Using Machine Learning to Determine Factors Influencing Undergraduate Student Success

By Jared Coffelt

Abstract

Understanding factors leading to student academic success is very important. This project deploys support vector classifiers, logistic regression with cross-entropy loss, and bootstrap aggregation models on a dataset of over 4,000 student records to create a model able to predict student success and identify the most influential determinants of that success. It was found that the models all produced accuracy scores around 75%, even with varying parameters. Additionally, the top two contributing factors are the number of credits approved in the first and second semesters, indicating that perhaps course load is more indicative of a student's success than other socioeconomic or demographic factors.

1. Introduction

Education–the ability to teach useful skills, to impart valuable wisdom to future generations is essential to maintaining a prosperous society. However, not everyone finds success in education, the reasons for which are not entirely obvious or well understood. Machine learning techniques, as they have grown more advanced in recent years, might serve to recognize previously unseen relationships, and help identify factors that might have the largest impact on student success.

This paper utilizes a dataset composed of numerous features that might explain a student's academic outcome. It applies three machine learning models for classification problems to the data, optimizes them, selects the best performing, and reports the most influential features.

2. Dataset

The dataset used for this paper, "Predict students' dropout and academic success" was retrieved from the UC Irvine Machine Learning Repository (Realinho et al., 2021). The data was originally used by Martins et al. (2021) "to identify students at risk of failure at an early stage of their academic path, so that strategies to support them can be put into place" (p.166). The data is pulled from several databases on the Polytechnic Institute of Portalegre (IPP) in Portugal. The data covers records of undergraduate students enrolled in a variety of majors between the 2008/09 and 2018/19 academic years and contains a variety of demographic, socio-economic and academic variables. In total, the dataset contains 4,424 records, and 36 features. The target variable, where students are classified based on status, include the classes of 'Dropout', 'Enrolled', and 'Graduate' as corresponding to 'Failure', 'Relative Success', and 'Success' respectively (Martins et al., 2021, p.169).

As given, the dataset had no missing values (Realinho et al., 2021), so no attempt was made to fill data gaps. Additionally, all the features, even the categorical variables, were given in numerical forms so that they could be read by the model. The target variable, however, was given as the nominal value and so was replaced with numeric values (i.e. 'Dropout', 'Enrolled', and ‘Graduate’ were replaced with the values of 0, 1, and 2. respectively). To confirm that the data was fairly evenly distributed among the classes, the distribution of target values was graphed in a histogram as seen in Figure 1. As the data seemed fairly evenly distributed, no further data manipulation was conducted.

