Technical note

Machine Learning models and feature relevance for student grade prediction

Ali I

Submitted: October 29, 2023, Revised: version 1, February 5, 2023, version 2, March 6, 2024
Accepted: March 14, 2024

Abstract
Student performance evaluation is a major challenge for most educational institutions including colleges and schools. To improve educational experiences and provide a better learning environment for students, it is important to find the factors primarily responsible for student academic performance. Today's educational institutions operate in a more complex environment with multiple modes of delivery, engagement, and assessments available. In this work, we performed a two-fold analysis; first to measure the relevance of these factors when considering student grades and second, to use Machine Learning models to predict student grades. The results were then compared along two different evaluation metrics to establish a framework for student grade prediction. Two publicly available datasets were used for this analysis: one from the college level and the other from the school level. The college level data found a greater effect of student engagement on overall performance, whereas the school level data suggested that features such as study time and parents' job were significant. Romantic relationships were found to affect grades at school level. Two categories of Machine Learning models were used; Linear Models and Ensemble Learning Models. The latter were found to perform well for the grade prediction problem when analyzing college level student grades. Linear models like Linear Regression and Support Vector Regression (SVR) performed better for school level student grades. These results can be used by educators to predict and analyze student performance both at college and school level.

Keywords
Machine Learning algorithms, Artificial Intelligence, Linear models, Decision trees, Ensemble learning, Student grade prediction, Feature relevance in grade prediction

Izhan Ali, Niskayuna High School, 1626 Balltown Road, Niskayuna, NY 12309, USA
izhan.ali08@gmail.com

1 Introduction

With recent advancements in the dissemination of educational content, more student-related information is available in the form of demographics and engagement level. This data can be useful in determining the effect of factors such as student background, demographics, engagement, and interaction levels on overall student performance. This information is advantageous both for students as well as for educational institutions. Students can use it to improve their grades and educational institutions can increase retention rates, build early intervention programs, and provide a better quality of education in general.

The ability to predict student scores before assessments is another powerful tool for teachers and educational institutions. Early intervention steps can be instituted if teachers/educators have a clear understanding of anticipated grades and the features affecting those grades. There is an increasing trend to use machine learning (ML) and artificial intelligence (AI) models to predict student performance. Albreiki et al. (1) provided a comprehensive literature review of the techniques in use. A stream of literature also focuses on model performance when predicting student grades (2). Xu et al. (3) suggested a novel ML approach for predicting student grades. Similarly, Almarabeh et al. (4), performed a detailed analysis and comparison of various classification techniques to predict student performance. Therefore, an effective framework to accurately predict student grades appears to be an addressable ongoing challenge. With the advent of more sophisticated prediction models being used across various domains, student grade prediction problems can utilize those techniques to obtain more accurate results. We therefore used ML/AI techniques to solve this problem. A background about ML/AI frameworks can be found in Lauriere (5).

In this work we addressed the two most relevant challenges in the field of student performance prediction: feature importance and model selection, with a view to more accurately evaluate and predict student performance. We performed our analysis on two real-world datasets that captured student performance at college and school level: 1) For edX courses at Harvard (6), 2) Secondary school data from Portugal (7).

1.1 Contributions

Our contributions are twofold:

1.1.1 *Model Selection*

Student grades were predicted using two different sets of models: 1) Linear Models: Linear Regression, Support Vector Machines and 2) Tree based models: Decision Trees, Random Forest, AdaBoost, Gradient Boost and XGBoost. The results were evaluated using a k-fold cross validation technique. Since we were essentially solving a regression problem, the evaluation metrics used were mean squared error (MSE) and mean absolute error (MAE).

1.1.2 *Feature importance for student grade prediction*

Out of the two categories of Machine Learning models implemented, we additionally analyzed one from each category to find features that had the most significant impact on student performance. The rationale for analyzing more than one model was to check for any common features that significantly impact the prediction results in more than one instance. In general, we

were interested in determining whether student engagement related features were more dominant in impacting student grades, or if student attributes like gender, age etc., had a greater affect.

1.2 Literature Review

Navang et al. (8) performed a comprehensive review of research papers addressing student performance prediction using ML techniques. They also emphasized studies that explored feature importance affecting student results. The most common models used for student performance prediction were Random Forest (RF), Decision Tree (DT), Naïve Bayes (NB), Support Vector Machine (SVM), Artificial Neural Network (ANN), Logistic Regression (LR), and K-Nearest Neighbor (KNN). Kovacic (9) measured the relevance of demographics on student performance. A decision tree-based classifier (CART) was used to predict student grades and achieved 60% accuracy. A decision tree-based approach was also used by Quadri et al. (10). Kotsiantis et al. (11) used an updated version of Naïve Bayes classifier to measure student performance. Lakkaraju et al.(12) suggested a scoring framework with several evaluation metrics to identify students at risk. Their study was conducted across two school districts. Navang et al. (13) used a classification model to find feature relevance for college students. A similar data mining approach to predict college student grades was used by Gil et al. (14). Many other studies in the literature used ML/AI to find features affecting student performance (2, 15, 16). Czibula et al. (17) applied a novel relational association rule mining classification model to predict academic performance. Tran et al. proposed a multiple model strategy to measure performance (18). The objective of our research falls in line with most of the existing research because we utilized multiple ML models to determine student performance. However, we additionally explored features that impact student performance the most. Our contribution is to compare the results across two datasets from different institution types: college versus high school. We further discuss the implications of using ML/AI techniques for student performance tasks and compare the results for the two datasets.

