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%

*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?*
I choose SVM, decision tree, and AdaBoost with the base estimator being decision tree. They're all widely used classifiers for binary inputs. As the data are imbalanced, the number of students passed is much more than those failed, the F1 score on class "failed" is an appropiate metric.
Answer:
I choose SVM, decision tree, and AdaBoost with the base estimator being decision tree. All of them are generally applied in both regression and classification, and here we focus on the classification side.

The dataset has two characteristics. First it's small. Second it's imbalanced, the number of students passed is much more than those failed. All three models can work well on such data. That's why I choose them.
SVM:

Pros:

- Works well on a wide range of classification problems, even problems in high dimensions and that are not linearly separable.

Cons:

- Theoretically, the main problem of SVM is the choice of kernal. This can be partially solved by grid search.
- Practically, the high algorithmic complexity and extensive memory requirements are the main concern.

Decision tree

Pros:

- Concept is simple.

Cons:

- Assumes the problem is linearly separable.
- Easily overfits. Sensitive to noise.
- When input features increases, the model complexity increases exponentially, which makes it unwieldy.

AdaBoost

Pros:

- Requires less tweaking of parameters or settings.
- Can be less susceptible to the overfitting problem than most learning algorithms.

Cons:

- Relatively expensive.

*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.*

Answer:

| **SVM** | **Training set size** | | |
|---|---|---|---|
| | 100 | 200 | 300 |
| Training time (secs) | 0.002 | 0.003 | 0.007 |
| Prediction time for training set (secs) | 0.001 | 0.002 | 0.006 |
| Prediction time for test set (secs) | 0.001 | 0.001 | 0.002 |
| F1 score for training set | 0.85549132948 | 0.869009584665 | 0.862579281184 |
| F1 score for test set | 0.782051282051 | 0.802721088435 | 0.810810810811 |

| **Decision tree** | **Training set size** | | |
|---|---|---|---|
| | 100 | 200 | 300 |
| Training time (secs) | 0.001 | 0.001 | 0.002 |
| Prediction time for training set (secs) | 0.000 | 0.000 | 0.000 |

| | | | |
|---|---|---|---|
| Prediction time for test set (secs) | 0.000 | 0.000 | 0.000 |
| F1 score for training set | 1.0 | 1.0 | 1.0 |
| F1 score for test set | 0.753623188406 | 0.701492537313 | 0.688524590164 |

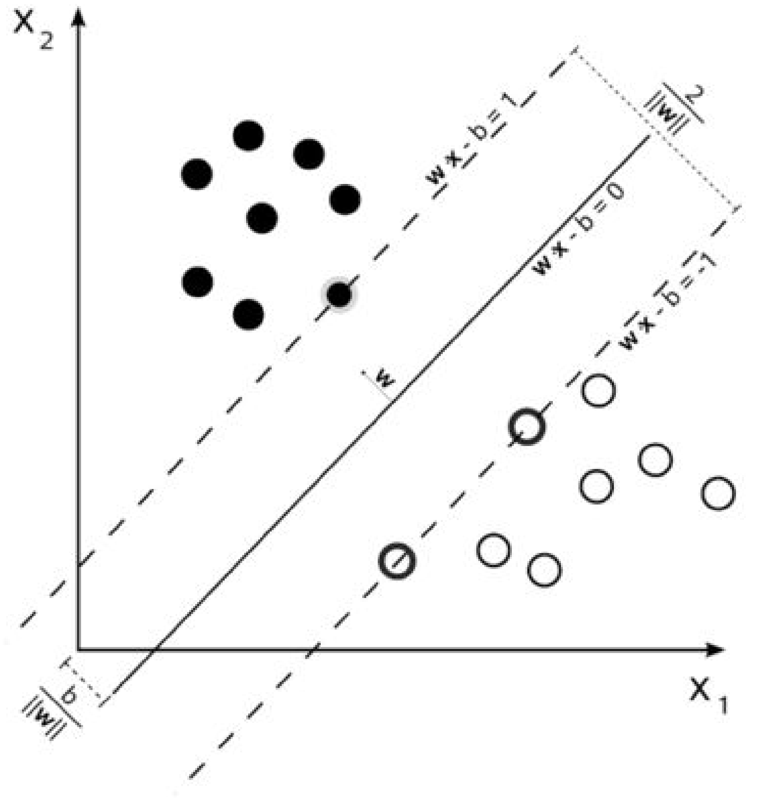
| **AdaBoost** | **Training set size** | | |
|---|---|---|---|
| | 100 | 200 | 300 |
| Training time (secs) | 0.048 | 0.069 | 0.062 |
| Prediction time for training set (secs) | 0.007 | 0.010 | 0.007 |
| Prediction time for test set (secs) | 0.007 | 0.007 | 0.006 |
| F1 score for training set | 0.957142857143 | 0.878378378378 | 0.855172413793 |
| F1 score for test set | 0.704 | 0.776119402985 | 0.755905511811 |

*Based on the experiments you performed earlier, in 1-2 paragraphs explain to the board of supervisors what single model you chose as the best model. Which model is generally the most appropriate based on the available data, limited resources, cost, and performance?*

Answer: Based on the tables above, SVM is the best. It gives the best results with very low cost. AdaBoost it the most expensive one and doesn't give better results. Considering the limited computing resources, it shouldn't be chosen. The decision tree is cheap but doesn't give as good results as SVM does. As the result, SVM is the most appropriate model for further tuning.

*In 1-2 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 make a prediction).*

Answer:



SVM is a type of linear separator. Suppose we want to split black circles from the white ones above by drawing a line. Notice that there are an infinite number of lines that will accomplish this task. SVMs, in particular, find the "maximum-margin" line - this is the line "in the middle". Intuitively, this works well because it allows for noise and is most tolerant to mistakes on either side.

*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?*
Answer: After tuning, for decision tree, the F1 scores for training and test sets are 0.833684210526 and 0.821917808219 respectively. Compared to untuned model, the training score doesn't change too much while test score significantly increases. We get a better model.