Model Selection for the Student Intervention System

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 difference between classification and regression is that the output of classification problem is a discrete number representing which class the data belongs to; the output of regression problem is a continuous number. In the student intervention problem, the output is a discrete number: whether or not the student need intervention. And therefore it is a classification problem.

2. Exploring the Data

Can you find out the following facts about the dataset?

Answer:

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

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.

Answer:

Algorithm 1: Logistic Regression

- **General Application:**
  Logistic regression can generate a linear decision boundary. It is widely used in many application especially for situation which there are many features but not too much data, because the linear decision boundary is less prone to overfitting.
- **Strength:**
  Fast in training. Extremely fast in predicting.
- **Weakness:**
  Can only generate linear decision boundary and thus less effective when the data is not linearly separable.
- **Performance table:**

Logistic Regression	Training set size		
	100	200	300
Training time (secs)	0.001	0.002	0.002
Prediction time (secs)	0.000	0.000	0.000
F1 score for training set	0.859	0.856	0.847
F1 score for test set	0.764	0.791	0.806

  - Dependent on different optimization method the theoretical running time in training is between O(n) to $O(nd^2)$ n-number of data, d-dimension of features.
  - From the experiment, the training time seems to increases linearly with data.
  - Prediction time is extremely fast.
  - F1 score for test set is increases with the number of training data.

Algorithm 2: SVM

- General application: SVM use the kernel trick to map the linearly non-separable data into higher dimensions. It can produce very complex decision boundary. It is best suitable for situations which the data is not linearly separable.
- Strengths: can create highly complex decision boundary and require less data to train than neural network.
- Weakness: higher computational and memory cost than logistic regression.

Performance table:

SVM	Training set size		
	100	200	300
Training time (secs)	0.001	0.003	0.008
Prediction time (secs)	0.000	0.001	0.002
F1 score for training set	0.878	0.868	0.876
F1 score for test set	0.775	0.781	0.784

  - Theoretically the data SVM requires quadratic programming, the running time is $O(n^3)$ for training. In prediction the running time is linear with number of the support vectors.

- From the experiment the training and prediction time both increase with data
- The F1 score for test set is almost constant with the increasing data

Algorithm 3:Adaboost

- General application: Adaboost is one of the ensemble methods which combine the predictions of several "weak" learners to build a more accurate and robust classifier.
- Strengths: can be used in conjunction with many other types of learning algorithm to improve their performance. It also higher fewer hyper-parameters to tune.
- Weakness: when the number of outliers is large, adaboost is susceptible to noise.

Performance table:

Adaboost	Training set size		
	100	200	300
Training time (secs)	0.066	0.063	0.056
Prediction time (secs)	0.007	0.007	0.004
F1 score for training set	0.948	0.896	0.864
F1 score for test set	0.767	0.776	0.782

- Theoretically running of depend on choose of week learner, for decision tree the running time is about O(n).
- From experiment Training time increase slightly with data
- Prediction time almost stay constant
- The F1 score for test set is almost constant with the increasing data

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?

- **Performance summary:**

  F1 score: All 3 algorithms have similar F1 score.

  Prediction time: Logistic Regression < SVM < Adaboost

The F1 score of the 3 algorithms are similar. Due to the small data set, the result has some randomness in it and the generally F1 score is about 78%.

The SVM has moderate prediction time which tends to increase as the training set size.

The logistic regression model is most computationally efficient, it takes almost no time in prediction, and the least time in training.

The adaboost predicting time of adaboost is highest for the current data set, however it increase very slowly as the data size grows.

- **Choice of model**

Logistic regression is chosen as the final classification model, because the result from the 3 algorithm is similar, and logistic regression is fastest algorithm of the three in training and is extremely fast in prediction. And therefore, it is the best algorithm for the limited resource.

Description:

Logistic regression uses the predictors such as age, absence, health, age etc to produce a probability of whether student will pass or fail, with 1 indicate certainly pass, 0 indicate certainly fail and the value between 0 and 1 indicate the probability in between.

The method works as follows: the predictors are weighted by certain coefficient and summed together to produce a score. The score is transformed by a logistic function to represent probability, as shown in Fig.1.

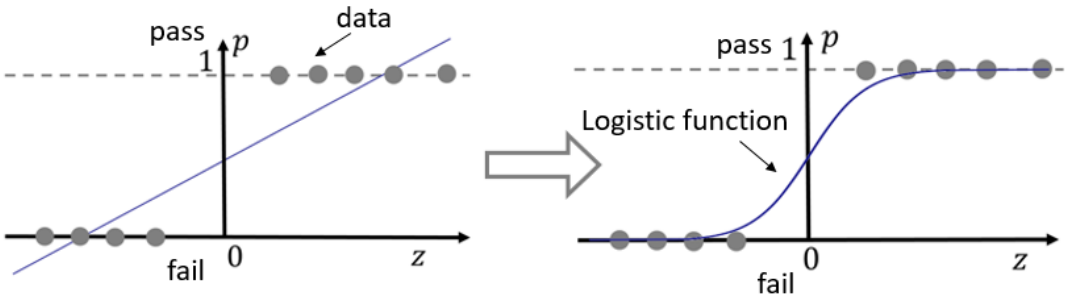

Fig.1 illustration of the logistic function

Training process:

The training process is to learn from the data to obtain the coefficients corresponding to each predictor. The coefficient are adjusted iteratively with the training data, so that they produce the low error in predicting whether a student will pass.

Prediction process:

The prediction process is to predict whether a student will pass from the value of the predictors. It compute the weighted sum of each predictor time its coefficients. Then the value is transferred to the probability of whether a student will pass or fail.

Fine-tune the model

The major tuning parameters in logistic regression is regulation parameter c. c is the inverse of the regularization strength, the small the value the stronger regularization.

Final model

c = 0.05

Logistic Regression	Training set size
	300
Training time (secs)	0.002
Prediction time (secs)	0.000
F1 score for training set	0.832
F1 score for test set	0.780