2 Materials and Methods

2.1 Problem Formulation

In this research our objective was to predict student grades and determine which features affected student grades the most. This could be treated as a simple regression problem because the datasets had grades recorded as a numeric/continuous values.

2.1.1 *Performance prediction*

Student performance can be impacted by various features. The final grade received by a student is the most objective way of measuring performance. We used the final grade as the target (value to be predicted) variable in all our models.

2.1.2 *Feature Importance*

In machine learning literature, feature importance is a technique used to determine the relative importance of each feature/factor/variable in a dataset when building a predictive model. In this research we were interested in determining the features that impacted student performance in general. We further divided features into two major categories: 1) Student Attributes represent features that are specific to a

particular student, such as their demographics, age, family details, and their academic history 2) Student Engagement represent features that determine behavioral aspects like how much the student overall engaged with the material, i.e. material used, and hours interacted. For both datasets we determined which set was more dominant.

2.2 Dataset Description and Data Preprocessing

We used two different publicly available datasets for our research (6) and (7). The choice of our datasets was motivated by the fact that different studies in existing literature used datasets from different educational levels. For example, some used high school level grades only while others focused on college level performance. Our objective was to explore the effective usage of ML models as well as to determine feature importance across all educational levels. We selected our first dataset from Harvard college level courses and our second school level dataset from Portugal. We hypothesized that these contrasting datasets could uncover the similarities and differences between the two different student populations. Additionally, the second dataset was populated with grades from two different subjects; Mathematics and Portuguese language. This represented an opportunity to explore the contrast between the features that affected the grades for two different subjects.

2.3 Exploratory Data Analysis (EDA) and Data Preprocessing

For both the datasets, an extensive EDA was performed to look for missing data, duplicate values, and other anomalies. The dataset (6) is collected from Harvard's five different online courses and has 338223 observations and 20 columns. There were no duplicate values found. Missing data is an important issue and we found 11 out of 20 features had missing values. The target variable (grade) in this problem is a continuous value which is already scaled between 0 and 1. Around 25890 grades were missing from the original dataset. Because this is the target variable, we dropped these observations. Figure 1 shows the count of observations for each of the five courses.

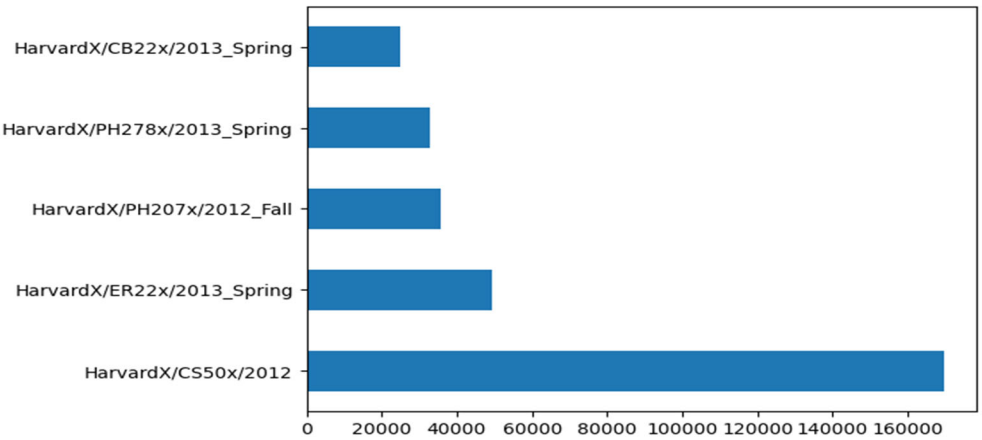

Figure 1. The number of observations (X axis) for five courses (with semester and year), on the Y axis.

For each of the five courses, the average grade value was calculated and shown in Figure 2.

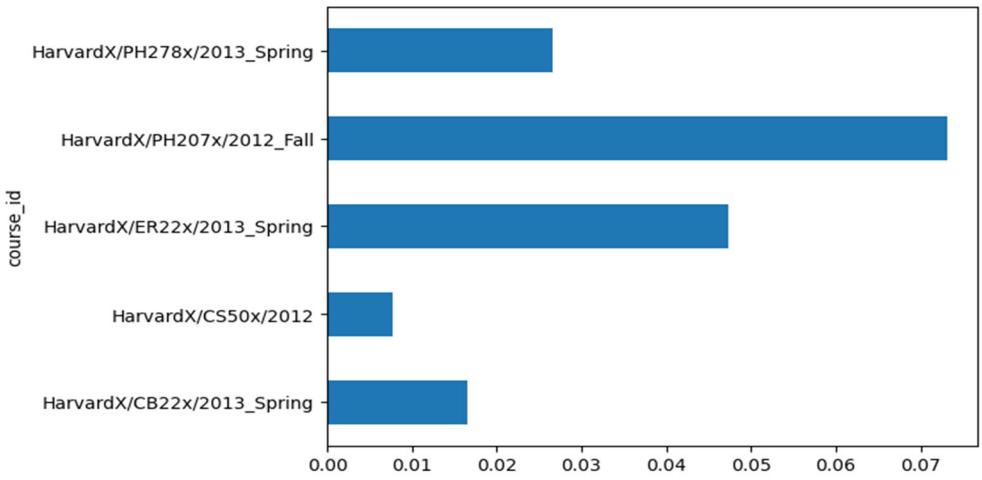

Figure 2. The average grade (X axis) for each course group.

