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?

Answer：

This is a classification problem. The classification process is taking some input and mapping them to some discreet labels, usually true or false. The goal to identify the students who might need early intervention is mapping students to need or not need to early intervention classes.

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.

Answer:

- 395
- 265
- 130
- 67.09%
- 31

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.

1. SVC
   Advantage:

   - Effective in high dimensional spaces.
   - Still effective in cases where number of dimensions is greater than the number of samples.
   - Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
   - Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

   Disadvantage:

   - If the number of features is much greater than the number of samples, the method is likely to give poor performances.
   - SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation (see Scores and probabilities, below).

	Training set size		
	100	200	300
Training time (secs)	0.001	0.003	0.008
Prediction time (secs)	0.002	0.002	0.002
F1 score for training set	0.875	0.8774	0.8809
F1 score for test set	0.7910	0.8012	0.75

2. Decision Tree

Advantage:

- Requires little data preparation.
- The cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree.
- Able to handle both numerical and categorical data.
- Uses a white box model. If a given situation is observable in a model, the explanation for the condition is easily explained by Boolean logic. By contrast, in a black box model (e.g., in an artificial neural network), results may be more difficult to interpret.
- Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model.
- Performs well even if its assumptions are somewhat violated by the true model from which the data were generated.

Disadvantage:

- Decision-tree learners can create over-complex trees that do not generalize the data well. This is called overfitting.
- Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This problem is mitigated by using decision trees within an ensemble.
- There are concepts that are hard to learn because decision trees do not express them easily, such as XOR, parity or multiplexer problems.
- Decision tree learners create biased trees if some classes dominate. It is therefore recommended to balance the dataset prior to fitting with the decision tree.

	Training set size		
	100	200	300
Training time (secs)	0.001	0.002	0.003
Prediction time (secs)	0.000	0.000	0.001
F1 score for training set	1.0	1.0	1.0
F1 score for test set	0.7309	0.6094	0.7259

3. Stochastic Gradient Descent
   Advantage:

   - Efficiency.
   - Ease of implementation (lots of opportunities for code tuning).

   Disadvantage:

   - SGD requires a number of hyperparameters such as the regularization parameter and the number of iterations.
   - SGD is sensitive to feature scaling.

	Training set size		
	100	200	300
Training time (secs)	0.000	0.001	0.002
Prediction time (secs)	0.001	0.001	0.000
F1 score for training set	0.7848	0.8354	0.8302
F1 score for test set	0.8043	0.7724	0.8111

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.

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).

Fine-tune the model. Use grid search 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?

I choose the SGD model as the best model. It has the best F1 score and time the lowest time consumption.

It is the linear classifiers (SVM, logistic regression, i.e.) with SGD training. This estimator implements regularized linear models with stochastic gradient descent (SGD) learning: the gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule (aka learning rate).

I choose parameters ‘loss’ and ‘penalty’ to fine-tune the model. The model’s final F1 score is as follow table. The loss is logistic regression and penalty is none.

	Training set size		
	100	200	300
Training time (secs)	0.001	0.002	0.001
Prediction time (secs)	0.000	0.000	0.001
F1 score for training set	0.8267	0.8120	0.8245
F1 score for test set	0.8105	0.7950	0.8182