify the best major based on their interests and past performances. Selecting a best major to study is a challenging decision for a student A robust recommendation system could be helpful in suggesting best major to the students and their academic performances can be improved. So, major prediction could also be very useful in improving the academic progress of a student The authors have planned to develop such recommendation systems in the future work of this study.

ACKNOWLEDGMENT

I would like to earnestly acknowledge the sincere efforts and valuable time given by my advisor Dr. Gary Weiss. His valuable guidance and feedback have helped in the successful completion of this project. I also thank him for providing the relevant data for this work. I would like to thank Samuel Stein and other lab members for the stimulating group discussions.

REFERENCES

[1] A.E. Tatar, D. Düştegör, "Prediction of Academic Performance at Undergraduate Graduation: Course Grades or Grade Point Average?," Applied Sciences, vol. 10, pp. 4967, Jan 2020.

[2] L.M. Zohair, "Prediction of Student's performance by modelling small dataset size," International Journal of Educational Technology in Higher Education, vol 16, pp. 27, Dec 2019.

[3] Z. Iqbal, A. Qayyum, S. Latif, J. Qadir, "Early student grade prediction: an empirical study," 2nd International Conference on Advancements in Computational Sciences (ICACS) IEEE, pp. 1-7, Feb 2019.

[4] Q. Hu, H. Rangwala, "Reliable deep grade prediction with uncertainty estimation," In Proceedings of the 9th International Conference on Learning Analytics & Knowledge, pp. 76-85, March 2019.

[5] Q. Hu, A. Polyzou, G. Karypis, H. Rangwala, "Enriching course-specific regression models with content features for grade prediction," In 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA) IEEE, pp. 504-513, Oct 2017.

[6] M. Kumar and A.J. Singh, "Evaluation of Data Mining Techniques for Predicting Student's Performance," International Journal of Modern Education & Computer Science, vol 8, 2017.

[7] M. Hunt, S. Lin, C. Kulkarni. "Predicting Course Grades".

[8] M.A. Al-Barrak, M. Al-Razgan, "Predicting students final GPA using decision trees: a case study," International journal of information and education technology, vol 6 pp 528, July 2016.

[9] A. Elbadrawy, A. Polyzou, Z. Ren, M. Sweeney, G. Karypis, H. Rangwala, "Predicting student performance using personalized analytics," Computer. vol 49, pp. 61-69, April 2016.

[10] S.A. Stein, G. M. Weiss, Y. Chen, D.D. Leeds, "A College Major Recommendation System," In Fourteenth ACM Conference on Recommender Systems, pp. 640-644, Sep 2020.

[11] Q.E. Booker, "A student program recommendation system prototype," Issues in Information Systems, pp. 544-551, 2009.

Supplementary Table

Table A1: Selected Papers on Grade Prediction

Year	Paper	Methods Used	Data Source	Features Used	Outcomes
2019 [4]	Reliable deep grade prediction with uncertainty estimation	Bayesian deep learning models - (1) Multilayer perception (MLP) (2) Long short term memory (LSTM) networks	Large five undergraduate majors were chosen.	Number of students, number of courses, number of grades	Course-specific Bayesian deep learning models for grade prediction were proposed.
2017 [5]	Enriching course-specific regression models with content features for grade prediction	Linear Regression	Data set obtained from four departments at George Mason University.	Student related features, Course related features and Instructor related features	For degree programs with high flexibility , predicting the grades with only content feature can give better results.
2017 [6]	Evaluation of Data Mining Techniques for Predicting Student's Performance	Decision Tree, Naïve Bayes, Random Forest, PART and Bayes Network	Student's data of 412 post-graduates is taken	Student Attributes: Age, Gender, Home Location, Communication etc.	Random forest algorithm showed the best result as compared to other algorithms
2016 [7]	Predicting Course Grades	(1) Support Vector Machines (SVMs) and (2) Collaborative Filtering (CF) - Cosine Similarity and Pearson's Correlation	Stanford's Course Rank website (10, 000 anonymized student transcripts and around 7000 courses) data	Student's previous course grades, Recent GPA by department (last three quarters),	Construct one potential value-predictor - an estimated grade that a student will receive for a given course.
2016 [8]	Predicting student's final GPA using decision trees: a case study.	Weka Tool Kit (algorithms implemented in Java), Classification Technique-Decision Trees	Transcript data for female students graduated in 2012 were collected	Student name, Student ID, Final GPA, Semester of graduation, Major, Nationality, Campus, All the courses taken by the student	Used education data mining to predict students' final GPA based on their grades in previous courses. J48 algorithm was used to discover classification rules.
2016 [9]	Predicting student performance using personalized analytics	Multiregression and Matrix Factorization Techniques	Trained and tested performance prediction methods on four types of datasets	Student and course number, course instructor, course level and the department offering that course	Forecast students' grades in future courses as well as in-class assessments.

