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 as the outcome variable is a discreet variable “Passed” with two outcomes - Yes (coded as 1) and No (coded as 0)
If the classification can correctly identify the possible value of outcome variable “Passed”, then the students identified as “Passed=No” are the students who would need intervention.

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 (%age) Number of features Use the code block provided in the template to compute these values.

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

Graduation Rate is the % of people who “passed” which also happens to be our target column. A baseline Predictor could be to always suggest “no intervention”. Such a model on whole data will give us an accuracy of 67.09% which will translate to F1 of

$$F1 = 2 * \frac{1 * 0.67}{1 + 0.67} = 0.803$$

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.

Nothing to report here

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 the 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.

Baseline Predictor for evaluating the improvement of each model: We will use the baseline model of always predicting “No Intervention” which translates to “passed=YES”. The accuracy and F1 for this model is given in table below

	Training Set	Test Set	Full Data
Accuracy	0.683	0.632	0.671
F1 score	0.812	0.774	0.803

F1 score of 0.774 on test set will be compared with each model to assess the improvement.

Model-1 : Logistic Regression

Theoretical-space complexity to represent model: O(m) – where “m” is model complexity in terms of number of weights that need to be storoed. Only few parameters need to be stored.
Theoretical-time complexity to make a prediction: O(m) – prediction is based on simple application of a formula. And the time for calculation depends on the number of parameters or model complexity

Logistic regression is a method of choice given the simplicity of computation and popularity of its use for binary classification tasks. A lot of data is categorical and hence a good model would be to use a maximum likelihood estimator on the conditional probability of output taking a value given the discrete input values.

For the Logistic regression, I used all the features. A better approach would have been to find the correlation between all the features and then use that to reduce the model complexity by throwing away features which show collinearity with other features.

Also for discrete (categorical/numerical) features, the training set should contain enough combinations for the model to learn well. A cursory look at the data did show that but could have been better. For example, assuming 30 features each with say 2 level, we would need 2^30 samples to be able to see all the combinations. However, training test of 300 or less is in opinion inadequate to train the model well.

Infact this is one of the major failings of Logistic regression. In general it requires more data points (training inputs) to provide a stable effective predictor. The advantages though are many fold in terms of the type of data and its underlying distribution. It can handle inputs which could be categorical or continuous. The inputs may follow any type of distribution. There is no assumption being made about the qualities of input variables while performing logistic regression.

Sample Size	Training Time (s)	Testset Prediction Time (s)	F1 Score - Train	F1 Score - Test
100	0.001	0.000	0.859	0.765
200	0.009	0.001	0.856	0.791
300	0.004	0.000	0.847	0.806

The model seems to indicate some improvement on the predictive power with F1 Score on test data being in the range of 0.806, while the baseline predictor on test set has an F1 score of 0.774 (baseline predictor as explained above).

Model-2 : Support Vector Machines

Theoretical-space complexity to represent model: O(m) – only few parameters need to be stored
Theoretical-time complexity to make a prediction: O(m) – prediction is based on simple application of a formula.

Support vector machines provide a robust classification method. Also the reason I picked it was due to the fact that dimensions (number of factors) were 30 while the sample training data is ranging from 100 to 300. SVM works well for highly dimensional data with smaller training data set. I have used the default kernel “rbf” with default value of C and gamma. Also did the scaling of the X_train and X_test to get it centered around zero and deviation to be standard (i.e. value of 1).

Sample Size	Training Time (s)	Testset Prediction Time (s)	F1 Score – Train	F1 Score - Test
100	0.057	0.002	0.927	0.773
200	0.004	0.001	0.896	0.765
300	0.009	0.003	0.903	0.770

SVMs provide a way to maximize the decision boundary thereby ensuring a better separation.

The major advantage of SVM is that it works very well for small data sets. And also when the space of inputs is highly dimensional which is the case for example with 30 features. SVM also benefit from the vary name that support vectors i.e. a few samples from training input are used in decision functions – we need to use only a small number of training points with non-zero alpha (called Support vectors) for classification of a new point. Further, the optimization problem for SVM is a convex one and hence there are no issues of settling down to local maxima.

Also as I used Radial Basis Function (RBF), the was no assumption made about the linearity of the features. The separation boundary can be a non linear one without trying out various degrees of non-linearity of features that would be required in case on logistic regression. The non-linearity is built into the definition of RBF.

The disadvantage of SVM are two a) number of samples must be significantly larger than the number of features and b) SVMs do not directly support probability estimates. Both these are not an issue in this case as we are using SVM as a classifier and even with the small sample size, 300 points vs 30 feature provide a good ratio to address point (a) above.

