Classification vs Regression

This is a classification problem because the target column is made up of categorical data. If it will have been a regression problem if the target column had been made up of real / continuous values.

Exploring the Data

- 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

Training and Evaluating Models

Decision Tree Classifer

Reason for using this classifier:
I choose Decision Tree Classifiers because

- it is a classifier that can easily be visualized by a non-technical audience
- once trained, predictions is done in logarithmic time
- it is capable of binary classification, this is useful since the result of our prediction is binary(whether a student passes or fails).

The Advantages:

- The decision tree can be easily visualized, which makes it easy to understand and interpret
- They require little data preparation
- The storage cost for generating the prediction tree is logarithmic relative to the quantity of data provided

The Disadvantages:

- Decision Trees can get overly complex performing well during training but not during prediction - a situation known as overfitting. There are ways to deal with overfitting.
- Decision Trees need to be rebuilt and can result in a completely new tree once there is a variation in the original data.
- Decision tree learners create biased trees if some classes dominate. It is therefore recommended to balance the dataset prior to fitting with the decision tree

Support Vector Classifier

Reason for using this classifier

I also choose Support Vector Classifier because

- They work well in complicated domains where there are a lot of features relative to the size of data available.
- There is not a lot of noise in our dataset. SVCs work well when the size of the dataset is small
- Our dataset does not contain a lot of noise. That makes it a good candidate for SVCs.

The advantages:

- Effective in high dimensional spaces - They could still provide good prediction when the number of features 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.

The disadvantages:

- It could have poor performance if the number of features is much greater than the number of samples
- It does not provide probability estimates.

Gaussian Naive Bayes

These are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of independence between every pair of features. In spite of their apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many real-world situations, famously document classification and spam filtering.

The Advantages:

- They require a small amount of training data to estimate the necessary parameters.
- 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

The Disadvantages:

- it is known to be a bad estimator

Classifier	F1 score(test)	F1 score(train)	Size	Train time	predict time
Decision Tree Classifier	0.650407	1.000000	100	0.001	0.000
Decision Tree Classifier	0.661017	1.000000	200	0.002	0.000
Decision Tree Classifier	0.837209	1.000000	300	0.003	0.000
SVC	0.800000	0.858896	100	0.002	0.001
SVC	0.815789	0.872131	200	0.006	0.002
SVC	0.828947	0.846316	300	0.037	0.005
GaussianNB	0.225000	0.409091	100	0.002	0.001
GaussianNB	0.738462	0.804270	200	0.006	0.001
GaussianNB	0.805755	0.817156	300	0.002	0.000

Choosing the Best Model

Reviewing the experiment's result

Based on my experiments carried out previously, I believe Decision Tree Classifiers is generally more appropriate based on the available data. To understand why, I will review the three models here in terms of Performance, Train time and Prediction Time. Remember, that one of our goals here is to use as little computation as possible.

Decision Trees

The average F1 Score 0.716 across all training sizes. The training time increased linearly with the training size and the prediction time is negligible.

Support Vector Classifiers.
The average F1 Score 0.826 across all training sizes. The training time increased exponentially with the training size and so does the prediction time. This model does have a better f1 score but it takes the most amount of time to both train and predict

Gaussian Naive Bayes
The average F1 Score 0.59 across all training sizes. The training time has no clear pattern relative to the training size, however, it is considerable low. The prediction time appears near constant, but it is more than the decision tree.

I would argue that the decision tree classifier is a better model for our data because it gives a good f1 score, get trained fast and also predicts data in negligible time.

How Decision Trees work

A decision tree is a set of rules used to classify data into categories. It looks at the variables in a data set, determines which are most important, and then comes up with a, tree of decisions which best partitions the data. The tree is created by splitting data up by variables and then counting to see how many are in each bucket after each split.
Imagine you were playing a guessing game where your opponent has a secret answer, but allows you to ask true or false questions. He then tells you if the answer to your question is true or false. How do you find the secret answer in the fewest number of questions? Let us assume that the game is to tell you if a student passes or fails. Some of the questions you can ask are Is the student in a rural or urban area?, Is the student on education support?,Does he have internet, is the student male or female, and so on. The algorithm groups tries to create a tree. To find out if a student passes or fails, you simply start asking questions. Each answer gives you a smaller question tree or tells you if the student will pass / fail. The tree is constructed to require the smallest number of questions to make a prediction.

Explaining the Final Model

My Model's final f1 score is 0.8102 To do arrive at that score, gridsearch showed that setting a maximum depth to 2 nodes and maximum features using the square root of features produces the best model f1 score.

Disclaimer:
Some of the writings here were based off sklearn's documentation[http://scikit-learn.org/stable].
This is because I found their explanations to appropriately describe classification models.