A Rising Tide Raises All Ships: An Analysis of Portuguese Secondary Students

Benjamin Castle
CISC5800
Professor Chen

I. Introduction and Summary

EDUCATIONAL data mining has emerged as a critical field for developing early intervention systems in academic environments. For this study, I utilized a dataset donated to the UC Irvine Machine Learning Repository in 2014 by Paulo Cortez, originally used in their 2008 paper. The dataset contains 649 samples with 30 features describing Portuguese students in their final year of secondary education, along with 3 target variables representing their grades at the end of each of the three terms (with the third term being the final grade).

Due to the nature of the population studied, many features are categorical in nature. These range from *traveltime*, which provides four categories describing commute duration to school, to *health*, which offers five categories describing students' self-assessment of their health status. The dimensionality of these features, combined with numerous non-binary variables, necessitates careful feature importance analysis as a primary research objective.

While the original study employed several classical machine learning models, both classifiers and regressors, it inadequately addressed the significant class imbalances present in various features. For instance, the binary *address* variable (determining urban or rural residency) exhibits a more than 2:1 imbalance, while *Pstatus* (indicating whether the student's parents live together) shows a more than 7:1 imbalance. Our research hypothesizes that addressing these imbalances would enhance feature importance assessment and model performance.

This paper builds upon Cortez's foundational work by implementing more rigorous class imbalance correction techniques to generate stronger precision and recall metrics. I maintain the original study's approach to the prediction task structure, focusing primarily on the final grade (G3), but reduce the number of modeling paradigms to just two: regression (exact score prediction) and binary classification (pass/fail). Multiclass classification is potentially useful in other situations, but when simply trying to identify whether a student requires the help to not fail, a binary model is sufficient.

Our research extends the algorithmic exploration beyond the original paper's decision trees, random forests, neural networks, and support vector machines. I incorporate logistic regression, which typically demonstrates robustness in educational feature spaces, and K-nearest neighbors, based on the intuition that secondary school students' tendency toward social clustering might generate algorithmically detectable neighborhoods in the feature space. Additionally, we explore ensemble methods to potentially improve upon the approximately 85% accuracy achieved by single algorithms in our preliminary experiments.

A central contribution of this paper is a detailed analysis of precision-recall tradeoffs, particularly focusing on optimizing the identification of at-risk students (the negative class). By adjusting classification thresholds and implementing targeted class balancing techniques, I aim to develop a framework that educational institutions can use to identify students requiring intervention before academic failure occurs. This is prioritized even at the cost of mis-classifying potentially passing students. With proper model construction, however, these should be those students who were only incrementally passing without any intervention, and so said intervention would be nevertheless welcome.

II. Technology Framework

All of the models were built in Python, with most coming from `sklearn`. All results were cross-validated with 5-fold cross validation, and hyperparameters were selected via a grid search. Experiments can easily be run from the command line by calling `TestRunner.py` with the relevant arguments, which can be found in the Github documentation or via the `--help` flag. All random events are controlled by the `--random_state` flag to ensure consistent reproducibility. Various important graphs and contextual information are output to `STDOUT` and to files within the repository to discover insights about the models.

III. Data Preparation

Due to the richness of the dataset beyond grades and ages, it is necessarily rife with challenges that need to be overcome before machine learning tasks can be performed.

The most important of these is the separation of the trimester grades. The final three columns of the dataset are `G1`, `G2`, and `G3`, which indicate the grade (from 0-20) that the student received at the end of the first, second, and third and final trimester. As may be expected, they exhibit a very strong correlation and reduce the predictive insight that can be gained from other features, and so they need to be separated. In all further tasks, we are exclusively using `G3` as target, omitting `G1` and `G2` entirely.