The datasets in (7) were populated with student records from two different secondary Portuguese schools for two subjects: Mathematics and Portuguese language. The mathematics dataset contained 395 observations and 33 columns. The Portuguese language dataset contained 649 observations and 33 columns. There were no duplicate observations. For both datasets, a similar set of EDA and preprocessing steps were performed. For both (6) and (7) any columns with more than 60% missing data were removed because such features carry little to no information and were not useful for the model. Columns with less than 60% missing data were populated with the mean of the rest of the values. For any categorical columns the missing data was populated with the mode. These are common imputation methods used in statistical analysis.

2.4 Feature Engineering (FE)

For the Harvard dataset (6), there were various time related features such as the date a student started the course, and their year of birth. These were used to calculate a new feature called Age. Similarly, the duration of a student's time spent attending the course was calculated using start time and end time variables in the dataset. After we added these new features of Age and Duration to the dataset, the original four features were dropped to avoid redundancy. Next, we analyzed categorical features. These were converted to a numerical format because ML algorithms could only use numbers as inputs. Three categorical variables were found: 'final_cc_cname_DI' – user provided identification based on the region they belonged to, 'LoE_DI' – highest level of education completed, and 'gender' – gender information provided by the student. We used Label Encoding from the sklearn package to encode these variables. Next, we plotted the correlation heatmap to determine any features that had little to no correlation with the target variable. Three of the features with low correlation were dropped. Correlation can detect bivariate relationships between both dependent and independent variables, but sometimes a variance inflation factor (VIF) is preferred as it can show the correlation of a variable with a group of other variables. Therefore, VIF was calculated for the dataset to further establish any multicollinearity among

the features. VIF is a measure of the strength of any linear relationship between two variables. A value of 1-5 is generally associated with low correlation, values of 5-10 are associated with moderate correlation and values > 10 imply that the variables are highly correlated. Multicollinearity can produce estimates of the regression coefficients that are not statistically significant. In this dataset only one variable (registered) was found to have a high VIF of > 20. This was dropped from the dataset. The numeric columns were scaled using a min max scaler from the sklearn package. Min-max scaling transforms features within the range of 0-1. Min-max scaling reduces the effect of outliers on the data values.

The datasets (7) (both for Mathematics and Portuguese Language) were populated with grades from three different quarters for each student. We calculated the average of these grades and used this calculated value as the target variable. Many other studies use two of the mid-year grades as input features to the models. We considered all grades as target values to be able to create a framework for prediction regardless of the (non)availability of grades for any particular quarter or semester. Both datasets contained numerical and categorical features. We used one hot encoding technique to create dummy features for categorical ones. Next, the numerical data was scaled using a min-max scaler. This scaling technique is useful because it removes any biases that might propagate into the model due to outliers. Before fitting the data to the models, we calculated correlations of variables/features with the target variable and dropped the features that showed correlation values < 0.01. Additionally, we calculated the VIF for both the datasets. Both datasets presented with high VIF for six features: mother's education, father's education, quality of family relationships, school principal, extra education support at school and whether the student was interested in pursuing higher education. We dropped these 6 features from further analysis because the same information was available in other family and school related variables/features. The table in Appendix shows the VIF values for the features after removing variables with high VIF values.

2.5 Modeling

In this section we briefly describe the two categories of models used for student grade prediction. In-depth details of these models can be found in (19). One model from each category was further analyzed for feature importance. We also describe the packages used to implement these models in Python. To maintain consistency the same hyperparameters (default) were used for all 3 datasets.

2.5.1 *Linear Models*

We used two linear models, Linear Regression and Support Vector Regression. Linear regression is a machine learning technique used for modeling the relationship between a dependent variable and one or more independent variables by fitting a linear equation to the observed data. Linear regression assumes a linear relationship, and it may not capture complex, non-linear patterns in the data. It is also sensitive to outliers and can be affected by multicollinearity (high correlation between independent variables). In our analysis, we capture the learned co-efficient of the model to determine feature relevance and fit each of the datasets to predict the target variable (grades). For implementation of linear regression, we used the module linear_model from the sklearn package in Python. The class is called LinearRegression. The input parameters

for this class are fit_intercept that was set to True, copy_X set to True, n_jobs and positive set to False. We used the default version of this class (for all the three datasets).

Support Vector Regression (SVR) is a machine learning technique used for regression tasks. It is a variation of Support Vector Machines (SVM), which are commonly used for classification problems. While SVM is used for classification by finding a hyperplane that maximizes the margin between classes, SVR is used for regression by finding a hyperplane that best fits the data while minimizing prediction errors. To implement SVR for all 3 datasets we used the class LinearSVR from sklearn.svm in python. This class has more flexibility in the choice of loss functions and scales as compared to its counterpart SVR (also found in the same library). This class also supports dense and sparse inputs. The input parameters were set to their default values for all implementations i.e. epsilon=0.0, tol=0.0001, C=1.0, loss='epsilon_insensitive', fit_intercept=True, intercept_scaling=1.0, dual='warn', verbose=0, random_state=None, max_iter=1000.

