1. Classification vs Regression

This goal as stated by the problem is to identify and classify if a given student is likely to require intervention or not. The goal is to predict an output which belongs to either one of the classes:

- Does not require intervention
- Requires intervention.

Hence this is a Classification problem. Regression problems are applicable to continuous space.

2. Exploring the Data

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: **30**

3. Preparing the Data

```
Feature column(s):-
['school', 'sex', 'age', 'address', 'famsize', 'Pstatus', 'Medu', 'Fedu',
'Mjob', 'Fjob', 'reason', 'guardian', 'traveltime', 'studytime', 'failures',
'schoolsup', 'famsup', 'paid', 'activities', 'nursery', 'higher', 'internet',
'romantic', 'famrel', 'freetime', 'goout', 'Dalc', 'Walc', 'health',
'absences']
Target column: passed

Feature values:-
  school sex  age address famsize Pstatus  Medu  Fedu     Mjob      Fjob  \
0     GP   F   18       U     GT3       A     4     4  at_home   teacher
1     GP   F   17       U     GT3       T     1     1  at_home     other
2     GP   F   15       U     LE3       T     1     1  at_home     other
3     GP   F   15       U     GT3       T     4     2   health  services
4     GP   F   16       U     GT3       T     3     3    other     other

     ...    higher internet  romantic  famrel  freetime goout Dalc Walc  \
0    ...       yes       no        no       4         3     4    1    1
1    ...       yes      yes        no       5         3     3    1    1
2    ...       yes      yes        no       4         3     2    2    3
3    ...       yes      yes       yes       3         2     2    1    1
4    ...       yes       no        no       4         3     2    1    2

  health absences
```

```
0      3        6
1      3        4
2      3       10
3      5        2
4      5        4

[5 rows x 30 columns]
```

4. Training and Evaluating Models

Support Vector Machines:

General Applications:

- Solving Classification problems.

Strengths:

- Effective in high dimensions.
- Effective in cases where number of dimensions is greater than number of samples.
- Multiple kernel functions can be specified for the decision function.

Weakness:

- SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.
- Sensitive to noise.
- If the number of features is much greater than the number of samples, the method is likely to give poor performances.

Reason for choosing this model:

The data consists of 31 features which is a relatively smaller dimension and the chances of noise being introduced into this dataset is relatively small and SVM works pretty well given these conditions, SVM does pretty well in classifying students who require interventions vs students who don't.

	Training set size		
	100	200	300
Training time (secs)	0.002	0.006	0.012
Prediction time (secs)	0.001	0.002	0.008
F1 score for training set	0.9459	0.888	0.883
F1 score for test set	0.7866	0.789	0.813

Decision Tree Classifier:

General Applications:

- Solving Classification and Regression problems.

Strengths:

- Simple to understand and to interpret. Trees can be visualized.
- The cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree.
- Performs well even if its assumptions are somewhat violated by the true model from which the data were generated.

Weakness:

- Prone to over-fitting due to complex trees.
- Decision trees can be unstable because small variations in the data might result in a completely different tree being generated.
- Decision tree learners create biased trees if some classes dominate. It is therefore recommended to balance the dataset prior to fitting with the decision tree.

Reason for choosing this model:
Given the low complexity of the data, Decision Tree Classifier would be a good candidate to find a solution to the classification problem.

	Training set size		
	100	200	300
Training time (secs)	0.001	0.002	0.009
Prediction time (secs)	0	0	0
F1 score for training set	1.0	1.0	1.0
F1 score for test set	0.753	0.72	0.778

K Nearest Neighbors Classifier:

General Applications:

- Solving Classification and Regression problems.

Strengths:

- Training time is low.
- Robust to noisy training data.
- Effective if training data is large.

Weaknesses:

- Biased by value of k.
- Memory limitations.

Reason for choosing this model:

Low training and prediction time requirements.

	Training set size		
	100	200	300
Training time (secs)	0.001	0.001	0.007
Prediction time (secs)	0.002	0.002	0.003
F1 score for training set	0.911	0.854	0.86
F1 score for test set	0.785	0.7518	0.785

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 recorded to make your case.

Based on the experiments that were performed with 3 different supervised learning classifiers, we came up with the following comparison chart for a training data set size of 300.