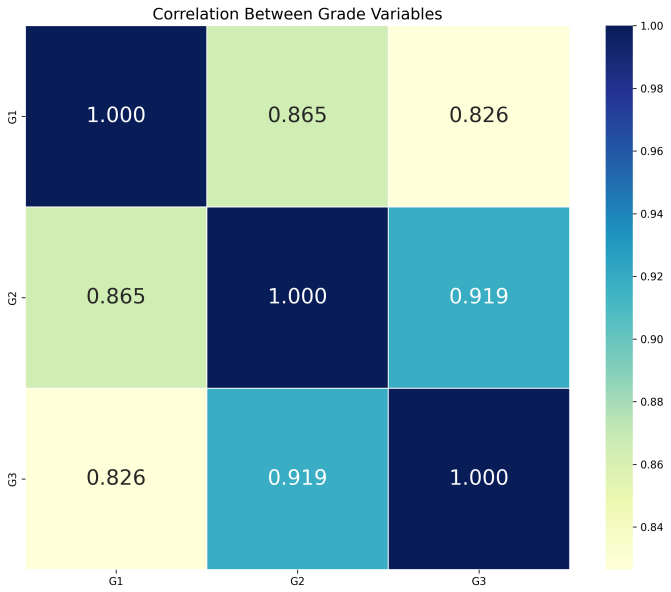

Fig. 1. Correlation between the trimester grade variables. The temporal (G1 and G3 exhibit the worst correlation) and causal (all correlations are above 0.8) relationships are displayed.

TABLE I
TABLE ENCODING

Type	Example	Encoding
Numerical	`age`	Scaled and recentered such that $\mu = 0$ and $\rho = 1$
Binary	`sex`	Converted to $\{0, 1\}$
Ordinal	`Medu`	Integer: left alone Nominal: encoded to integers, with order preserved
Categorical	`reason`	One-hot encoded

Notably, the original work allowed `G1` and `G2` to remain in their feature set, which significantly boosted their accuracy in a number of cases. As such, the accuracy of the models in this paper will sometimes struggle, comparatively. This, however, is the main contribution I present: if the administrators of this school wanted to identify students who were likely to not pass at the end of the year as the year started, they would not have access to the first and second trimester grades. Correctly predicting which students were going to fail, then, would fall upon other factors - such as those biographical features that were collected. This earlier intervention would surely have a tremendous impact on the likelihood of the students succeeding when they otherwise would not have.

After processing these targets, 30 features remain. These features were encoded based on their type, as seen in Table I.

IV. REGRESSION

For regression of student performance on target `G3`, I evaluated six models: Random Forest, Linear Regression, Ridge Regression, LASSO Regression, Support Vector Machine for Regression (SVR), and Gradient Boosting.

TABLE II
REGRESSION MODEL PERFORMANCE COMPARISON

Model	R	RMSE	MAE
Random Forest	0.268	2.777	2.017
Linear Regression	0.220	2.868	2.164
Ridge	0.269	2.775	2.024
Lasso	0.258	2.796	2.057
SVR	**0.291**	**2.733**	**2.008**
Gradient Boosting	0.269	2.776	2.021

A. Model Selection Rationale

Random Forest models tend to work exceptionally well on categorical data and are robust against overfitting. Linear Regression is simple to explain and understand, which would be useful for convincing school administrators of the model's efficacy. Ridge and LASSO, Linear Regression algorithms which modify the model's coefficients to introduce feature selection or dampening, fill a similar niche, but can be somewhat stronger than un-penalized Linear Regression. Support Vector Machines and Gradient Boosting have the potential to provide excellent depth of insight on complex datasets.

B. Performance

All of the models tested performed quite poorly. Without the assistance of `G1` and `G2`, the regression models were unable to achieve an r^2 greater than 0.3 and generated much higher root mean squared error (RMSE) and mean absolute error (MAE). As seen in Table IV, SVR performed the best, with the highest r^2, lowest RMSE, and lowest MAE. Even still, this model was unable to perform very well, although there was a small improvement over Cortez's result.

V. BINARY CLASSIFICATION

A. Class Imbalance

The Cortez notes that a passing grade for this schooling system is a 10, so for binary classification, the data was split into the negative, < 10 class and the positive, ≥ 10 classes. This, unfortunately, creates a significant class imbalance - despite the uniquely high dropout and failure rate in the larger population, 85% of students in the dataset still passed and graduated, as seen in Fig. 2. Therefore, for binary classification, imbalance correction techniques needed to be applied.