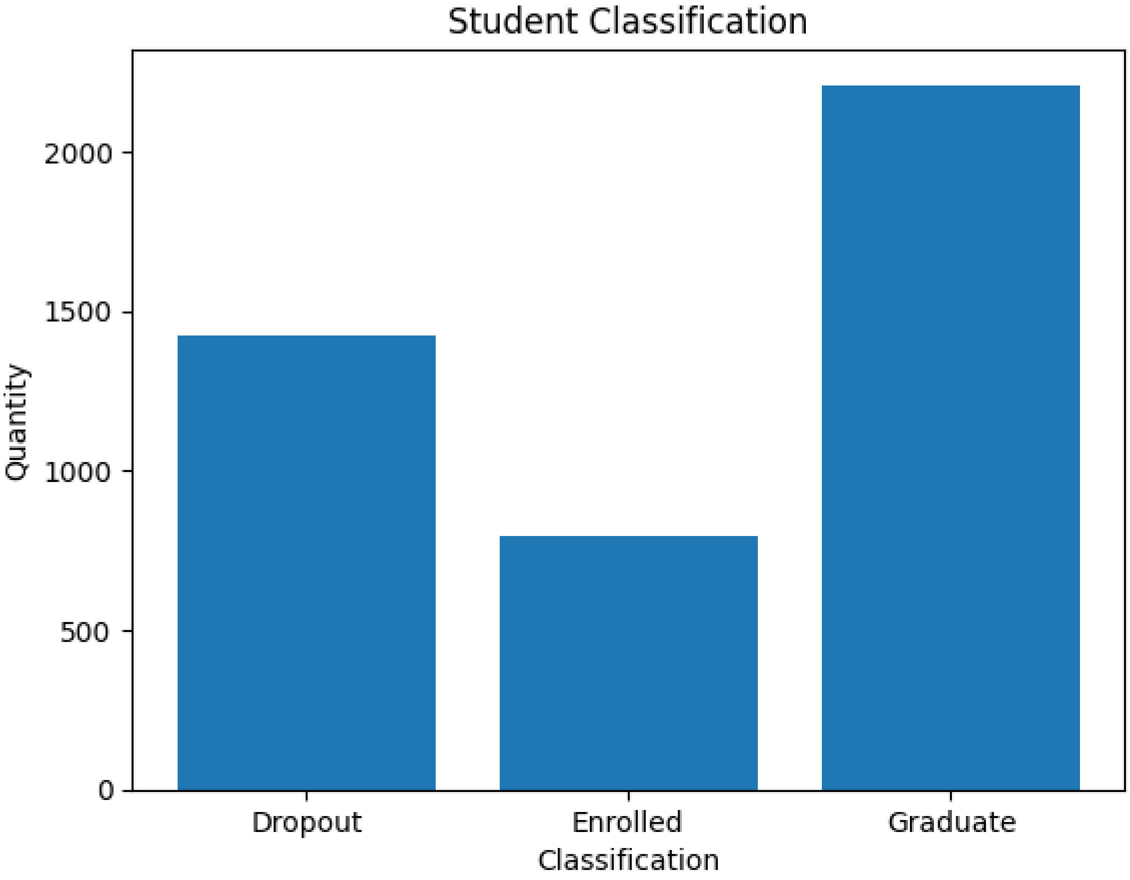

Figure 1. Number of students with a status of having dropped out, still enrolled, or graduated after a course.

3. Model Development

For this paper, three different classification models were used to compare against one another. These include support vector machines (classifiers), Softmax regression, and bootstrap aggregation.

3.1. Selecting Models

Support Vector Machine/Support Vector Classification (SVC)

SVC was chosen initially based on the scikit-learn algorithm cheat sheet which states that for a dataset containing between 50 and 100,000 samples that is predicting a category based on labeled data, linear SVC should first be attempted (Scikit-learn, n.d.d). Other resources indicate that support vector machines are appropriate for problems with high-dimensionality, and even in instances where the number of samples is outnumbered by the number of dimensions (Scikit-learn, n.d.b) This seems applicable to this dataset given that there are moderate number of features and not an extremely large number of samples.

As such, the assumption was made that this would be the best performing model, specifically the linear SVC. Other methods were tested using a variety of C and gamma variations, but the linear SVC always outperformed, although only slightly. The linear SVC model was tested with C values of 0.5, 1.0, 5.0, and 10.0.

Logistic (Softmax) Regression

Logistic Regression was chosen as the second model, but was not expected to perform well. Logistic regression is primarily for binary classification (Bortnik, 2020a), and this dataset has three classes. While several estimators employing a one-vs-rest approach could be used, for a more efficient model, Softmax regression–which employs cross-entropy loss to allow the model to solve for all classes simultaneously (Bortnik, 2020a)--was used instead. The sklearn LogisticRegression model was used with the parameter of 'multi_class' set to 'multinomial' to enable cross-entropy loss and thus Softmax regression (Scikit-learn, n.d.g). As this method was assumed to be inferior to the SVC model (Scikit-learn, n.d.d), and presents few parameters to change (Bortnik, 2020a; Scikit-learn, n.d.g), only the C value was varied from 0.5, 1.0, 5.0, and 10.

Bootstrap Aggregation

The final method employed is bootstrap aggregation (bagging), an ensemble estimator (Scikit-learn, n.d.e). While considered a more "last-ditch" effort for this dataset (Scikit-learn, n.d.d), and thus assumed inferior to the SVC model, using ensemble methods can reduce variance and thus improve the predictive performance of the model (Bortnik, 2020.b; Scikit-learn, n.d.e). The standard DecisionTreeClassifier (Scikit-learn, n.d.e) estimator was used, and to test different performance, the number of estimators was varied between 5, 10, 15, and 20.

3.2. Training the Models

To train and test the models, input data was split into 70% training data and 30% test data, with each model being tested on the same data. To further improve results, especially with a moderately-sized data set (Lozinski, 2023), kfold cross validation was used. 5 folds were incorporated (k=5), with additional folds not seeming to increase performance.

As mentioned previously, several variations of each model were tested by adjusting appropriate parameters. With each variation, the accuracy of the best fold for each model was returned. The best model was then determined to be the model with the highest accuracy.

3.3. Model Verification

Once the best model was determined, several additional steps were taken to examine the data. First, a confusion matrix was generated. From this, precision and recall values were calculated for each class.

