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 problem.
- Because the label variable is discrete, especially it is binary, ‘yes’ and ‘no’.

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
- Graduation rate of the class (%)
  - 67.09%
- Number of features (excluding the label/target column)
  - 30

Use the code block provided in the template to compute these values.

3. Preparing the Data ***# it seems not need any work here, dosen’t it?***

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.

4. Training and Evaluating Models

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?
- 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.

Model 1 - Decision Tree

- Decision trees are commonly used in operations research and operations management. Another use of decision trees is as a descriptive means for calculating conditional probabilities. Medical research and practice have long been important areas of application for decision tree techniques. *# come from wiki decision tree* and here.
- Advantages:
    - easy to understand and interpret
    - it is a white box model
    - powerful expressive
- Disadvantages:
    - easy to overfit without tuning parameters

- Lots of features' variable are discrete, and others' numeric variables are easy to discretize, which can well handle by decision tree. And decision tree as a white box what should accept by staff easily.
- *# I already wrote training size in code... and i don't want to change it, after all the more outcomes the easier to analyze*

Decision Tree	Training set size				
	300	252	203	155	107
Training time (secs)	0.002	0.002	0.002	0.001	0.002
Prediction time (secs)	0.000	0.000	0.000	0.000	0.000
F1 score for training set	1.000	1.000	1.000	1.000	1.000
F1 score for test set	0.7077	0.6720	0.6772	0.6885	0.6992

Model 2 - SVM

- SVMs are helpful in text and hypertext categorization as their application can significantly reduce the need for labeled training instances in both the standard inductive and transductive settings.Classification of images can also be performed using SVMs. # from wiki svm page
- Advantages: # from sklearn svm page
    - Effective in high dimensional spaces.
    - Still effective in cases where number of dimensions is greater than the number of samples.
    - memory efficient.
    - Versatile: different Kernel functions can be specified for the decision function.
- Disadvantages:

    - If the number of features is much greater than the number of samples, the method is likely to give poor performances.
    - SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation
    - SVM is a binary classifier, while dealing with multi-class classification, it has use one-vs-all method, which will significant increase training time.
- There are 30 features in this dataset, after preprogress some of features, there will have 48 features in dataset, that is a very high dimension data, and SVM is a very effective algorithm to deal with high dimensional spaces.
-

SVM-SVC	Training set size				
	300	252	203	155	107
Training time (secs)	0.029	0.005	0.004	0.002	0.001
Prediction time (secs)	0.005	0.003	0.002	0.001	0.001
F1 score for training set	0.8747	0.8943	0.8874	0.8889	0.8974
F1 score for test set	0.8212	0.8133	0.7973	0.8212	0.7867

model 3 - Logistic Regression(LR)

- LR is used widely in many fields, including the medical and social sciences. For example, the Trauma and Injury Severity Score (TRISS), which is widely used to predict mortality in injured patients, was originally developed by Boyd et al. using logistic regression. # from wiki logistic regression page
- Advantages:
    - robust
    - can handle non-linear effects
    - no homogeneity of variance assumption
    - Normally distributed error terms are not assumed
- Disadvantages:

    - requires much more data to achieve stable, meaningful results
    - LR is a binary classifier, while dealing with multi-class classification, it has use one-vs-all method, which will significant increase training time.
- LR needs at least 50 data points to achieve stable, meaningful results, here i have enough data points to train, LR is a great classifier algorithm too, why not give it a shoot? The most important reason why choose LR is because it's robust, no one want to see an unstable algorithm at all.
-

Logistic Regression	Training set size				
	300	252	203	155	107
Training time (secs)	0.361	0.014	0.006	0.003	0.001
Prediction time (secs)	0.157	0.001	0.001	0.000	0.001
F1 score for training set	0.8356	0.8177	0.8362	0.8519	0.8816
F1 score for test set	0.7737	0.7463	0.7519	0.7442	0.7188

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.

- The SVC should be the best model, because it has highest test set f1_score among three models, also it was very efficiency. F1_score seems stable while the training set size going down.

Which model has the best test F1 score and time efficiency?