2.5.2 *Tree Based/ Ensemble Models*

We used a variety of Tree based/Ensemble models including Decision Tree, XGBoost, Random Forest, AdaBoost and Gradient Boosting Machines. A decision tree is a machine learning algorithm used for both classification and regression tasks. It is a tree-like structure where an internal node represents a feature or attribute, the branches represent decisions or rules, and the leaves represent outcomes or class labels or numerical values. However, they are also prone to overfitting, especially when the trees are allowed to grow deep, and they may not always capture complex relationships in the data, but these are the building blocks of most ensemble techniques. For implementation of this model we used the class DecisionTreeRegressor from sklearn.tree. This class provides a lot of flexibility in terms of input parameters. For example, the criterion parameter uses different types of loss functions. We used the default 'squared_error' for all 3 datasets. The splitter variable was set to the best value. The setting for all input parameters: sklearn.tree.DecisionTreeRegressor(data, criterion='squared_error', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, ccp_alpha=0.0, monotonic_cst=None)

"Extreme Gradient Boosting," or XGBoost, is an ensemble learning method, specifically a gradient boosting algorithm, which means it builds an ensemble of decision trees to make predictions. It has set the standard for gradient boosting algorithms and is often a top choice when dealing with structured data (20). We used the model for grades prediction and analyzed the weights/coefficients learned by the predictive model to determine feature relevance. XGBoost is a powerful library with three sets of parameters: general, booster and learning task parameters. We installed and used the Python package for XGBoost. A detailed list of XGBoost parameters can be found under the official documentation (22). We did not change any of the default parameter settings for each dataset.

Random Forest is an ensemble method that combines the predictions of multiple decision trees to make more accurate and stable predictions. It leverages the wisdom of crowds, where the collective decision of many trees is often

better than that of a single tree (21). We fit this to both the datasets and analyzed the coefficients that have a greater impact on grade prediction. The implementation was done using sklearn as shown: sklearn.ensemble.RandomForestRegressor(n_estimators=100, *,criterion='squared_error', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=1.0, max_leaf_nodes=None, min_impurity_decrease=0.0, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, ccp_alpha=0.0, max_samples=None, monotonic_cst=None)

AdaBoost is primarily known as a classification algorithm, but it can be adapted for regression tasks by modifying the original algorithm. This variation is called AdaBoost for Regression. It is designed to fit regression models instead of classifiers and works by combining a set of weak regression models into a strong regression model. For each of our datasets we used the following configuration of this package in python (imported from sklearn.ensemble): sklearn.ensemble.AdaBoostRegressor(estimator=None, *, n_estimators=50, learning_rate=1.0, loss='linear', random_state=None)

Gradient Boosting Machines is a machine learning technique for both regression and classification tasks that builds a powerful predictive model by combining the predictions of multiple "weak" models, typically decision trees. It is an ensemble learning method, which means it forms an ensemble of models to create a stronger and more accurate predictor. The implemented class belongs to the sklearn.ensemble family of models and we used the following configuration for our implementation of this model: sklearn.ensemble.GradientBoostingRegressor(*, loss='squared_error', learning_rate=0.1, n_estimators=100, subsample=1.0, criterion='friedman_mse', min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_depth=3, min_impurity_decrease=0.0, init=None, random_state=None, max_features=None, alpha=0.9, verbose=0, max_leaf_nodes=None, warm_start=False, validation_fraction=0.1, n_iter_no_change=None, tol=0.0001, ccp_alpha=0.0)

2.6 Training and Testing

Training is an important aspect of machine learning models. We were working with two different datasets with a different number of features and different number of observations. The datasets in (7) did not have many observations and were therefore prone to overfitting. The college data in (6) was large and could potentially require a longer training time. Taking both the challenges into account, we built a uniform training strategy across both datasets. We used k-fold cross validation technique to train and evaluate the models. Under this technique the data is split into k-subsets called folds. The model is trained and evaluated k-times using a different subset/fold as the test (or validation) set each time. Performance metrics are averaged across all folds to estimate the model's generalization performance to unseen data. This is a more robust technique for model evaluation than a simple train test split strategy. The choice of k (number of folds) in k-fold cross-validation depends on the dataset size, desired bias-variance trade-off, and computation times.

For the HarvardX data (6) we selected k as 20 because of the large number of observations (309,638). For the (7) dataset we selected k as 5

(both for Math and Portuguese data) because the number of observations was less than 650. The training dataset was used to fit the model and the test (also called validation) dataset was used to predict and compare results. The target variable (variable to be predicted) was the grade of students. For the Harvard dataset (6) the grade value was already provided as a continuous value between 0-1. For the (7) datasets, we calculated the target variable by taking the average of three different grades provided for each student as discussed previously.

2.7 Model Evaluation

We were solving a regression problem of determining student grades. Prediction results for such regression-based problem can be evaluated using different evaluation metrics, the Mean Squared Error and the Mean Absolute Error. The Mean Squared Error is the average of the squared difference between predicted and actual values. This error penalizes outliers disproportionately because the differences are squared. The Mean Absolute Error is the average absolute difference between predicted and actual values.

3 Results and Discussion

Our results reveal Linear models perform better for school level data (both Math and Language grades (7)) and Ensemble models work better with college level data (6).

3.1 Feature Importance Analysis

The feature importance analysis was conducted as part of the model building process. In addition to grade prediction, we analyzed significant features that contributed to these results. The exact results of our findings can be found on the link provided under the materials section below. Here we specify some of the top variables/features that were found most relevant by the models used and the category to which they belong, i.e., Student Attributes or Student Engagement. A detailed description of each feature can be found on the respective data source websites. Additionally, we picked one model from each of the two categories of models used for student grade prediction. The first choice was Linear Regression for both datasets because this is the most widely used model for regression problem. The second choice is XGBoost for college level data (6) because this is a larger dataset and XGBoost works better for large datasets and Random Forest for school level data (7) because these are smaller datasets and Random Forest models fits smaller datasets better. Figure 3 -5 depict all the results in terms of the model coefficients (on the y -axis) and features (x-axis).

