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?

This seems more likely to be a classification problem. Output of this problem is binary and discrete: either the students need early intervention or not.

2. Exploring the Data

Can you find out the following facts about the dataset?

- Total number of students: 395
- Number of students who passed: 265
- Number of students who failed: 130
- Graduation rate of the class (%): 67.09%
- Number of features (excluding the label/target column): 30

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

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.

a. ID3 algorithm for Decision Tree Classifier

In theory, ID3 algorithm runs in O(nlogn), taking O(n) space. In classification problem, decision tree model builds a tree, in which each node is a simple decision rule.

I chose the model because this is a simple classification problem, and Decision Tree classifier allows me to visualize the classifier easily.

Strengths of Decision Tree:

- Simple to interpret and visualize
- Query time and space complexity is only O(logn)
- Perform well on both numerical and categorical data

Weaknesses of Decision Tree:

- Tendency to overfit the data (need to set max_depth, or prune the tree afterwards)
- Can become unstable: small variation in data can generate an entirely different tree. In this example, higher training data (300), generated a different tree, ultimately leading to a lower F1 score.

Result

Training set Size	100	200	300

Training time (secs)	0.002	0.002	0.002
Prediction time (secs)	0.000	0.000	0.000
F1 score for training set	0.91	0.85	0.83
F1 score for test set	0.81	0.81	0.76

b. Support Vector Machine

In classification problem where data is separable, SVM constructs a hyperplane that best separate data (hyperplane that has higher margin).

I chose the model because it is effective in still effective in problems that have high number of features and small numbers of data (like this problem).

Strengths of SVM:

- Still effective in high dimensional space, to an extend when relative to number of training samples
- Memory efficient: O(n) space

Weaknesses of SVM:

- This model seems to require more tuning of parameters: kernel choice, C, gamma, etc.
- Expensive training time

Result

Training set Size	**100**	**200**	**300**
Training time (secs)	0.006	0.005	0.011
Prediction time (secs)	0.002	0.003	0.008

F1 score for training set	0.90	0.89	0.88
F1 score for test set	0.77	0.80	0.78

c. Instance based learning (KNN)

This model makes prediction by looking at nearest neighbors.

I chose the model because it can be highly effective in this case: conventional wisdom states that students with similar family background are likely to share similar educational prospects.

Strengths of KNN:

- Require minimal training time
- Successful in problems where decision boundary is irregular

Weaknesses of KNN:

- Require higher query time and query space

Result

Training set Size	**100**	**200**	**300**
Training time (secs)	0.002	0.003	0.003
Prediction time (secs)	0.005	0.008	0.006
F1 score for training set	1.00	1.00	1.00
F1 score for test set	0.77	0.79	0.79

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 recored 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 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?

To whom it may concern,

To identify students who require early intervention, I propose that we use Instance based learning model, specifically k nearest neighbor.

As a quick overview, given a new student (called the subject), this model predicts whether the subject needs early intervention by looking at other similar students. At each run, the model looks at existing students in the database, computes and selects 20 students who share the most similar background/features with the subject. If more than half of these 20 other students have failed to graduate, the subject likely needs early intervention; otherwise, (s)he likely does not need early intervention. This model does not need to be trained; all computations happen when making prediction.

This model is advantageous for two reasons. Firstly, it makes sense by conventional wisdom. For example, students from the same neighborhood are likely to have similar educational prospect. Secondly, it saves resource by requiring minimal training time. When new batch of students are added to the database, the model doesn't have to be retrained.

There is concern that querying time and space is less efficient than other models. However, in practice, this doesn't seem to be a legitimate concern for our use case. Querying space grows only

linearly as new students are added to the database. Querying time also seems to be very minimal (a couple dozens of milliseconds)

In conclusion, I believe that Instance based learning is an effective model in our case. It is both accurate and efficient.

The final F1-score is 0.8