Model Codes (Selected Codes only)

Linear Regression

First Term

```
from sklearn.model_selection import train_test_split
import numpy as np
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score
#extact the feature values
features = ['first term']
X = final_dataset.loc[:, features].values
#extract the label values
y = final_dataset['grad_GPA']
#define train and test dataset of X and y respectively
X_train, X_test, y_train, y_test = train_test_split (X, y, test_size=0.2, random_s
tate=0)
# Create linear regression object
regr2 = linear_model.LinearRegression()
# Train the model using the training sets
regr2.fit(X_train, y_train)
# Make predictions using the testing set
y_pred = regr2.predict(X_test)
# The coefficients
print('Coefficients: \n', regr2.coef_)
print('Mean squared error: %.2f'
      % mean_squared_error(y_test, y_pred))
print('Coefficient of determination: %.2f'
      % r2_score(y_test, y_pred))
print(y_pred)
```

Second Term

```
from sklearn.model_selection import train_test_split
import numpy as np
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, r2_score
#split feature set from the data frame
features = ['second term']
#set above columns as the feature
X = final_dataset.loc[:, features].values
#set final_score column as the laber
y = final_dataset['grad_GPA']
#add train_test_split to split train and test data
X_train, X_test, y_train, y_test = train_test_split (X, y, test_size=0.2, random_s
tate=0)
# Create linear regression object
svr = SVR(kernel = 'rbf'
# Train the model using the training sets
svr.fit(X_train, y_train)
# Make predictions using the testing set
y_pred = svr.predict(X_test)
print('Mean squared error: %.2f'
      % mean_squared_error(y_test, y_pred))
# The coefficient of determination: 1 is perfect prediction
print('Coefficient of determination: %.2f'
      % r2_score(y_test, y_pred))
```

K-NN

```
# import k nearest neighbor
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
import numpy as np
#split feature set from the data frame
features = ['second term']
#split the train and test data
X = final_dataset.loc[:, features].values
y = final_dataset['label']
X_train, X_test, y_train, y_test = train_test_split (X, y, test_size=0.2, random_s
tate=0)
# Create linear regression object
clf = KNeighborsClassifier(n_neighbors=3)
# Train the model using the training sets
clf.fit(X_train, y_train)
# Make predictions using the testing set
y_pred = clf.predict(X_test)
%time
from sklearn.metrics import classification_report
y_pred = clf.predict(X_test)
print('accuracy %s' % accuracy_score(y_pred, y_test))
print(classification_report(y_test, y_pred))
print(final_dataset)
```

Naïve Bayes

```
From sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
import numpy as np
features = ['second term']
X = final_dataset.loc[:, features].values
y = final_dataset['label']
X_train, X_test, y_train, y_test = train_test_split (X, y, test_size=0.2, random_s
tate=0)
# Create linear regression object
nbc = GaussianNB()
# Train the model using the training sets
nbc.fit(X_train, y_train)
# Make predictions using the testing set
y_pred = nbc.predict(X_test)
%time
from sklearn.metrics import classification_report
y_pred = nbc.predict(X_test)
print(y_pred)
print('accuracy %s' % accuracy_score(y_pred, y_test))
print(classification_report(y_test, y_pred))
print(final_dataset)
```