For college level student performance (6), Linear Regression shows a higher impact of Student Attributes like gender, level of education etc., on the overall grades. XGBoost reveals a higher effect of the variables that fall under our student engagement category. In summary, for college level grades there is an impact of features from student attributes and their engagement uniformly. The results for both the models are plotted in Figure 3 below. The categorical variables were encoded and each of the encoded values can be seen on the x-axis.

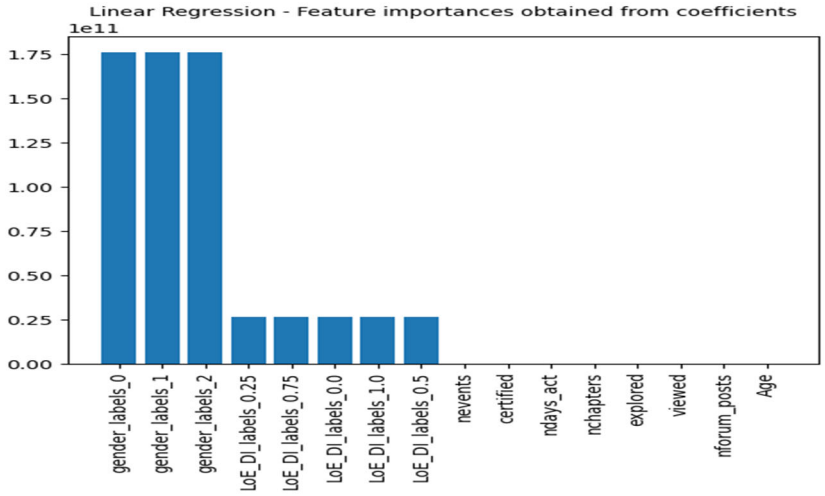

Figure 3a. Feature Importance plotted for HarvardX data for Linear Regression.

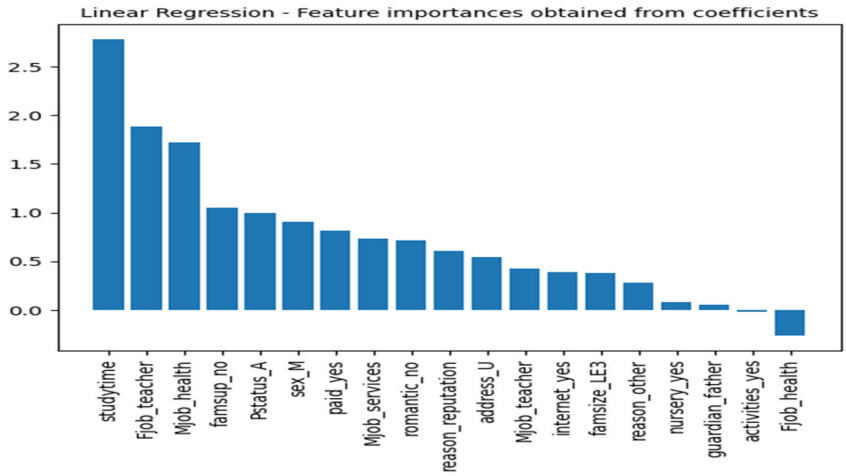

Figure 3b. Feature Importance plotted for HarvardX data for XGBoost.

For secondary school-level math performance, both models produce results that have both Student Attributes and Student Engagement related features. One point that stands out is that all results reveal the relevance of study time (number of hours spent) as a high impact feature for math grades. Linear regression shows that Father's job as a teacher and mother's job in healthcare are also relevant. Analyzing coefficients of the Random Forest model reveals that romantic relationship as a feature impacts math grade. Results are shown in Figure 4.

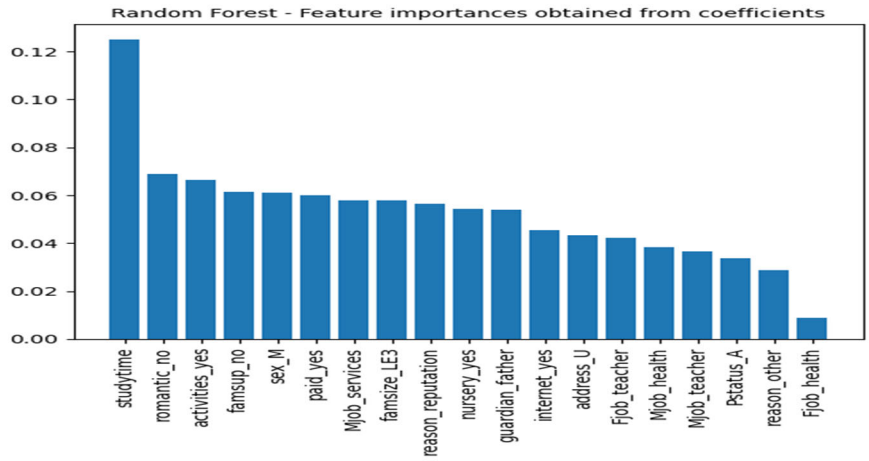

Figure 4a. Feature Importance plotted for Secondary school math data for Linear Regression.

