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 problem is best suited for classificaton. This is due to the binary nature of the output, where we require a determination of the student as "need early intervention", or not. The current problem does not include a need to predict a specific value, such as a grade, for each student. If it did, this would be a regression problem.

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

Number of features: 30

Graduation rate of the class: 67.09%

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

See notebook for these answers

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.

Experimentation showed three different models that may best classify the data: LogisticRegression, AdaBoost, and KNeighbors.

Logistic Regression

- Space& Complexity - LogisticRegression is fast to query(test/train Delta), and is an eager learner, implying a lower resource usage and quicker prediction times.
- General Applications(strengths/weaknesses) - This model is used for binary classification type problems.It has many advantages, including speed, and it may be able to avoid overfitting in many cases. It does require more data to perform at

it's best, as can be seen in the charts above. Additionally, the data should be linearly seperable.

- Why this model? This model is a good fit due to it's classification capabilities and it's speed. Another advantage in this case is that it is possible to output a probability, to use, for instance, as a probability that a student is in need of intervention.

AdaBoost

- Space& Complexity - AdaBoost is a boost algorithim, and we can see that it has the longest training time in any of the candidates (by 2 order of magnitude), although prediction time is good.
- General Applications(strengths/weaknesses) - AdaBoost is a powerful classifier that has the advantage of needing less configuration to acheive good results, as well as being good at avoiding overfitting due to the use of multiple weak learners. It is however more sensitive to noisy data, and has very high resource usage.
- Why this model? This model may be a good fit to it's ability to acheive good results with little training, and the ability to avoid overfitting the data

KNeighbors

- Space& Complexity - KNeighbors is a lazy learner, and requires more memory resources, although training time is very fast. Prediction time(testDelta&trainDelta) are highest of the candidates. This model uses the neighbors with similar features features to predict the label.
- General Applications(strengths/weaknesses) - KNeighbors ability as a binary classifier make it an appropriate candidate. It has the advantage of quick training, but is held back by slowed prediction time. It also has an advantage in that data may be added to the model easily as it is aquired.
- Why this model? This model has the advantage of speedy training with good accuracy as a classifier, and would be well suited to an application that continuously adds data but has a reduced need for predictions.

training data size: 100

	testF1	testDelta	trainF1	trainDelta	timeToTrain
LogisticRegression	0.771429	0.000	0.885906	0.000	0.003
KNeighborsClassifier	0.741259	0.124	0.847682	0.118	0.001
AdaBoostClassifier-plain	0.725926	0.007	0.992806	0.006	0.101

