Student Intervention Report

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 is a classification type of machine learning problem. You are trying to put students into discrete buckets - this implies classification problem. If you were trying to measure how much intervention a student requires (i.e. a continuous output variable), then it 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 Starter code snippets for these steps have been provided in the template.

See iPython Notebook

4. Training and Evaluating Models

Choose 3 supervised learning models that are available in scikit-learn, and appropriate for this problem. For each model:

Support Vector Machine (SVM)

What are the general applications of this model

This model is good for finding the best boundary between classes of data. For our example it can separate graduates from non-graduates using all of the information we have available.

What are its strengths and weaknesses?

SMV is good at simplifying complex data to get a boundary between the classes. It can take some time to train the data and it is more complex than the other algorithms.

Given what you know about the data so far, why did you choose this model to apply?

SVM is good at finding the boundary between classes (for us, graduates and non-graduates). It can find the complex boundaries with minimal data. It is also relatively fast in finding the boundary.

Training set size	**100**	**200**	**300**
Training time (secs)	0.002	0.005	0.010
Prediction time (secs)	0.001	0.002	0.002
F1 score for training set	0.8824	0.8814	0.8785
F1 score for test set	0.7619	0.7638	0.7919

Decision Tree

What are the general applications of this model

Decision trees are models that basically play the game "Twenty Questions" with the data to form a boundary. It keeps asking questions until it can properly classify the data.

What are its strengths and weaknesses?

Decision trees are fast. Unfortunately, they tend to overfit the training set. This can cause them to be less effective when given new information.

Given what you know about the data so far, why did you choose this model to apply?

Decision trees are a good way to classify data. They are fast, easy to understand, and fairly quick to train.

Training set size	**100**	**200**	**300**
Training time (secs)	0.001	0.002	0.003
Prediction time (secs)	0.000	0.000	0.000
F1 score for training set	1.000	1.0000	1.0000
F1 score for test set	0.6140	0.7244	0.7384

K-Nearest Neighbors (KNN)

What are the general applications of this model

The old saying goes, "Birds of a feather, flock together." That is the principle of K-Nearest Neighbors. It stands to reason that students with similar profiles would be similar in graduation rate.

What are its strengths and weaknesses?

KNN is very fast to train, it is also fairly accurate - even with new inforamation. But, it keeps all of the information, so it can be slow to predict.

Given what you know about the data so far, why did you choose this model to apply?

KNN is another model that is fairly intuitive. It is generally accurate and easy to train.

Training set size	**100**	**200**	**300**
Training time (secs)	0.001	0.001	0.001
Prediction time (secs)	0.003	0.009	0.003
F1 score for training set	0.8281	0.8472	0.8644
F1 score for test set	0.7077	0.7353	0.7801

5. Choosing the Best Model

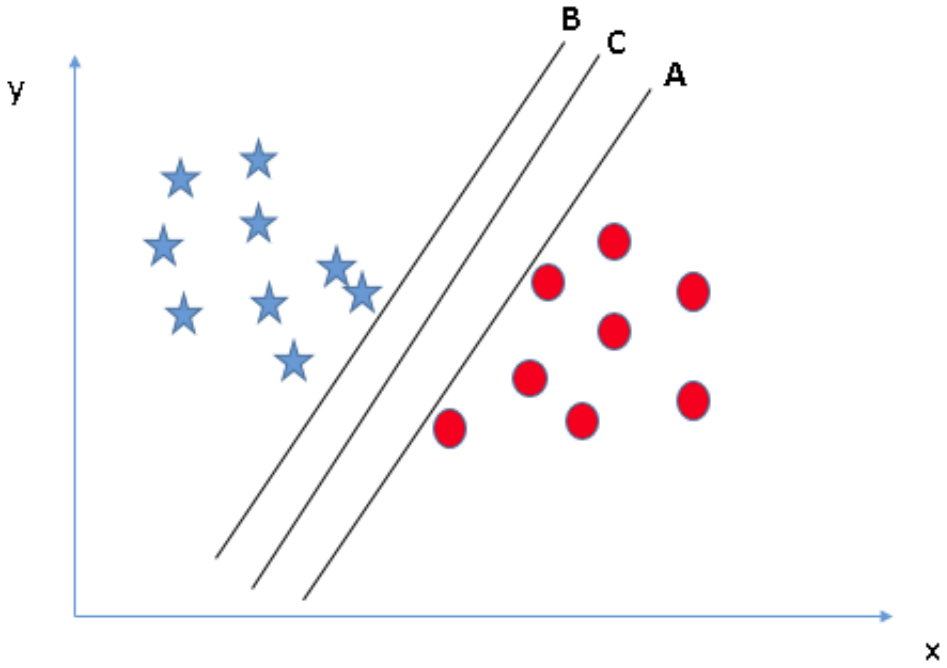
Based on the experiments you performed earlier, in 2-3 paragraphs explain to the board of supervisors what single model you choose as the best model. Which model has the best test F1 score and time efficiency? Which model is generally the most appropriate based on the available data, limited resources, cost, and performance? Please directly compare and contrast the numerical values recored to make your case.