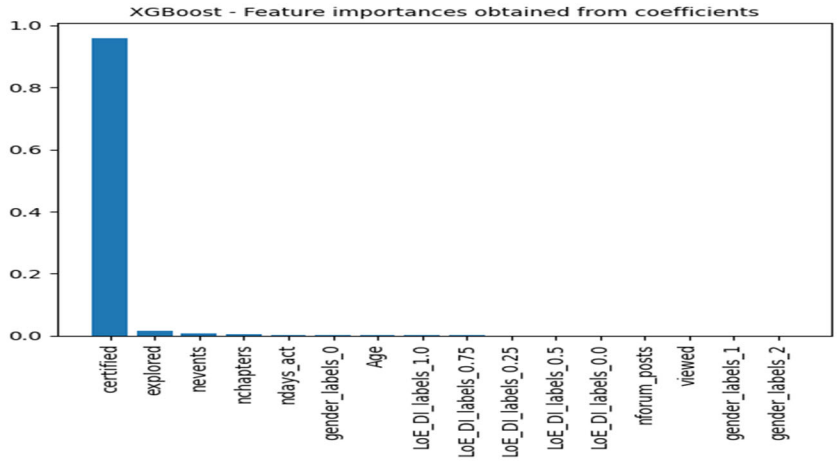

Figure 4b. Feature Importance plotted for Secondary school math data for Random Forest Regressor.

Study time was found relevant for Portuguese (language) grades as well by the two models. School type, mother's job in health and father's job as a teacher are other prominent factors. Female gender is found to be relevant by the linear regression model for language grades. Male gender appears to be one of the top 5 significant features found by the Random Forest regressor for math grades. This may be linked to the general bias that exists where female students perform better at language-related subjects versus male students for math or science. Some of the other impactful features include extra support provided by the school, family educational support, which school the student attended, and family support. At school level family related attributes have a greater impact on student grades. This is summarized in Figure 5 below.

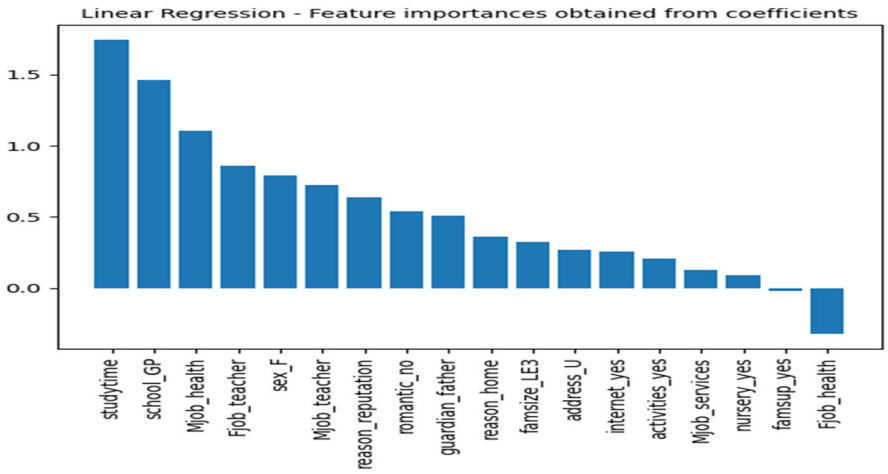

Figure 5a. Feature Importance plotted for Secondary school Portuguese data for Linear Regression.

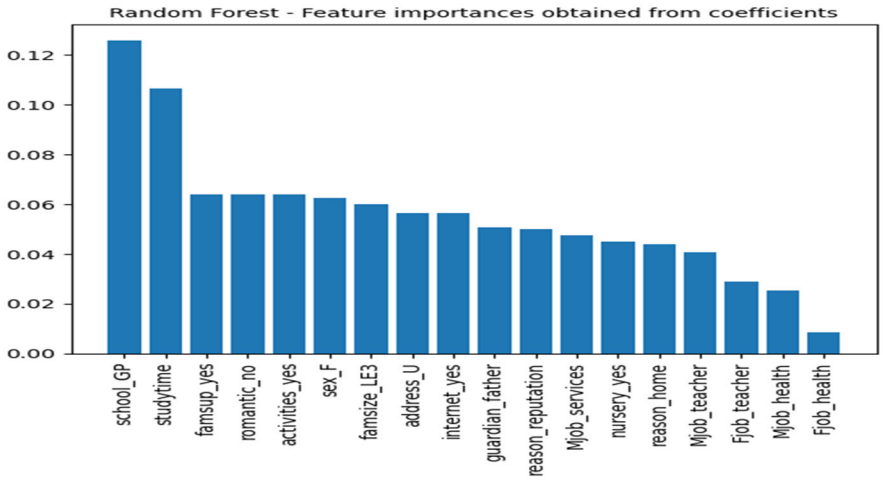

Figure 5b. Feature Importance plotted for Secondary school Portuguese data for Random Forest Regressor.

Student Engagement attributes have a higher impact for students at college level. This is an important result because family attributes seem to have a lower impact on student performance at college level when compared to school level. Another factor that impacts language grades found by Random Forest model is romantic relationships. These results are summarized in Table 1.

Table 1. Best performing models with features that affected those models most.

