P2: Build a Student Intervention System

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?

Classification, because I'm trying to predict whether the student succeeded or failed and this is a discrete output, in other words the output is not a set of continuous numbers it's just a set of two classes.

2. 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: 67.09%`

4. Training and Evaluating Models

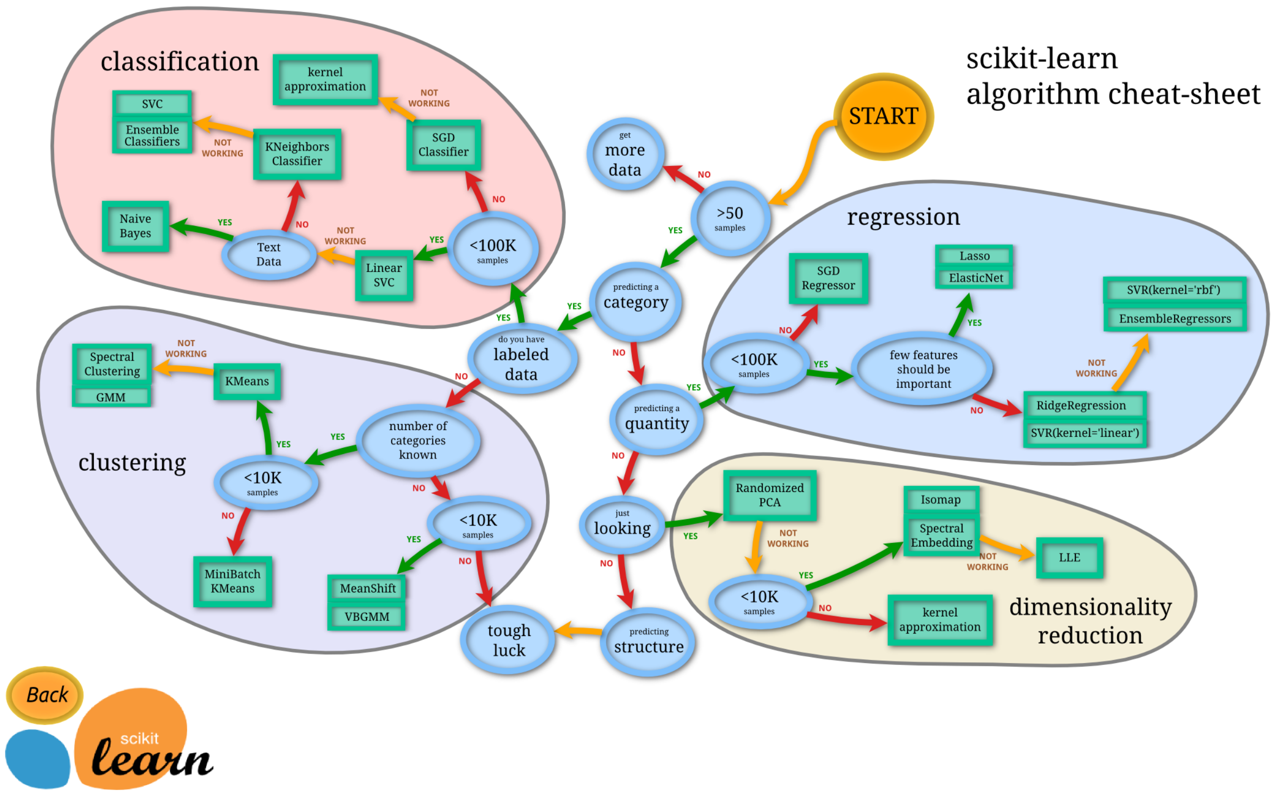
What are the general applications of this model? What are its strengths and weaknesses?

KNN –
Is a lazy learner which means it doesn't take time to learn instead it stores the data and the work is only done when it's asked to predict a given value, so it takes very small time to learn but on the other hand it takes more time to query (predict) and takes more space than other models, can do classification and regression.
SVM –Is an eager learner which learns as soon as it's given the data and then throws the data away so it takes less space than KNN and less predicting time than KNN but more training time.
SVM is not recommended for applications which has a lot of data as it takes a lot of time to be trained.
Decision Trees – Eager learners and take less space less predicting time more training time than KNN but less than SVM,
SVM and KNN are considered black boxes which means you can't visualize what's happening to the user, Decision trees are considered an open box so it's easy to visualize what's happening inside
One problem with trees are they can overfit easily if the level of the tree increases above a certain value and it needs pruning to overcome it, they

can’t solve regression problems.

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

From the data I knew that the problem is classification, given the many classification algorithms, there was no intuition of why I should choose DT, SVM, K-NN over other algorithms, after some searching I followed the bellow figure on sklearn documentation and from it I chose those three models, I think its ok to choose other models that can classify the data over the ones I already chose and only the score of my model on the testing set would tell me which ones are good and which are bad for my problem.



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.

SVM

	100	200	300
Training Time(sec)	0.002	0.004	0.008
Prediction Time(sec)	0.001	0.002	0.002
F1 score for training set	0.909	0.872	0.856
F1 score for test set	0.751	0.758	0.786

KNN

	100	200	300
Training Time(sec)	0.001	0.002	0.002
Prediction Time(sec)	0.001	0.002	0.003
F1 score for training set	0.846	0.851	0.853
F1 score for test set	0.781	0.755	0.779

Decision Tree

	100	200	300
Training Time(sec)	0.001	0.002	0.003
Prediction Time(sec)	0	0	0
F1 score for training set	1.0	1.0	1.0
F1 score for test set	0.755	0.655	0.693

5.Choosing The Best Model

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?

Based on the above tables, SVM takes a lot of training time as the data increases the training time increases by a factor of 2 (in KNN training time will be way less than that of SVM if we have bigger number of students as it doesn't train), prediction time for SVM is definitely less than KNN as the data increases, but KNN prediction time doesn't scale as fast as SVM training time, and they both have approximately same testing score, so I chose KNN over SVM as it's computationally better and about the same accuracy but we will sacrifice storage.

Training time of Decision trees is of course more than that of KNN but it takes no prediction time, but the decision tree already overfitted with the given data as it produces 1.0 score and way less testing score 0.693 , so decision tree doesn't generalize well in this case , I even tried to decrease the maximum depth of the tree when I initialized it to reduce overfitting but it continues to produce relatively high training score (approx. 0.9) and not good testing score(approx. 0.71), so I chose KNN over Decision tree mainly because it's accuracy is better.

In 1-2 paragraphs explain to the board of supervisors in layman's terms how the final model chosen is supposed to work.

K nearest neighbor is an algorithm it’s target is to predict the output of some data.
First it’s given some input data and their corresponding results (called training data), these data is stored maybe locally in a database and then the algorithm does nothing, it only took some space for the given data so it doesn’t search or try to conclude any patterns in the given training data in other words it doesn’t train on the given data it only stores it.
the algorithm does extra work only when it’s asked to predict the output of a certain new given input, it enters the database and searches for a value equals to the input if it found it, it will return the corresponding result, if it didn’t find the same value of the input it looks for it’s nearest neighbors, it may look for only one or more neighbors so that’s why it’s called K Nearest Neighbors, and nearest here means a lot of things based on the problem, it may mean nearest as in the nearest distance , or nearest as in relatively similar category to the input and many other things, and then it mixes (maybe taking the average or taking the most repeated result) the results of all the chosen nearest neighbors outputs and returns the final value.

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 F_1 score?

After tuning the number of nearest neighbors in the KNN algorithm, final F1 score over the testing set increased to be exactly 0.8