Based on the experiments performed above with four different models, I recommend the Support Vector Machine Model (SVM). Looking at the tables above, one can see that both SVM and K-Nearest Neighbors (KNN) performed best on the test set. When considering the each of the training and prediction times, KNN performs about the same as SVM. However, due to its nature, KNN requires significantly more prediction time than training time. Having run the model several times, it looks to me as though the SVM training time for 300 datapoints is a bit of an anomaly of the run itself and should not be construed to generalize the time it will take the model. (Results are kept as is to match the iPythin Notebook.

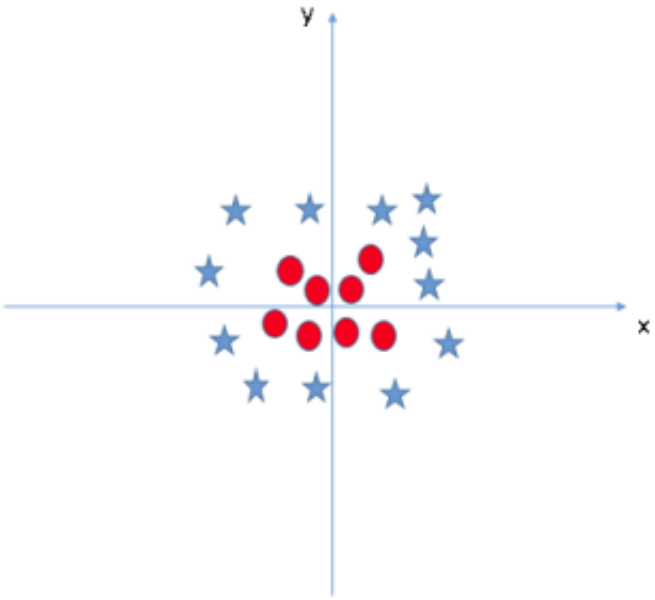
This model will be used more often for predicting how students will perform and thus prediction time will be more important than training time. Additionally, SVM was able to get the best result using the least amount of data (100 points). This will aid us in when we have limited ability to gather data.

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).