Dataset	Model	Results
(6) HarvardX	Linear Regression	Student Attributes (gender, level of education)
	XGBoost (ensemble model)	Student Engagement (certified, explored, number of events, number of chapters)
(7) Math	Linear Regression	Student Attributes and Engagement (study time, father's job as teacher, mother's job in health)
	Random Forest (ensemble model)	Student Attributes and Engagement (study time, romantic relationship, activities, gender as male, family support)
(7) Portuguese	Linear Regression	Student Attributes and Engagement (study time, school, mother's job in health)
	Random Forest (ensemble model)	Student Attributes and Engagement (school, study time, family support, romantic relationships)

3.2 Performance prediction analysis

In this section we describe the results of our grade prediction analysis.

For dataset (6), across both evaluation metrics, the XGBoost algorithm was the top performer with a MSE of 0.000887 and MAE of 0.005679 as shown in Table 2. XGBoost is an optimized version of ensemble learning and is known to perform well on large datasets.

Table 2. Results for dataset 6.

Model	Results (MSE, MAE)
Linear Regression	0.001651, 0.010721
Support Vector Regression	0.001878, 0.007765
Decision Tree	0.001782, 0.007391
XGBoost	0.000887, 0.005679
Random Forest	0.000918, 0.005757
AdaBoost	0.001770, 0.011912
Gradient Boost	0.000916, 0.006000

For dataset (7) mathematics scores, the Linear Regression model resulted in the lowest error across both metrics when predicting school student math grades as shown in Table 3. SVR, Random Forests and Gradient Boost models also showed comparable performance. Random Forest is better suited to datasets with more noise.

Table 3. Results for dataset (7) Mathematics scores.

Model	Results (MSE, MAE)
Linear Regression	13.058390, 2.931116
Support Vector Regression	13.413584, 2.974642
Decision Tree	26.328871, 4.186075
XGBoost	18.914289, 3.446513
Random Forest	13.652215, 2.966616
AdaBoost	13.928979, 2.991857
Gradient Boost	13.374467, 2.946342

For dataset (7) Portuguese scores, the Support vector regression (SVR) yielded the best results for language grade prediction as shown in Table 4. This is a smaller dataset with linear relationships. SVR is known to handle overfitting problems in smaller datasets better which could potentially be the reason for its better performance.

Table 4. Results for dataset (7) Portuguese scores.

Model	Results (MSE, MAE)
Linear Regression	6.829265, 2.020196
Support Vector Regression	6.790504, 2.014802
Decision Tree	13.932073, 2.853832
XGBoost	10.172614, 2.501624
Random Forest	7.703926, 2.135062
AdaBoost	7.420530, 2.104285
Gradient Boost	7.308372, 2.058145

4 Conclusion, Limitations, Future Research

The regression problem of predicting student grades was solved for two different student groups: college and secondary school students. Within the group of secondary school students, we predicted grades for math and language. Two sets of Machine Learning Models were used for this analysis: Linear Models (Linear Regression, Support Vector Regression) and Tree based Ensemble Models (Decision Tree, Random Forest, XGBoost, AdaBoost, Gradient Boost). Additionally, feature relevance for this problem was also explored. For college grades, the XGBoost algorithm resulted in the lowest MSE and MAE (0.000887, 0.005679), for school math grades, Linear Regression resulted the lowest error rate i.e., MSE and MAE (13.058390, 2.931116). Language grades were best predicted by Support Vector Regressor, with a 6.790504 MSE and a 2.014802 MAE. From this analysis we concluded that Linear models in general are better suited for school grade prediction. On the other hand, for college level data, the Tree based Ensemble models were better suited. These results could be explained by the fact that college level datasets were larger, and more information was available with the advent of online learning. School class sizes are usually small with less information available. Hence, Linear Models perform better for this group. The features that affected the school level grades the most were found to be the student's study time and the parents' job type. Gender affected language and math grades. School level grades were also affected by romantic relationships. For college level

grades, student engagement related features had the most impact. This is important information for college level educational institutions when designing their courses, because regardless of the student's background, better engagement may lead to improved performance.

We were able to uncover some of the important similarities and distinctions between student performance from two different student populations. We applied some of the most popular machine learning techniques/models to perform student grade prediction and feature importance. Hyperparameter tuning of these models may sometimes lead to better insights. We did not perform an in-depth parameter search and hyperparameter tuning for the models, rather the best default model was used for analysis. This was because the objective of this study was to devise a fundamental framework for educators to be able to predict student grades at all education levels by comparing two major categories of Machine Learning models. In future work, we will measure the effect of hyperparameter tuning. With the advent of deep learning, large datasets are better represented by neural networks given their ability to capture non-linearities in data. Deep learning models were not explored in this research because the school level datasets were not large enough for these data hungry models. As larger datasets are made available, deep learning models can be used to get more granular insights into features and prediction tasks. Finally, we need more comprehensive student and course related data to perform a more in-depth analysis of student performance evaluation. For example, relationship related information was not available for college data although romantic relationships were found to have an effect in the other two datasets. Several other obviously and self-evident important factors like instructor effectiveness, peer influence and school rigor, among others, were not explored in this research as these were not captured by any of the three datasets used. Lastly, data collection at different grades in both schools and colleges may represent another area for future research.