- SVC model has the best f1_score but not the best efficiency.
- Decision tree model has the best time efficiency but the worst f1_score, and according to it's highest training f1_score a.k.a 1.0, I am strongly suspecting this model is overfitting.

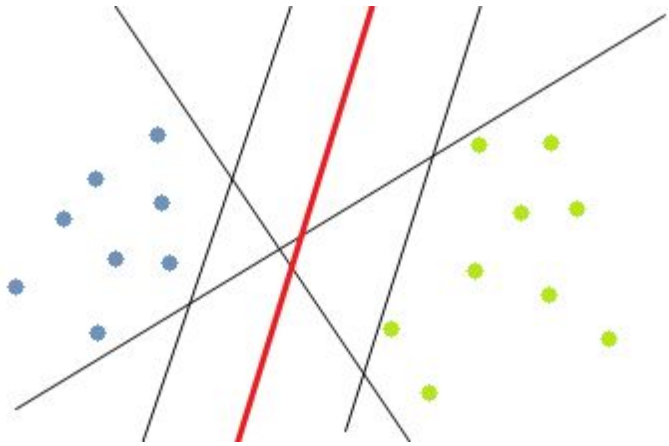
Which model is generally the most appropriate based on the available data, limited resources, cost, and performance? Please directly compare and contrast the numerical values recored to make your case.

- The SVC model generally the most appropriate based on acaliable data, limited resources, cost and have the best performance.
- Although SVC model has longer training time than decision tree model, but it is still within the accpetable range. And we care about accuracy much more than training time.
- While training set size limit to 107, SVC model still have the highest f1_score 0.7867

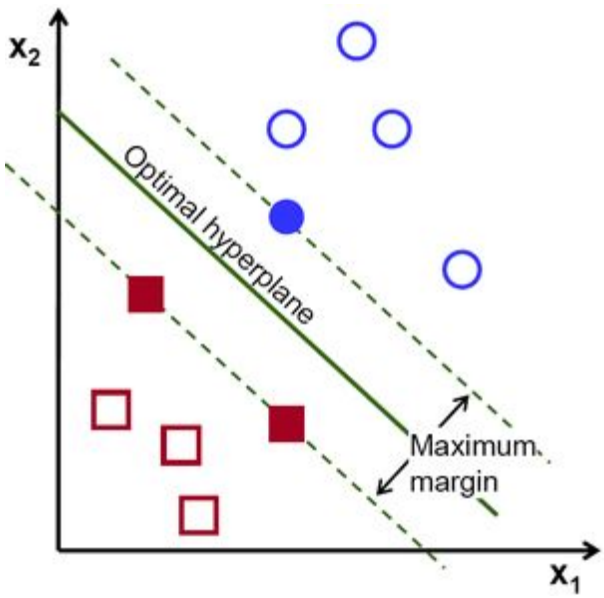
In 1-3 paragraphs explain to the board of supervisors in layman's terms how the final model chosen is supposed to work (for example if you chose a decision tree or support vector machine, how does it learn to make a prediction).

- SVM always tries to find out a best line (in high dimensional spaces we call it hyperplane) a.k.a the decision boundry to predict correctly, the best line means a line

that far away from all correct predict point as far as it can, so that SVM will be robust.