Next, features were ranked. As the 'feature_importances_' function was not available for these models (Scikit-learn, n.d.b; Scikit-learn, n.d.c; Scikit-learn, n.d.e; Scikit-learn, n.d.g; Scikit-learn, n.d.h), permutation importance was used instead. This method tests the importance of a feature to a model by randomly shuffling a feature's values, breaking the relationship between the feature and the target (Breiman, 2001 as referenced by Scikit-learn, n.d.a). The permutation importance is then calculated as the decrease in the scoring metric, with the larger the permutation importance meaning the feature is more important (Breiman, 2001 as referenced by Scikit-learn, n.d.f).

The permutation importances of the features were then summed together, and each feature contributing less than 1% of the total importance was removed (Scikit-learn, n.d.a). To verify that these features were the largest contributors, the best performing model was retrained on this reduced training data, using the same kfold procedure as before (although the random selection likely varied). The model was then tested and the resulting accuracy compared to the others.

4. Results

The accuracy scores for each of the models are listed in Table 1.

Table 1. Accuracy scores for each model under varying parameters for two separate runs with training and test data randomized differently.

Run 1	**C**	**0.5**	**1**	**5**	**10**
	SVC	0.758	0.758	0.759	0.759
	Logistic Regression	0.756	0.756	0.755	0.755
	n_estimators	**5**	**10**	**15**	**20**
	Bagging	0.723	0.739	0.753	0.759
Run 2	**C**	**0.5**	**1**	**5**	**10**
	SVC	0.776	0.776	0.777	0.777
	Logistic Regression	0.777	0.776	0.777	0.778
	n_estimators	**5**	**10**	**15**	**20**
	Bagging	0.734	0.761	0.758	0.771
Average of Runs 1 and 2	**C**	**0.5**	**1**	**5**	**10**
	SVC	0.767	0.767	0.768	0.768
	Logistic Regression	0.766	0.766	0.766	0.766
	n_estimators	**5**	**10**	**15**	**20**
	Bagging	0.729	0.750	0.755	0.765

The greatest score during Run 1 for the SVC model was 0.759 (C=5,10), for the Logistic Regression model was 0.755 (C=0.5,1.0), and for the Bagging model was 0.759 (n_estimators =20). Since the SVC model and the Bagging model both showed the highest accuracy, and in a subsequent run, the Logistic Regression model showed the highest accuracy for C=10 at 0.778, a complete second run was conducted, as shown in Table 1. As no clear strongest model could be determined, an average for each variation was conducted. This was based on other runs (not recorded), where SVC was consistently the best performing model. From the averaged values, the highest scoring models were the SVC models with both C = 5 and C = 10 scoring 0.768. Arbitrarily, a C value of 10 was used for further analysis. Note, that for further validation, the training and test data was randomized again and so the results are not representative of the exact model but a close approximation.

This model's confusion matrix is shown in Figure 2. As can be seen, the number of true positives for 'Graduate' are very large, and those for 'Dropout' are larger as well. On the other hand, the true positives for 'Enrolled' are comparatively small.

