Student intervention

1. Classification vs Regression

Your goal is to identify students who might need early intervention - which type of supervised machine learning problem is this, classification or regression? Why?

Our target output being discrete (even boolean : pass or fail exam). This is a classification problem.

2. Exploring the Data

Can you find out the following facts about the dataset?

- Total number of students
- Number of students who passed
- Number of students who failed
- Graduation rate of the class (%)
- Number of features (excluding the label/target column)

Use the code block provided in the template to compute these values.

Total number of students: 395
Number of students who passed: 265
Number of students who failed: 130
Number of features: 30
Graduation rate of the class: 67.09%

3. Preparing the Data

Execute the following steps to prepare the data for modeling, training and testing:

- Identify feature and target columns
- Preprocess feature columns
- Split data into training and test sets

Starter code snippets for these steps have been provided in the template.

4. Training and Evaluating Models

Choose 3 supervised learning models that are available in scikit-learn, and appropriate for this problem.

For each model:

- What are the general applications of this model? What are its strengths and weaknesses?
- Given what you know about the data so far, why did you choose this model to apply?
- Fit this model to the training data, try to predict labels (for both training and test sets), and measure the F1 score. Repeat this process with different training set sizes (100, 200, 300), keeping test set constant.
- Produce a table showing training time, prediction time, F1 score on training set and F1 score on test set, for each training set size.

Note: You need to produce 3 such tables - one for each model.

Naive Bayes

We only have 300 data points to train our algorithm. Naive Bayes is known to perform well when we don't have much data to train our model. That's why it makes a good candidate. They also have the advantage of being fast when compared to more complicated models.

The disadvantage is that it assumes that features are independent from one another which does not make it a good candidate for cases when we most probably have highly correlated input variables.

training set size	training time	prediction time	F1 on training set	F1 on testing
100	0.002	0.001	0.57	0.46
200	0.002	0.001	0.36	0.41
300	0.001	0.000	0.79	0.81

Support Vector Machines

Support Vector Machines has the advantage of being effective when we have a lot of features (here 48 features after preprocessing, which is a high number given we only have 300 training points).
Disadvantage is that it usually requires scaling to properly compare features which is an extra step, that could lead to poor performance in our context.

training set size	training time	prediction time	F1 on training set	F1 on testing
100	0.006	0.001	0.92	0.76
200	0.004	0.002	0.86	0.81
300	0.009	0.002	0.86	0.81

Decision Tree

Decision trees require little data preparation (for instance, it copes with our unscaled vectors) and it is easy to understand. Our data is not too unbalanced (67% of one class, 33% of another) to be a serious problem to this technique.

On the other side, It is prone to overfitting if we don't limit the size of the tree (minimum sample per node, per leaf, max depth of the tree). A tree leaves setting apart small number of instance is specialized in the training data and hence won't generalize well.

training set size	training time	prediction time	F1 on training set	F1 on testing
100	0.003	0.000	1	0.74
200	0.002	0.000	1	0.74
300	0.002	0.000	1	0.74

5. Choosing the Best Model

Based on the experiments you performed earlier, in 2-3 paragraphs explain to the board of supervisors what single model you choose as the best model. Which model has the best test F1 score and time efficiency? Which model is generally the most appropriate based on the available data, limited resources, cost, and performance? Please directly compare and contrast the numerical values recorded to make your case.

We did not tune the parameters of our model yet. When looking at the tables above, the results of the Decision tree are very interesting : the model is time efficient (less than 0.001 second to make a prediction,

the best result), in that sense it would be a good candidate. The second interesting thing is that we notice a important gap between training score (F1 of 1 on the training set for 100, 200 and 300 instances) and testing score (F1 of 0.74 on the testing set). It clearly overfits but we might be able to solve that issue when tuning the parameters of the model. The margin of improvement is greater than it is for other models.

If we manage to improve the performance of our tree, it can become the most performant and the quickest of all 3 models. As there is a good chance to do so, I am choosing this model.

If it turns out the F1 does improve much with parameter tuning, we will revert to SVM which has a better performance (F1 of 0.81 on the training set) and is still quick at making predictions (0.002s in the 300 data points training set case).

In 1-3 paragraphs explain to the board of supervisors in layman's terms how the final model chosen is supposed to work (for example if you chose a decision tree or support vector machine, how does it learn to make a prediction).

a Decision tree figures out what features separate better the different values of our output variable. It does that according to a measure of purity (Entropy or Gini coefficient for instance). The tree start with the feature that best set apart output values and repeat this process until there are no more points to classify.

For instance if all students with internet would pass the exam and all students without internet would fail, our tree would be able to perfectly classify our data points according to this criteria. It would ask 1 question ("Do you have an internet connection?") and would predict accordingly the result to the exam.

Fine-tune the model. Use gridsearch with at least one important parameter tuned and with at least 3 settings. Use the entire training set for this.

What is the model's final F1 score?

We manage to raise the F1 score of our decision tree at 0.78. It is the faster model, but SVM would probably perform better after tuning.