References

1. Albreiki B, Zaki N, Alashwal H. A systematic literature review of student' performance prediction using machine learning techniques. Educ Sci. 2021;11(9). http://doi.org/10.3390/educsci11090552

2. Ma X, Zhou Z. Student pass rates prediction using optimized support vector machine and decision tree. In: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC). IEEE; 2018:209-215. https://ieeexplore.ieee.org/abstract/document/8301756

3. Xu J, Moon KH, Van Der Schaar M. A machine learning approach for tracking and predicting student performance in degree programs. IEEE J Sel Top Signal Process. 2017;11(5):742-753. https://ieeexplore.ieee.org/document/7894238

4. Almarabeh H. Analysis of students' performance by using different data mining classifiers. Int J Mod Educ Comput Sci. 2017;9(8):9. https://doi.org/10.5815/ijmecs.2017.08.02

5. Lauriere J-L. Problem Solving and Artificial Intelligence. Prentice-Hall, Inc.; 1990.

6. HarvardX. HarvardX Person-Course Academic Year 2013 De-Identified dataset, version 3.0. Published online 2013. http://doi.org/10.7910/DVN/26147

7. Cortez P. Student Performance. Published online 2014. https://doi.org/10.24432/C5TG7T

8. Nawang H, Makhtar M, Hamzah W. A systematic literature review on student performance predictions. Int J Adv Technol Eng Explor. 2021;8(84):1441-1453. https://doi.org/10.19101/IJATEE.2021.874521

9. Kovacic Z. Early prediction of student success: Mining students' enrolment data. Published online 2010. https://doi.org/10.28945/1281

10. Quadri MM, Kalyankar N V. Drop out feature of student data for academic performance using decision tree techniques. Glob J Comput Sci Technol. 2010;10(2):2-5. https://globaljournals.org/GJCST_Volume10/gjcst_vol10_issue2_ver1_paper7.pdf

11. Kotsiantis S, Patriarcheas K, Xenos M. A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education. Knowledge-Based Syst. 2010;23(6):529-535. https://doi.org/10.1016/j.knosys.2010.03.010

12. Lakkaraju H, Aguiar E, Shan C, et al. A machine learning framework to identify students at risk of adverse academic outcomes. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ; 2015:1909-1918. http://doi.org/10.1145/2783258.2788620

13. Nawang H, Makhtar M, Shamsudin SNW. Classification model and analysis on students' performance. J Fundam Appl Sci. 2018;9(6S):869. http://doi.org/10.4314/jfas.v9i6s.65

14. Gil PD, da Cruz Martins S, Moro S, Costa JM. A data-driven approach to predict first-year students' academic success in higher education institutions. Educ Inf Technol. 2021;26(2):2165-2190. http://doi.org/10.1007/s10639-020-10346-6

15. Aiken JM, De Bin R, Hjorth-Jensen M, Caballero MD. Predicting time to graduation at a large enrollment American university. PLoS One. 2020;15(11):e0242334. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0242334

16. Hussain M, Zhu W, Zhang W, Abidi SMR. Student engagement predictions in an e-learning system and their impact on student course assessment scores. Comput Intell Neurosci. 2018;2018. https://www.hindawi.com/journals/cin/2018/6347186/

17. Czibula G, Mihai A, Crivei LM. S PRAR: A novel relational association rule mining classification model applied for academic performance prediction. Procedia Comput Sci. 2019;159:20-29. https://doi.org/10.1016/j.procs.2019.09.156

18. Tran TO, Dang HT, Dinh VT, Truong TMN, Vuong TPT, Phan XH. Performance prediction for students: A multi-strategy approach. In: BULGARIAN ACADEMY OF SCIENCES; 2017. http://doi.org/10.1515/cait-2017-0024

19. (MIT) K murphy. Probabilistic Machine Learning An Introduction. https://probml.github.io/pml-book/book1.html

20. Chen T, Guestrin C. XGBoost: A scalable tree boosting system. Proc ACM SIGKDD Int Conf Knowl Discov Data Min. 2016;13-17-785-794. https://doi.org/10.1145/2939672.2939785

21. Reza M, Miri S, Javidan R. Random Forests. Int J Adv Comput Sci Appl. 2016;7(6):1-33. https://doi.org/10.14569/ijacsa.2016.070603

22. https://xgboost.readthedocs.io/en/release_0.90/parameter.html#parameters-for-tree-booster

Supplementary File

https://github.com/IzhanAli08/MyResearch

Appendix

Table – HarvardX Data

Variables	VIF
viewed	1.155765
explored	2.626144
certified	1.830941
nevents	1.92416
ndays_act	2.433233
nchapters	3.363034
nforum_posts	1.034541
Age	1.046277
Duration	1.148798
final_cc_cname_DI_labels	1.123574
LoE_DI_labels	1.003009
gender_labels	1.036964

Table – Math and Portuguese Data

Variables	VIF for Math	Variables	VIF for Portuguese
studytime	2.80481	studytime	2.618255
sex_M	2.20134	school_GP	3.752235
address_U	4.492624	sex_F	2.613552
famsize_LE3	1.514097	address_U	3.832599
Pstatus_A	1.192325	famsize_LE3	1.457318
Mjob_health	1.313987	Mjob_health	1.266234
Mjob_services	1.636123	Mjob_services	1.485736
Mjob_teacher	1.604221	Mjob_teacher	1.437371
Fjob_health	1.166196	Fjob_health	1.142836
Fjob_teacher	1.199716	Fjob_teacher	1.176902
reason_other	1.196644	reason_home	1.515838
reason_reputation	1.537563	reason_reputation	1.57527
guardian_father	1.386364	guardian_father	1.351831
famsup_no	1.780959	famsup_yes	2.679765
paid_yes	2.169123	activities_yes	2.001419
activities_yes	2.121672	nursery_yes	4.475701
nursery_yes	4.428559	higher_yes	7.966936
internet_yes	5.746401	internet_yes	4.516066
romantic_no	2.706951	romantic_no	2.686594