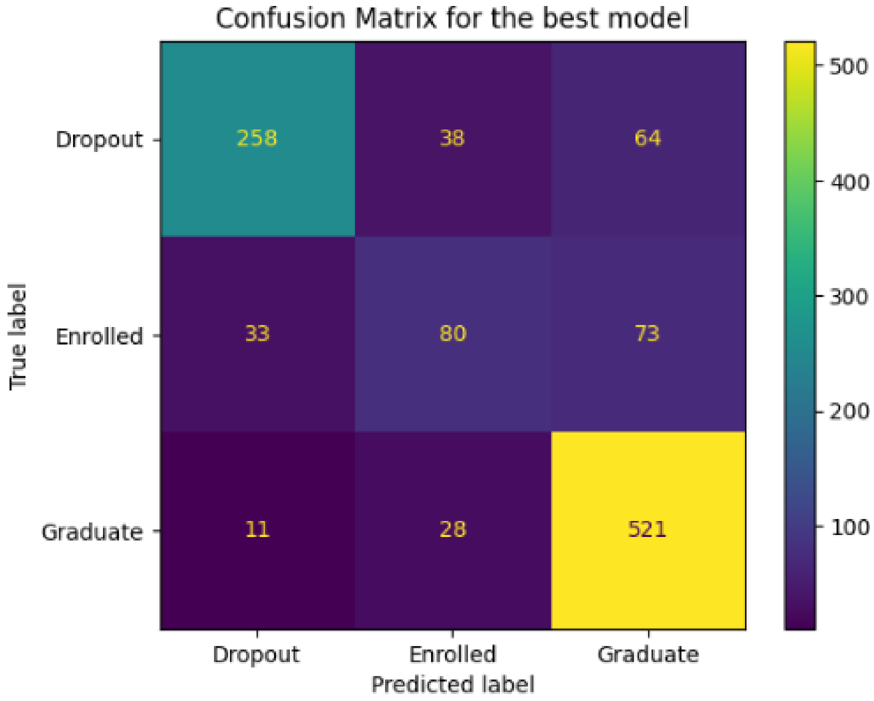

Figure 2. Confusion matrix for the selected model, linear SVC with C = 10

The values for precision for class *i*, as calculated using Equation 1,

$$Precision_i = \frac{True\ positive\ guesses\ of\ i}{True\ positive\ guesses\ of\ i + False\ positive\ guesses\ of\ i}, \text{(Gupta, 2020)} \qquad (1)$$

and recall for class *i* (as calculated using Equation 2)

$$Recall_i = \frac{True\ positive\ guesses\ of\ i}{True\ positive\ guesses\ of\ i + False\ negative\ guesses\ of\ i},\ \text{(Gupta, 2020)} \qquad (2)$$

are shown in Table 2.

Table 2. Precision and recall values for each class

Target	**Precision**	**Recall**
Dropout	0.854	0.720
Enrolled	0.548	0.430
Graduate	0.792	0.930

The results of the permutation importance analysis are shown in Table 3, with two features, ‘Curricular Units 2nd semester (approved)’ and ‘Curricular Units 1st semester (approved)’ contributing most significantly to the predictive power.

Table 3. Permutation importance of each feature, in order from smallest to largest, for linear SVC, C = 10

Feature	Importance		Feature	Importance
Curricular units 1st sem (without evaluations)	-0.0022		Curricular units 2nd sem (without evaluations)	0.0024
Course	-0.0016		Curricular units 1st sem (evaluations)	0.0025
Inflation rate	-0.0016		Mother's occupation	0.0029
Curricular units 1st sem (grade)	-0.0013		Previous qualification	0.0031
Educational special needs	-0.0009		Curricular units 2nd sem (credited)	0.0040
Previous qualification (grade)	-0.0004		Displaced	0.0043
GDP	0.0000		Debtor	0.0047
Father's occupation	0.0005		Gender	0.0052
Application mode	0.0007		Age at enrollment	0.0054
Admission grade	0.0009		Curricular units 1st sem (enrolled)	0.0060
Mother's qualification	0.0011		Curricular units 2nd sem (evaluations)	0.0060
Marital status	0.0011		Unemployment rate	0.0085
Daytime/evening attendance	0.0013		Curricular units 1st sem (credited)	0.0121
Scholarship holder	0.0013		Curricular units 2nd sem (grade)	0.0250
Nacionality	0.0016		Tuition fees up to date	0.0458
Father's qualification	0.0020		Curricular units 2nd sem (enrolled)	0.0481
Application order	0.0020		Curricular units 1st sem (approved)	0.1526
International	0.0024		Curricular units 2nd sem (approved)	0.2009

Finally, the reduced model used the top 9 most important features: ‘Curricular units 1st sem (enrolled)’, ‘Curricular units 2nd sem (evaluations)’, ‘Unemployment rate’, ‘Curricular units 1st sem (credited)’, ‘Curricular units 2nd sem (grade)’, ’Tuition fees up to date’, ‘Curricular units 2nd sem (enrolled)’, ‘Curricular units 1st sem (approved)’, ‘Curricular units 2nd sem (approved)’. Its accuracy was 0.734.

5. Discussion

There are several points to discuss. First, the accuracy scores are remarkably close among the models. While it appears each model can outperform the others depending on the parameter, there is very little variability within the models as their parameters vary. This suggests that the models are all very close in performance. The data may also be mostly linearly-separable and that other relationships are unlikely to be present since the linear SVC model and logistic regression–which is intended more for linear relationships (Bortnik, 2020a) both perform reasonably well and similarly to each other. However, it could also be that they both perform equally poorly. Another potential factor could be that appropriate combinations of parameters were not found to get desired changes in characteristics. Extreme C values were tested separately however, and also did not suggest much change in score.

The confusion matrix, with the large value for true positive guesses of Graduate and Dropout suggest that the model is particularly adept at predicting those two statuses, at least compared to the 'Enrolled' status.