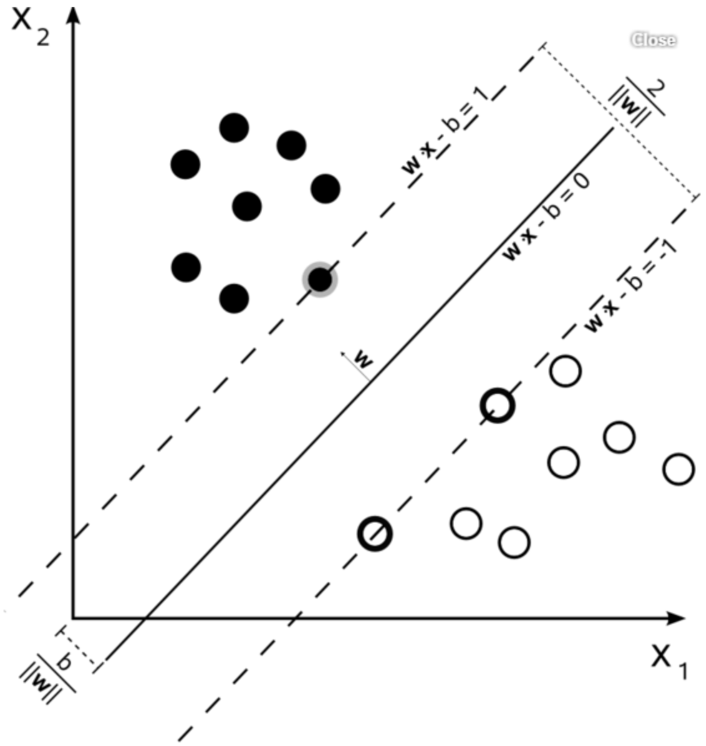
	Training set size: 300		
	Support Vector Machine	Decision Tree Classifier	K Nearest Neighbors
Training time (secs)	0.012	0.009	0.007
Prediction time (secs)	0.008	0	0.003
F1 score for training set	0.883	1.0	0.86
F1 score for test set	0.813	0.778	0.785

As seen from the above table, the Decision Tree Classifier model with default parameters is clearly over-fitting the training data with a perfect F1 score of 1.0. It has a lower F1 score on the test set compared to other 2 models which is a clear result of over-fitting.

K Nearest Neighbors with default parameters on the other hand has the lowest training time and intermediate prediction time compared to the other 2 models. The F1 score on the test set is less than that of SVM but greater than that of Decision Tree Classifier.

The best model from the experiments seems to be the model based on Support Vector Machines. With default parameters, this SVM model has a better F1 score on the test set as compared to the other models with default parameters. The training time and prediction time of SVM is slightly/negligibly higher compared to the other 2 models. Based on the data available, a SVM based model would be most suitable to solve this classification problem.

A Support Vector Machine based model works by creating a linear separation boundary between the data points that represent two different classes as shown below. For purposes of illustration, if our data set comprised of only 2 features X1 and X2, then the job of the SVM model during training would be to find the optimum linear separator which classifies the training correctly into its appropriate classes which in our case is: Does the student require intervention or not. Note that there are an infinite number of lines that will accomplish this task. SVMs, in particular, find the "maximum-margin" line - this is the line "in the middle". Intuitively, this works well because it allows for noise and is most tolerant to mistakes on either side.


X_2
Close
$\mathbf{w}\cdot\mathbf{x} - b = 1$
$\frac{2}{\|\mathbf{w}\|}$
$\mathbf{w}\cdot\mathbf{x} - b = 0$
$\mathbf{w}\cdot\mathbf{x} - b = -1$
$\mathbf{w}$
$\frac{b}{\|\mathbf{w}\|}$
X_1

The entire available dataset is split into a training data set and a test data set. The SVM model is trained on the training data set and the performance of the trained SVM model is evaluated by predicting labels of the test data set.

Using grid search the SVM model was tuned to get a Final F1 score of **0.82.** The final parameters obtained from the GridSearch was able to bring down the prediction time of the test data set in **0.003 seconds.**

Final SVM Parameters:

- Kernel = rbf
- C = 1.0
- Gamma = 0.1

Final Model Stats	
Training time (secs)	0.012
Prediction time (secs)	0.003
F1 score for training set	0.97
F1 score for test set	0.820