Question 1 - Classification vs. Regression

Your goal for this project is to identify students who might need early intervention before they fail to graduate. Which type of supervised learning problem is this, classification or regression? Why?

Answer: Classification. Because we actually distinguish whether a student need intervention. Whether or not, the answer is limited and discrete, so it's a classification problem.

Question 2 - Model Application

List three supervised learning models that are appropriate for this problem. What are the general applications of each model? What are their strengths and weaknesses? Given what you know about the data, why did you choose these models to be applied?

Answer: The models I choose are Decision Tree, Support Vector Machine, and K-Nearest Neighbors.

Decision Tree: Classifying By learning simple decision rules inferred from the data features. General application are given many attribute and give out a decision, like bank credit system evaluates whether a person can repay the loan given the profile of him.
Pros: Simple to understand and interpret; Requires little data preparation; Able to handle numerical and categorical data; Able to handle multi-output problems.
Cons: Decision-tree learners easily over-fitting, need prune or set max depth to deal with; Unstable and easily influenced by outliers and dominated classes; Heuristic algorithms sometimes cannot get the globally optimal.
The reason I choose is this is a simple model which good at classification problem, and only need little data.

Support Vector Machine: A set of supervised learning methods used for classification, regression and outliers detection. SVMs are widely used, a example is speech recognition. Pros: Effective in high dimensional spaces; Memory efficient; Versatile, with different kernel functions can specified for the decision function. Cons: SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.
The reason I choose this is because it's versatile and widely used.

K-Nearest Neighbors: A straightforward way to classify for data with discrete labels. General application of the method are clustering, like recommend system to clustering client into similar interest group.
Pros: Easy to implement with only a few hyperparameters to tune; Good classification to outliers; Especially suit multi-classification problem, that's because this model can also deal with unsupervised problem.
Cons: Lazy calculation, whose computational cost and memory cost are high; Hard to interpret, can not conclude a decision rules.

The reason I choose this is take a bit at this method, which is good at classification but quite different from the previous two.

Question 3 - Chosing the Best Model

Based on the experiments you performed earlier, in one to two 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: I'd like to choose SVM as the best model. Considering the prediction time, SVM is a little more time-consuming than Decision Tree, but quite less than K-Nearest Neighbors. And SVM gets the highest F1 Score on test set, which means it's good on generalization. Besides, the data usage by SVM is also small, which only feed a training set size of 100 to reach a good model.
Decision Trees, on the contrary, gets full F1 Score on training set but rather poor on test sets. Which means the overfitting problem occurs and poor generalization ability. And KNN method, good result on F1 score but time-consuming, makes it's not the best choice.

Question 4 - Model in Layman's Terms

In one to two paragraphs, explain to the board of directors in layman's terms how the final model chosen is supposed to work. For example if you've chosen to use a decision tree or a support vector machine, how does the model go about making a prediction?

Answer: An example of using SVM in classifying: Suppose some given data points each belong to one of two classes, and the goal is to decide which class a new data point will be in. In the case of support vector machines, a data point is viewed as a p-dimensional vector (a list of p numbers), and we want to know whether we can separate such points with a (p-1)-dimensional hyperplane. This is called a linear classifier. There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest margin between the two classes. So when we have new points to predict, the largest margin can reduce error if the new point is near the margin hard to be classified.

The training processing is simple, giving a set of data and trying to find the hyperplane so that the distance from it to the nearest data point on each side is maximized. And the predict process is to project the new data point into our p-dimensional space, and distinguish which sub-space separated by hyperplanes the point belongs to. The SVM can also apply to non-linear data. By using kernel trick, in another word, applying nonlinear kernel function instead of original dot product can transform original feature space into high dimensional one. Although the classifier is a hyperplane in the transformed feature space, it may be nonlinear in the original input space. That's how SVM applying to nonlinear data.

Question 5 - Final F_1 Score

What is the final model's F_1 score for training and testing? How does that score compare to the untuned model?

Answer: F1 score for training : 0.8894

F1 score for testing : 0.8387

F1 on training is increasing, and F1 on testing almost unchanged. Which means the model is a little bit more precise, but the generalization ability doesn't improve. It's not the model limits the performance but the insufficient data.