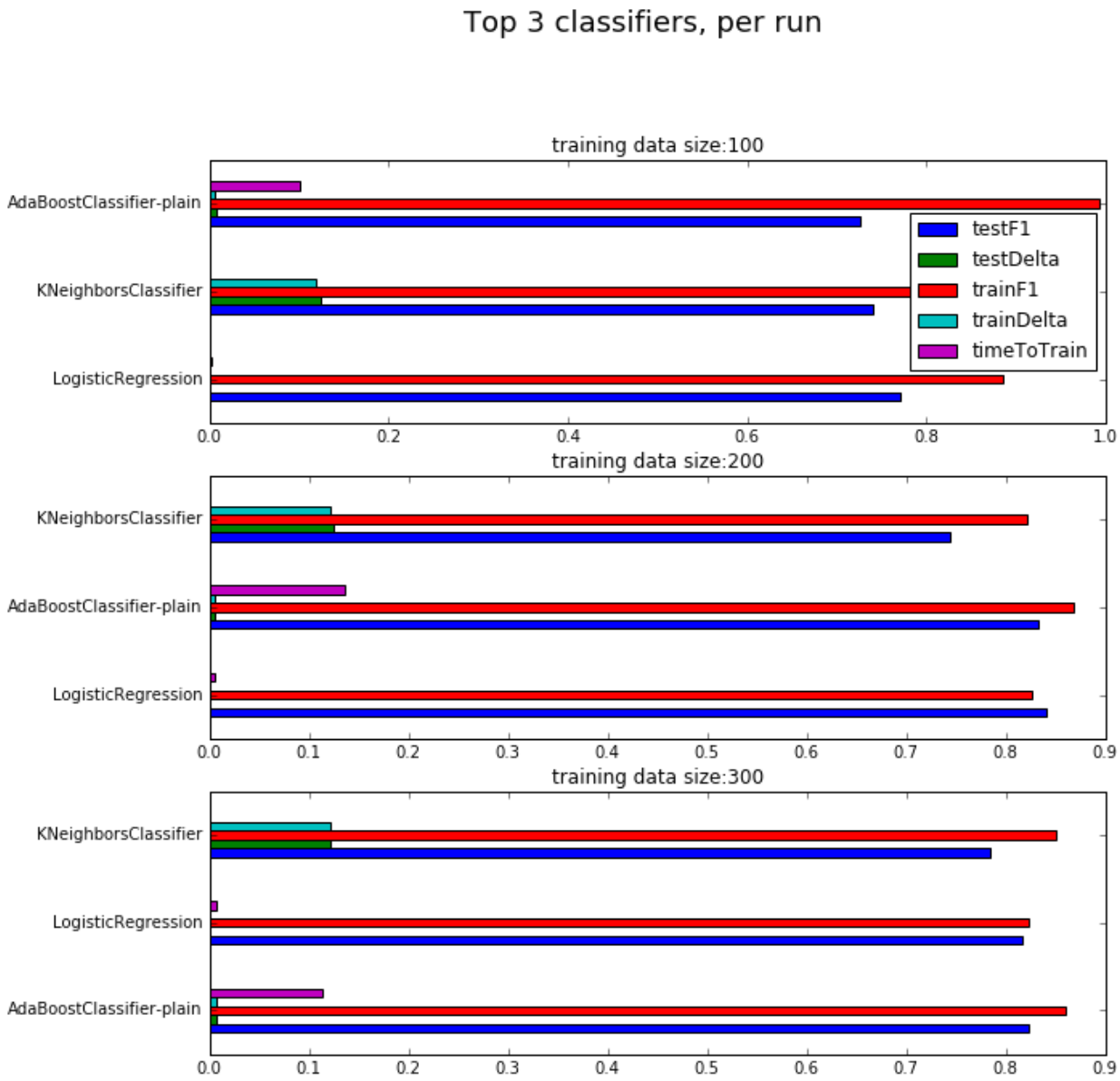
training data size: 200

	testF1	testDelta	trainF1	trainDelta	timeToTrain
LogisticRegression	0.841379	0.001	0.825503	0.000	0.006
AdaBoostClassifier-plain	0.833333	0.006	0.868966	0.005	0.136
KNeighborsClassifier	0.744828	0.124	0.821053	0.121	0.000

training data size: 300

	testF1	testDelta	trainF1	trainDelta	timeToTrain

AdaBoostClassifier-plain	0.823529	0.007	0.860520	0.007	0.113
LogisticRegression	0.816901	0.001	0.822727	0.001	0.007
KNeighborsClassifier	0.783784	0.121	0.850467	0.122	0.001



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?

- We have selected Logistic Regression as our initial classifier due to it's speed and low resource usage. As more data is added to the model, we should continue to expect lower resource usage than other models and fast prediction speed. We also have the possibility to express predictions about students as a probability of the need for intervention. As compared to the other tested classifiers, we may see that time taken for prediction is very significantly lower than KNeighbors, and it's training time (as well as prediction time) is significantly lower than AdaBoost. While these may have been a good reason to choose the model, it also happens that it has performed the best even after tuning all models, according to F1 score.
- Logistic Regression works by coming up with an equation based on the existing student data that can give us the chance that a new student may need intervention.It assumes that each piece of data (age/health/absences/etc) about the student is unrelated(independant) to the rest of the data, and we also need 10-20 times as many students as we have independant data points about the student for the analysis to work effectively.

- It works out the equation that best predicts the pass/no pass value using the existing student data such as age/health/absences/etc. It does this by taking the natural log of the odds of each item of student information, and uses a technique to estimate the parameters which would by the most likely to produce the observed results(maximum likelihood). This is done for each of the independant student data fields to produce the values for the needed equation.
- A prediction about a student can then be made by using the equation generated from the prior phase. It uses the information we have about the student as inputs to the previously produced equation, to give us a prediction that they will pass or not. The students likelyhood of passing may be expressed in this prediction as passing/not passing, or as a probability to pass.
- After tuning the model below we see a final score of F1 score: 0.0.83870.