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%
- 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.**

1. **SVM**

   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).

   Reason to choose:

   - The SVM model constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space, the separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class (so-called functional margin). The SVM model maximize the margin between the classes, so it must be noise endurance.

	Training set size		
	100	200	300
Training time (secs)	0.002	0.004	0.009
Prediction time (secs)	0.002	0.002	0.002
F1 score for training set	0.8718	0.8590	0.8690
F1 score for test set	0.8116	0.8026	0.7895

2. Decision Tree

Advantage:

- 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.
- 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.

Reason to choose:

- The decision tree is non-parametric supervised learning method, easy to implement and requires little data preparation. Decision tree learning is generally best suited to problems with the following characteristics:

  1. The decision tree approach is appropriate for the target function has discrete output values. In this case, the output value has two class, pass or not pass.

  2. Instances are represented by attribute-value pairs. There is a finite list of attributes and each instance stores a value for that attribute. When each attribute has a small number of distinct values, it is easier for the decision tree to reach a useful solution. The algorithm can be extended to handle real-valued attributes.

  3. The training data may contain errors. Errors in the classification of examples, or in the attribute values describing those examples are handled well by decision trees, making them a robust learning method.

  4. The training data may contain missing attribute values. Decision tree methods can be used even when some training examples have unknown values (e.g., humidity is known for only a fraction of the examples).

	Training set size		
	100	200	300
Training time (secs)	0.002	0.002	0.004
Prediction time (secs)	0.000	0.000	0.000
F1 score for training set	1.0	1.0	1.0
F1 score for test set	0.6954	0.6364	0.6723

3. **Naive Bayes**

Advantage:

- Conceptually very easy to understand.
- Very effective

Disadvantage:

- Initialization is a bit time consuming
- Assumes independence of features.

Reason to choose:

- Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality. Base on the experiments results, the naïve Bayes approach has the lowest time consumption.

	Training set size		
	100	200	300
Training time (secs)	0.002	0.001	0.001
Prediction time (secs)	0.000	0.001	0.001
F1 score for training set	0.8244	0.8111	0.7849
F1 score for test set	0.7793	0.7817	0.8

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?

Based on the experiments, the Naïve Bayes has the lowest time consumption. The SVM model has the best F1 score, for 100 training point, the F1 score for test set is 0.8116. Even if the SVM is the most time consumption model, I still choose the SVM model as the best model because it has the best F1 score.

The SVM models is used to maximize the margin between the different classes. As we know there are many decision boundaries can separate the positive and negative classes but which decision boundary we should choose is a problem. The SVM model is an approach to choose the decision boundary to maximize the margin between different classes. Hence the model could be much more stable and will not be affect by noises easily. There is an important parameter required in the SVM, the kernel function, the kernel function is used to map the sample from higher dimensions to lower, in many cases, the samples are not linearly spreadable, and the kernel function can transfer those samples to linearly separable.

I choose parameters 'gamma' and 'C' to fine-tune the model. The model's final F1 score is as follow table. Best model parameter: {'C': 0.01, 'gamma': 1.0000000000000001e-09}

	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	0.8095	0.8421	0.8255
F1 score for test set	0.8008	0.7947	0.8199