For this application, three methods were utilized:

1) **Synthetic Minority Over-sampling Technique** (SMOTE) - generating synthetic members of the minority class until there are balanced amounts of each class.
2) **Class weighting** - weighting the penalty of misclassifying higher for the negative minority class compared to the positive majority class. Many of `sklearn`'s models have a `class_weighting` parameter built-in, which weights each class by the inverse of its frequency proportion relative to the training set.

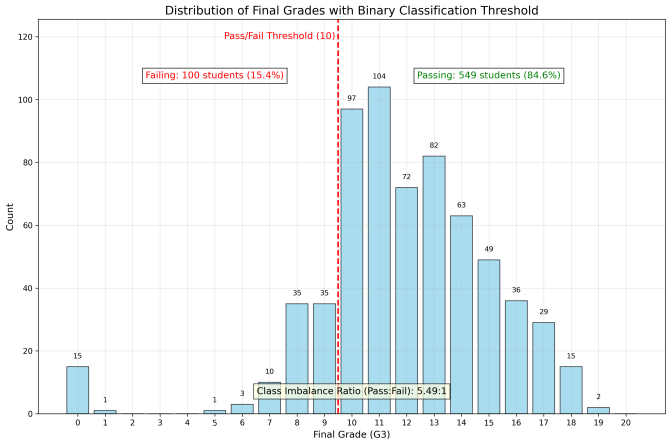

Fig. 2. Distribution of G3. The majority of students pass, albeit usually by small margins.

3) **Under-sampling** - removing random samples from the majority class until the imbalance is removed. Performs poorly and will not be discussed. The dataset is already quite limited in samples, even before many are removed.

B. Single Models

For binary classification of student performance, I evaluated five primary models: Random Forest, Logistic Regression, K-Nearest Neighbors (KNN), SVM, and Gradient Boosting. Each algorithm was selected based on specific characteristics that align with educational data analysis requirements.

1) Model Selection Rationale: Random Forest was chosen for its robustness to overfitting and ability to capture non-linear relationships between educational features. Logistic Regression offers interpretability, crucial when explaining prediction factors to educational stakeholders. KNN was included based on the hypothesis that educational outcomes often cluster among similar student profiles. SVM and Gradient Boosting provide sequential learning advantages for capturing complex interactions between educational features.

2) Methodology: All models in this section were trained using SMOTE, as discussed previously. I employed a comprehensive grid search with 5-fold cross-validation to identify optimal hyperparameters, with two probability decision thresholds: one left at the default 0.5, and one automatically tuned to prioritize recall for failing students (the minority class). This second focus aligns with the goal of early intervention system development, where identifying at-risk students is paramount.

3) Performance Analysis of Table III:

a) SVM: The SVM algorithm achieved a decent 64% balanced accuracy, but was unable to accurately capture many of the negative class instances. This will become a pattern for these models with the default threshold.

b) K-Nearest Neighbors: The KNN model achieved 61% balanced accuracy with optimal $k = 11$ neighbors. Its performance in identifying failing students was middling. The algorithm's strength in capturing localized patterns in the feature space likely made it ineffective at properly classifying those students who were near the class split at G3$= 10$, since there would be, on average, more neighbors to the right of the split, pulling the negative class samples across the midline.

c) Gradient Boosting: The Gradient Boosting implementation achieved a balanced accuracy of 62%, but it only did so at the cost of recall for failing students at 36%. Its sequential learning process highlighted complex feature interactions, with `failures` emerging as by far the most important in its prediction process. Gradient boosting seems to struggle in this feature environment.

d) Random Forest: The Random Forest classifier achieved balanced accuracy of 70% with 55% recall for failing students. The most influential features identified were `failures` and `school`, hinting at a potential collinear relationship between the number of classes a student has failed and the likelihood of them failing in the future. The model demonstrated strong robustness against noisy educational data while maintaining reasonable interpretability through feature importance analysis.