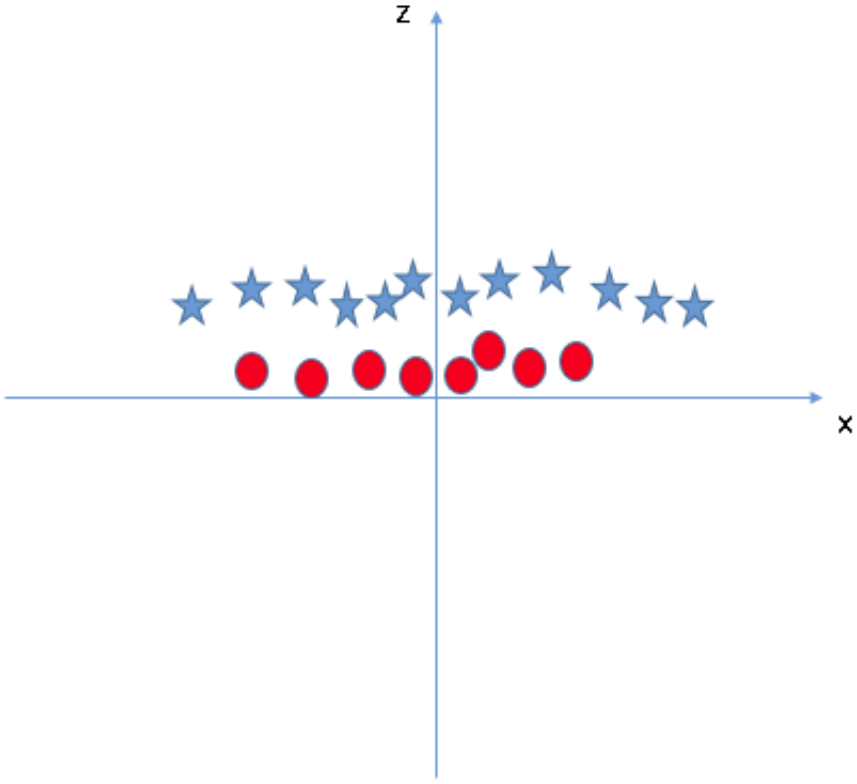
Our goal with this model is to correctly classify those students that will graduate vs. not graduate. So we want the model to divide each of the students into one of two groups. Support Vector Machines (SVM) work by finding the dividing line between the students into those two groups. In training the SVM the algorithm finds the "support vectors" - or lines at the edge of each of the two groups. (See Lines A & B below). It chooses the best line - where best is the line that has the most margin (or space) between the two groups. (See Line C below)



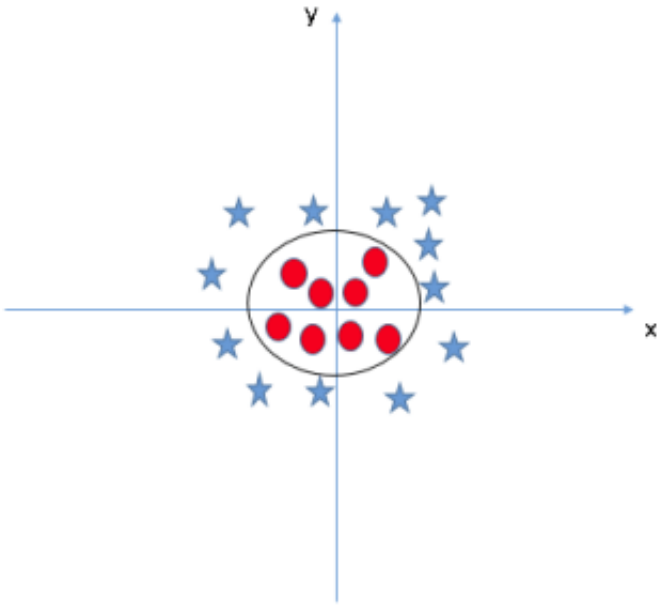
This is an easy task if you just have one or two inputs that are easily separated by a line. More often our data looks something more like the image below, meaning that it is not just a straight line that can separate the two groups.



In cases like this, SVM uses a "kernel trick" or mathematical substitution that will transform the data into something easier to separate with a line like the image below.



As you can see this makes the data easy to divide with a straight line. When we translate back to the original we get the image below.



To predict whether a new student might graduate or not, the SVM would categorize the new student based on which side of the line the new student sits.

Images from [http://www.analyticsvidhya.com/blog/2015/10/understaing-support-vector-machine-example-code/ (http://www.analyticsvidhya.com/blog/2015/10/understaing-support-vector-machine-example-code/)](http://www.analyticsvidhya.com/blog/2015/10/understaing-support-vector-machine-example-code/)

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.

See iPython Notebook

What is the model's final F1 score?

F1 score for test set: 0.8050