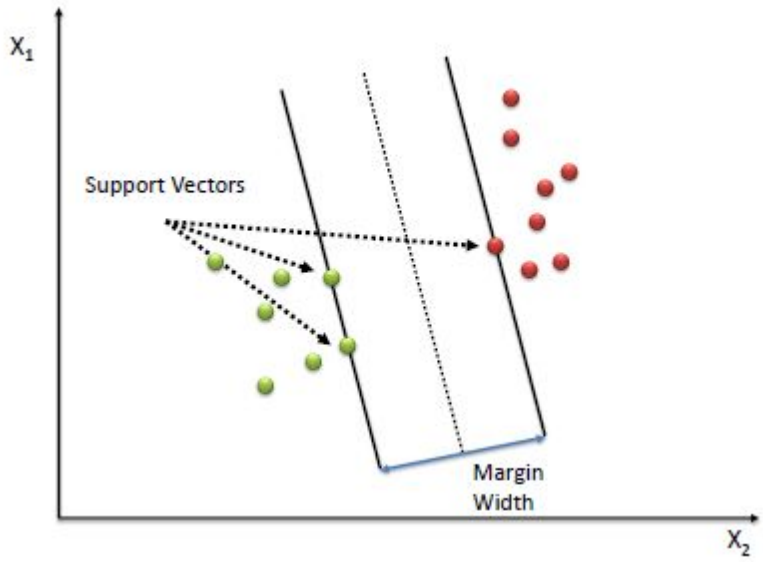


- But in actual performance, we use margin rather than line, because it would make the math behind SVM easier, you may have an intuition of margin with the image below, we call the area bewteen two lines of dashes as margin.

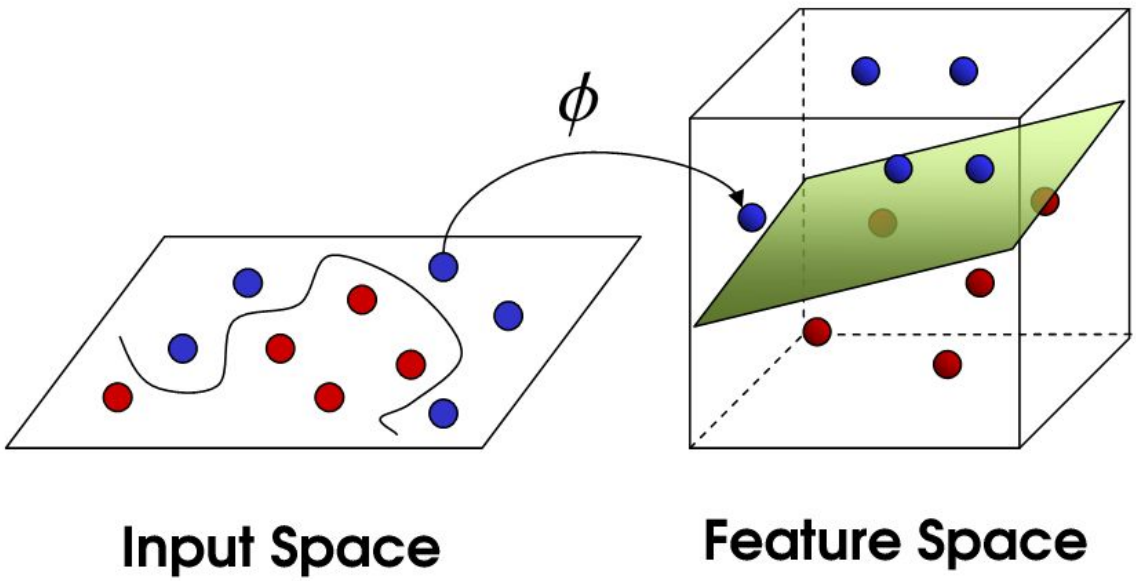


- Among all data points, only those points which close to margin boundry actually useful, which we call it support vector, and that's why we call this algorithm as

support vector machine.



- When SVM need solve a classifiation task with lots of features like student intervention, it will use the kernel trick transform it in high dimensional spaces, then find out a best hyperplane.



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?

- The best parameters are {'kernel': 'rbf', 'C': 0.01}
- The final training f1_score is 0.799996815892
- The final test f1_score is 0.8125

Here is a problem confusing me while tuning the SVC model.

When i use grid_searchCV tuning my model automatically, i actually write {'kernel':('linear','rbf','poly')...} in variable ***parameters***, and grid_searchCV gave outcome as

best_parameter: {'kernel': 'linear', 'C': 0.03}

best_training_f1_score: 0.818769947264

best_test_f1_score: 0.794701986755

It seems like grid_searchCV choose this parameter setting beacause of it's higher f1_score, but this model performance worser in the test set. This discovery makes me wonder, is it true that no even grid_searchCV can find out the 'best' parameter combine among all pre-giving parameters? Well, 'best' define as most generalized model.

(In order to get a pretty outcome, i had deleted 'linear' form parameter.)

One of my explanation is that there are some bias hiding in this test set and producing a untrustworthy outcome, but i don't sure if this explanation is right.

Wish can get a from viewer, and thanks a lot.