e) Logistic Regression: Logistic Regression achieved only 72% balanced accuracy but had 85% recall for at-risk students. Despite its simpler structure, it performed competitively with more complex models, particularly in terms of negative class precision and recall. This model is almost certainly the most effective to use in this simple manner, without threshold tuning, thanks to not having to drop precision beyond the other models. This indicates that the model, with its strong balanced accuracy, is very capable of detecting the overwhelming majority of failing students. This does mean that many passing students are being flagged as at-risk by this algorithm, but considering the very centralized distribution of final grades, there are many students who could still use support despite passing.

4) Feature Importance Analysis: A consistent pattern emerged across all models regarding feature importance. Fig 3 displays the top 15 features identified by the Random Forest model. Notably, `failures` and `school` consistently ranked highly across all models, suggesting their fundamental importance in predicting student performance. These insights align with Cortez and offer actionable guidance for intervention strategies. School choice is an interesting division, indicating that perhaps one of the schools is of a higher quality than the other or that one of the schools serves a population with fewer resources. Of note is the fact that the one-hot encoding column `reason_reputation` is also in this top 15 - one school appears to have a reputation for excellence, leading parents with the resources to pick to oftentimes pick that one, leading to the self-perpetuating cycle which ends with that school having more resources and

TABLE III
COMPARISON OF MODEL PERFORMANCE WITH DEFAULT AND OPTIMIZED DECISION THRESHOLDS

Algorithm	Default Threshold (0.5)				Threshold	Optimized for Minority Recall			
	Bal. Acc.	Min. Recall	Min. Prec.	ROC AUC		Bal. Acc.	Min. Recall	Min. Prec.	ROC AUC
svm	0.641	0.606	0.253	0.716	0.77	0.678	0.879	0.234	0.716
knn	0.613	0.606	0.225	0.671	0.91	0.571	0.939	0.176	0.671
gradient_boosting	0.621	0.364	0.353	0.761	0.98	0.701	0.606	0.351	0.761
random_forest	0.699	0.545	0.400	0.773	0.70	0.711	0.879	0.259	0.773
logistic_regression	0.718	0.848	0.272	0.758	0.59	0.717	0.879	0.264	0.758

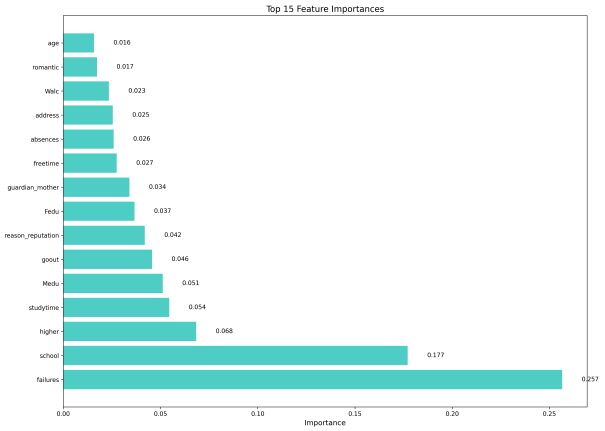

Fig. 3. Top 15 features for predicting student performance, as identified by the Random Forest model. Bar length corresponds to relative importance. This visualization highlights which student characteristics most strongly influence academic outcomes.

better outcomes for its students.

C. Single Models Threshold Tuning

Similarly to before, I evaluated all the same models, but tuned their threshold to achieve at either 85% minority class recall, or the next best that could be achieved.

1) Methodology: All models in this section were trained using SMOTE and grid search, then their prediction probabilities were investigated to find the prediction threshold closest to 0.5 at which they would report an 85% minority class recall.

2) Performance Analysis of Table III:

a) SVM: The SVM algorithm's balanced accuracy remained similar at 68% and was able to increase its negative class recall to 88%. The negative class precision remained almost identical, indicating that the model's predictive power was only increased by the threshold modification.

b) K-Nearest Neighbors: The KNN model was only harmed by trying to achieve higher negative class recall. Balanced accuracy suffered and precision was reduced to meaningless levels. Recall increased markedly, as the extremely high threshold predicted almost every sample to be in the negative class.

