**Project Description**

As education has grown to rely more on technology, vast amounts of data has become available for examination and prediction. Logs of student activities, grades, interactions with teachers and fellow students, and more, are now captured in real time through learning management systems like **Canvas** and **Edmodo**. This is especially true for online classrooms, which are becoming popular even at the primary and secondary school level. Within all levels of education, there exists a push to help increase the likelihood of student success, without watering down the education or engaging in behaviors that fail to improve the underlying issues. Graduation rates are often the criteria of choice, and educators seek new ways to predict the success and failure of students early enough to stage effective interventions.

A local school district has a goal to reach a 95% graduation rate by the end of the decade by identifying students who need intervention before they drop out of school. As a software engineer contacted by the school district, your task is to model the factors that predict how likely a student is to pass their high school final exam, by constructing an intervention system that leverages supervised learning techniques. The board of supervisors has asked that you find the most effective model that uses the least amount of computation costs to save on the budget. You will need to analyze the dataset on students' performance and develop a model that will predict the likelihood that a given student will pass, quantifying whether an intervention is necessary.

You may also visit our [**Projects GitHub**](https://github.com/udacity/machine-learning) to have access to all of the projects for this Nanodegree, which includes the **Building a Student Intervention System** project.

**Project Files**

For this assignment, you can find the project files as a downloadable in the Resources section asstudent\_intervention.zip. You may also visit our [**Projects GitHub**](https://github.com/udacity/machine-learning/tree/master/projects) to have access to all of the projects available for this Nanodegree. While some code has already been implemented to get you started, you will need to implement additional functionality to successfully answer all of the questions included in the notebook. You can find the included questions for reference on the following slide.

**Software and Libraries**

For this project, you will need to have the following software installed:

* [**Python 2.7**](https://www.python.org/download/releases/2.7/)
* [**NumPy**](http://www.numpy.org/)
* [**pandas**](http://pandas.pydata.org/)
* [**scikit-learn**](http://scikit-learn.org/stable/)
* [**iPython Notebook**](http://ipython.org/notebook.html)

**Deliverables**

The following files should be included as your submission, and can be packaged as a single zip file for convenience:

* The student\_intervention.ipynb file with fully implemented, functional code, with all code blocks executed and showing output.
* An HTML or PDF report of the project (you may either export the notebook as HTML with your answers included, or submit a separate PDF with only the questions and your answers).

You may also visit our [**Projects GitHub**](https://github.com/udacity/machine-learning) to have access to all of the projects for this Nanodegree, which includes the **Building a Student Intervention System** project.

## Questions and Report Structure

**Your model will be evaluated on three factors:**

* Its [**F1 score**](https://en.wikipedia.org/wiki/F1_score), summarizing the number of correct positives and correct negatives out of all possible cases. In other words, how well does the model differentiate likely passes from failures?
* The size of the training set, preferring smaller training sets over larger ones. That is, how much data does the model need to make a reasonable prediction?
* The computation resources to make a reliable prediction. How much time and memory is required to correctly identify students that need intervention?

### 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?

### 2. Exploring the Data

Can you find out the following facts about the dataset?

* Total number of students
* Number of students who passed
* Number of students who failed
* Graduation rate of the class (%)
* Number of features (excluding the label/target column)

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

### 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.

### 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**](https://docs.google.com/document/d/1Goxw-6M0umOCokCFqTr-g7SU_5f6MIm_wqXmh9Cnjhw/pub) 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.

### 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.

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).

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?