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 is a binary classification machine learning problem because we are trying to predict a discrete value, which in this case is either a “yes” or “no”.

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 = 30

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

See code.

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.*

Decision Tree Classifier

Decision tree classifier is a useful algorithm for working with categorical data. Its strength lies in its simplicity and tolerance to outliers in the data set. It also deals well with missing data and is good at weeding out irrelevant features in the data set. Its biggest weakness is its ability to easily overfit the data.

This algorithm was chosen because of the categorical nature of the data set and also because it is easy to interpret and explain.

	100	200	300
Training time (sec)	0.001	0.001	0.002
Prediction Time (sec)	0.000	0.000	0.000
F1 score for training set	1.0	1.0	1.0
F1 score for test set	0.672	0.732	0.787

Gaussian Naïve Bayes

Naïve Bayes is an extremely simple and fast algorithm that works well in cases where the data set is small because it rarely overfits. It is useful for classifying categorical data. Its biggest drawback is that it assumes a conditional independence between features given a class. This is usually not the case. Dependencies between the features cannot be modeled by naïve bayes.

This algorithm was chosen because of the categorical nature of the data set and also because of its simplicity, speed and low memory footprint.

	100	200	300
Training time (sec)	0.001	0.001	0.002
Prediction Time (sec)	0.000	0.000	0.000
F1 score for training set	0.847	0.841	0.804
F1 score for test set	0.803	0.724	0.763

Random Forest Classifier

Random forest classifier is an ensemble classifier. It consists of multiple decision trees all constructed using randomly selected subsets of the data. The output class is the mode of classes output by each decision tree. It is one of the most accurate learning algorithms available and works best with large datasets. Like decision trees, it is good at weeding out irrelevant feature. The biggest drawback with random forest is its resource requirements as the number of trees constructed gets large. This makes it unsuitable for real-time predictions.

This algorithm was used for the categorical nature of the data set and because it usually produces more accurate results than decision trees.

	100	200	300
Training time (sec)	0.017	0.017	0.017
Prediction Time (sec)	0.001	0.001	0.001
F1 score for training set	0.992	0.993	0.998
F1 score for test set	0.719	0.716	0.731

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).

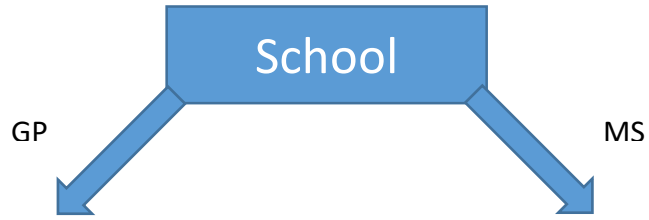
Fine-tune the model. Use gridsearch with at least one important parameter tuned and with at least 3 settings. Use the entire training set for this.

What is the model's final F1 score?

Based on experiments performed, the decision tree model is the best model for this use case. It produced the best F1 score on test data; 0.787. The second best score was from the naïve bayes model which produced a score of 0.763. The random forest model came in last at 0.731 which was possibly due to the small size of the dataset. The decision tree model and the naïve bayes model were tie for best time efficiency with training time at 0.002 sec and prediction time of 0.000 sec.

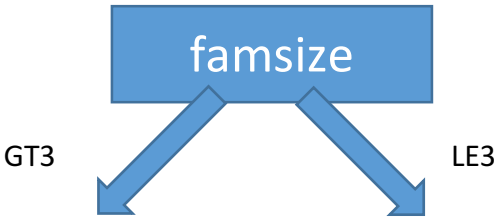
Given the size of the data given and its categorical nature, and given that we have limited resources and cost considerations, the most appropriate model for this use case would be the decision tree model. The random forest classifier though generally more powerful and accurate than the decision tree classifier or the naïve bayes classifier is not suitable in this situation because of its large resource requirements and the small sized data given. The decision tree and naïve bayes classifiers take the same amount of time to train and predict, 0.002 sec and 0.000 sec respectively, but the decision tree classifier wins out because it produces a slightly more accurate model for the data.

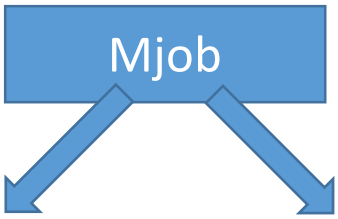
During the learning phase the decision tree classifier uses the data to construct a tree similar to the one shown below. The objective is to split the data into subsets until all examples in a particular subset are either of passing students only or of failing students only. To illustrate, assuming we take the first 3 and last 3 examples in the data, then we can build a decision tree that looks something like this.



school	sex	famsize	pass
GP	F	GT3	no
GP	F	GT3	no
GP	F	LE3	yes

school	age	Mjob	pass
MS	21	other	no
MS	18	services	yes
MS	19	other	no





school	sex	famsize	pass
GP	F	GT3	no
GP	F	GT3	no

school	sex	famsize	pass
GP	F	LE3	yes

school	age	Mjob	pa
MS	18	services	ye

school	age	Mjob	pass
MS	21	other	no
MS	19	other	no

The decision tree classifier picks a criterion (ie. School, famsize, mjob) to split the data on based on the possible values of that criterion. The process is repeated for each subset using a different criterion until all the students in each subset of the data are all either passing students or failing students.

In the predictive phase, the classifier uses the tree it has constructed to determine which subset a new example falls into. So if the school for the new example was GP then the classifier would follow the path on the left. It will walk down the tree until it could go no further. If it end in a subset with all passing students the it classifies the new example as a passing student.

After fine-tuning the decision tree model using gridsearch, the final F1 score of the model was 0.8.