TABLE IV
ENSEMBLE MODEL PERFORMANCE COMPARISON

Ensemble	Bal. Acc.	Min. Recall	Min. Prec.
No. 1	0.268	2.777	2.017
No. 2	0.220	2.868	2.164
No. 3	0.269	2.775	2.024

c) Gradient Boosting: When tuning the prediction threshold, Gradient Boosting saw a marked increase in its predictive power. Interestingly, despite its threshold increasing to almost 1, the balanced accuracy and recall jumped, while the precision remained almost identical. This suggests that threshold tuning is tremendously effective in this case.

d) Random Forest: The Random Forest classifier achieved balanced accuracy of 71% for failing students, and saw a marked increase in negative class recall. The precision dropped significantly, however, indicating that at this 0.7 threshold, the model is beginning to predict far too many members of the positive class as potential failing students.

e) Logistic Regression: Logistic Regression achieved only 72% balanced accuracy but had 85% recall for at-risk students. Despite its simpler structure, it performed competitively with more complex models, particularly in terms of negative class precision and recall. This model is almost certainly the most effective to use in this simple manner, without threshold tuning, thanks to not having to drop precision beyond the other models. This indicates that the model, with its strong balanced accuracy, is very capable of detecting the overwhelming majority of failing students. This does mean that many passing students are being flagged as at-risk by this algorithm, but considering the very centralized distribution of final grades, there are many students who could still use support despite passing.

3) Comparative Insights: Figure 4 presents a comparative analysis of the negative class recall of all five models vs the probability threshold. Naturally, as the threshold is increased, the negative recall increases - more samples fall under the probability threshold and are classified as failing. As seen in the analysis, even at a tremendously high threshold, Gradient Boosting experiences very little change in its negative class recall. Most other models experience a steeper linear relationship, with logistic regression spiking aggressively in the middle but falling just short of the 0.85 cutoff.

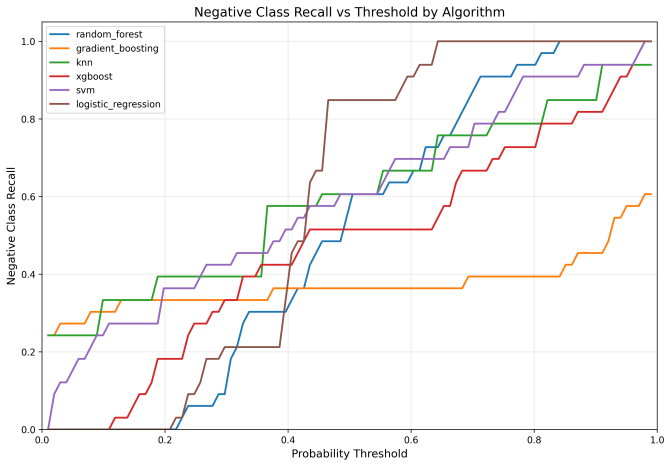

Fig. 4. Chart displaying the effect increasing the decision threshold has on negative class recall.

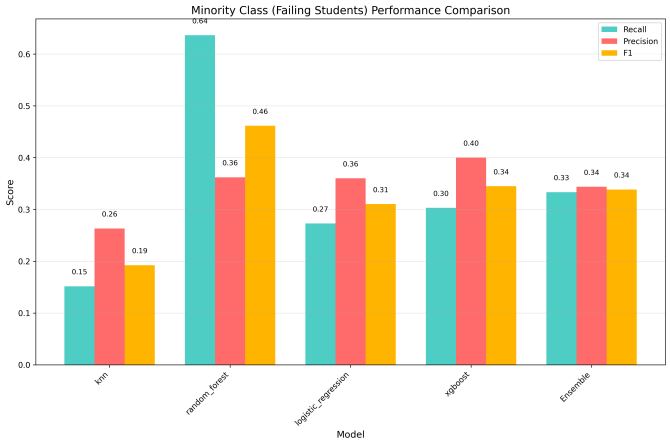

Fig. 5. Ensemble 2 Individual Model Contributions

D. Ensemble

Based on the success of the single classification models, I wanted to try creating an ensemble of those methods. Ensemble methods tend to have more ability to detect minority classes, so in this section, I ran experiments involving three different ensembles of the previous models in an attempt to generate higher negative class recall while retaining more balanced accuracy.