It is also interesting that the two most impactful features are 'Curricular units 1st sem (approved)' and 'Curricular units 2nd sem (approved)'. While it is unclear what exactly "approved" units are relative to "enrolled" or "credited" units, it seems reasonable that the more credits the student has or has the potential to obtain, the more likely they are to earn enough credits to graduate–assuming a positive relationship with success. Assuming a negative relationship, it is possible that having too many credits could overburden students and lead to failure.

The importance of these two school experience-related factors, rather than demographic factors, suggests that this model would be more useful for assessing how a student's anticipated course load might impact their success. The other factors might then help to predict how students with similar course loads but varying demographic characteristics might experience different academic success. However, emphasis should likely not be placed on predictions based solely on demographic or socioeconomic factors.

6. Conclusion

In this project, data on over 4,400 undergraduate students at the Polytechnic Institute of Portalegre were used to generate three models to predict student success. These models included a support vector machine/classifier model, a Softmax regression model, and a bootstrap aggregation model. Different parameters were used to attempt to optimize each model, but the scores for all the models remained around 75%. The two most important features in the model were actually not demographic or socioeconomic characteristics, but related to the number of units approved in their first two semesters, suggesting that the academic requirements or courseload are better predictors of academic success.

7. References

Bortnik, J. (2020a). *Introduction to machine learning lecture 3.3. Multiple classes and optimization* [Lecture video]. University of California, Los Angeles, Los Angeles, California.

Bortnik, J. (2020b). *Intro to Machine Learning lecture 4.3. Ensemble learning* [Lecture video]. University of California, Los Angeles, Los Angeles, California.

Gupta, M. (2020, April 20). Calculating precision & recall for multi-class classification. *Data Science in your pocket.* https://medium.com/data-science-in-your-pocket/calculating-precision-recall-for-multi-class-classification-9055931ee229

Lozinski, A. (2023, November 21). *Introduction to Machine Learning for the Physical Sciences: Week 8 CNNs & GANs* [Lecture video]. University of California, Los Angeles, Los Angeles, California.

Martins, M. V., Tolledo, D., Machado, J., Baptista, L. M.T., & Realinho, V. (2021). Early prediction of student's performance in higher education: A case study. In Á. Rocha, H. Adeli, G. Dzemyda, F. Moreira, & A. M. Ramalho Correia (Eds.), *Trends and Applications in Information Systems and Technologies: Volume 1* (Vol. 1, pp. 166-175). Springer International Publishing. https://doi.org/10.1007/978-3-030-72657-7_16

Realinho, V., Martins, M. V., Machado, J., & Baptista, L. (2021). *Predict students' dropout and academic success*. UCI Machine Learning Repository. Retrieved December 8, 2023, from https://doi.org/10.24432/C5MC89

Scikit-learn. (n.d.a). *4.2. Permutation feature importance — scikit-learn 1.3.2 documentation*. Scikit-learn. Retrieved December 8, 2023, from https://scikit-learn.org/stable/modules/permutation_importance.html#id2

Scikit-learn. (n.d.b). *1.4. Support Vector Machines — scikit-learn 1.3.2 documentation*. Scikit-learn. Retrieved December 8, 2023, from https://scikit-learn.org/stable/modules/svm.html#classification

Scikit-learn. (n.d.c). *1.13. Feature selection — scikit-learn 1.3.2 documentation*. Scikit-learn. Retrieved December 8, 2023, from https://scikit-learn.org/stable/modules/feature_selection.html

Scikit-learn. (n.d.d). *scikit-learn algorithm cheat-sheet* [Flowchart]. Scikit-learn. Retrieved December 8, 2023, from https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Scikit-learn. (n.d.e). *sklearn.ensemble.BaggingClassifier — scikit-learn 1.3.2 documentation*. Scikit-learn. Retrieved December 8, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html#sklearn.ensemble.BaggingClassifier

Scikit-learn. (n.d.f). *sklearn.inspection.permutation_importance — scikit-learn 1.3.2 documentation*. Scikit-learn. Retrieved December 8, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.inspection.permutation_importance.html

Scikit-learn. (n.d.g). *sklearn.linear_model.LogisticRegression — scikit-learn 1.3.2 documentation*. Scikit-learn. Retrieved December 8, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Scikit-learn. (n.d.h). *sklearn.pipeline.Pipeline*. Scikit-learn. Retrieved December 8, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html