a) Methodology: Much as in the single model implementation, each model of the ensemble was trained with grid search to derive the optimal hyperparameters for the negative class recall. Then, each model's classification of each sample was considered and the sample was classified by majority vote. This new "model" was then tested for all of the relevant metrics.

b) Ensemble 1: The first is an ensemble of Random Forest, Logistic Regression, and XGBoost. This setup was quite weak, generating a decent balanced accuracy of 68% but a poor negative class recall of only 48%. I suspect that this model was not as effective as I had expected because the poor

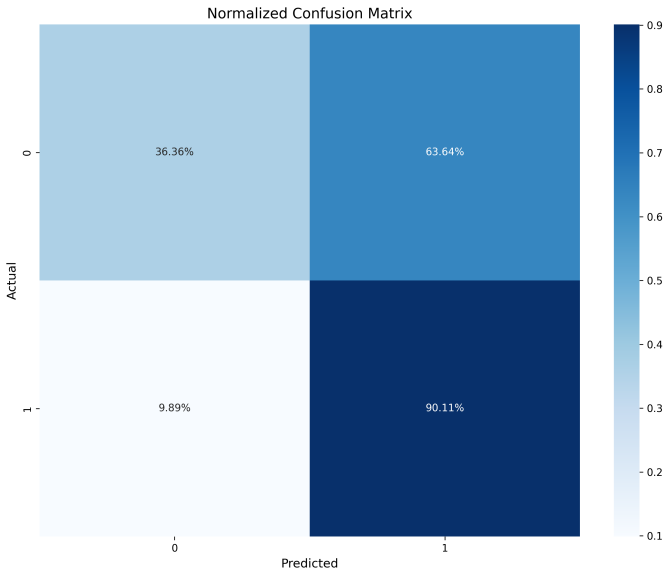

Fig. 6. Confusion matrix for Ensemble 3

predictive power of XGBoost trumped the ability of Random Forest and Logistic Regression to detect the negative minority classes when they were unable to collectively outvote it.

c) Ensemble 2: The second is an ensemble of KNN, Random Forest, Logistic Regression, and XGBoost. The intention behind this ensemble was to include the KNN model's ability to gather all of the samples which had very low ($\leq$ 5) G3 scores. Unfortunately, it seems that the weakness of the KNN that was identified in the single model case won out - this ensemble generated a poor 61% balanced accuracy and only 33% negative class recall. Figure 5 displays this behavior, with KNN having an abysmal 15% recall with only a 26% precision for the negative class.

d) Ensemble 3: Finally, I created an ensemble of Random Forest, SVM, and Gradient Boosting. This resulted from analyzing the prior individual models' performance. Random Forest has an exceptionally high negative class recall, and Gradient Boosting has the best precision of all the models. Based on the curve in Figure 4, SVM has the potential to be very effective in both categories. Unfortunately, despite SVM attaining a recall of 70%, its 27% precision for failing students meant that the ensemble as a whole was ineffectual.

e) Ensemble Analysis: It would seem that a majority voting ensemble of these models is not a tremendously effective method of identifying negative class samples. Interestingly, these ensemble models were all very powerful at detecting the positive class. Figure 6 displays the confusion matrix for Ensemble 3, which turns out to be a very optimistic model. This is surprising considering the grid search process is attempting to train the models to predict for negative recall.

REFERENCES

[1] P. Cortez, *Student Performance*, UCI Machine Learning Repository, 2008. [Online]. Available: https://doi.org/10.24432/C5TG7T.