SVMs also suffer from the lack of transparency of results. Given the mapping of input space to an infinite dimensional space in RBF, it is not intuitive to explain the outcome of an RBF.

I used the default value of C (C=1) and gamma (auto i.e. 1/n_features = 1/30 in our case). The result can be improved by using gridSearch on C and gamma. C is a measure of in-sample classification error. Higher the C, lower is the generalization. While a low C, higher is the generalization due to wider margin. This can lead to a higher in-sample classification error. Similarly, gamma controls the local effect of points in training sample. A low gamma gives a sharp and narrow peak at each training point there by a lower bias and higher variance which in turn could lead to overfitting. Opposite is true for higher gamma wherein you get flatter peaks leading to smother decision boundaries. Various cross-validation strategies could be used to find the optimal values of gamma.

Model-3 : Random Forest

Theoretical-space complexity to represent model: O(ntree * n log(n)) – Multiple trees have to be stored – each with space complexity of n log(n)
Theoretical-time complexity to make a prediction: O(ntree * d log(d)) – where “d” is the depth of tree. If “d” is not known then we can substitute it by “n” – no. of samples

Random forest is a good classification model which uses bagging to generate multiple CART trees and then at the time of prediction combines the prediction of each CART tree to produce a final classification outcome. A single decision tree may suffer from high variance or high bias based on how they are tuned. Random forests uses an average across multiple CART models to find a balance between the two extremes of high bias and high variance. They are part of ensemble learning method.

The boot strapping method usually leads to better performance as it decreases variance without increasing bias. This is based on assumption that the ensembles trees are not correlated. Bagging helps in keeping individual trees from being correlated.

Like decision tree, the advantage of Random forests is easy interpretability.

The main downside is the cost of producing and storing tree. As model needs to create multiple trees and each such tree needs to be stored, it leads to expensive storage.

Also at the time of prediction, the input data needs to be compared against multiple trees, the prediction time is also high as compared to above two models – Logistic Regression and SVM.

I decided to use the random forest on the given data, as the size of data is limited and wanted to reduce the chances of decision tree getting into extreme situation of high bias or high variance.

Sample Size	Training Time (s)	Testset Prediction Time (s)	F1 Score - Train	F1 Score - Test
100	0.008	0.001	0.992	0.760
200	0.008	0.001	0.993	0.785
300	0.009	0.001	0.993	0.809

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 chose as the best model. Which model is generally the most appropriate based on the available data, limited resources, cost, and performance?

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 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?

The F1 score of three models as reported above are experimental F1 score from a single run. The experimental F1 score will fluctuate around the theoretical F1 score. Once we allow for this variation, we see that F1 scores of the three models, as reported above, do not give any indication about the relative superiority of any one of the three models. Though in the data above, logistic regression offers a better F1 score (0.806 for n=300) over the other two models (0.770 & 0.809 for n=300), I would conclude that that F1 score offered by all the three models on test set is fairly close to each other.

As the three models offer about the same F1 score, I would recommend to use the model which is most efficient in terms of limited resources and lowest cost of processing power required to perform predictions.

Keeping these constraints in mind, I would recommend "Logistic regression". Logistic Regression is an efficient model both in terms of storage and processing required to predict outputs. A logistic model is completed defined by 31 numbers (number of features + 1), so it is very efficient from storage stand point.

Even predictions using logistic regression require computation of a very simple function based on input values and model numbers. Roughly the calculation involved in making a single prediction is the order of 30 additions and 30 multiplications followed by one exponent which is very fast and hardly consumes any power.

The advantage of "Logistic Regression" is also that it is one of the most inexpensive models to train. So as the data volume grows, retraining the model will be inexpensive. Further, the storage of model and prediction on new inputs do not depend on the size of input which makes it very scalable in future.

Before making these recommendations, I carried out a fairly extensive experiment to try out various models and comparing the performance and effectiveness of several options. I tried a modelling technique called "Grid Search" to find optimal prediction model for two types of models – "Logistic Regression Model" and "Support Vector Machines".

"Logistic Regression", the model being recommended, is very similar to the intuitive model of regression for output estimation except that Logistic regression is suitable for a problem like ours wherein we need to predict the need of intervention. The model works by estimating the relative importance of each input parameter. The final model is a set of 30 numbers, one numerical number for each input parameter which signifies the relative importance of that parameter.

The top 5 features as identified by the "Logistic model" are:

1. "number of past failures": Higher is the number of past failure, higher is the chance that student will not pass the final exam.
2. "number of times student goes out with friends" : again higher is the number of times student goes out, higher is the chance that student will fail the final exam.
3. If a student has family support for education, he is more likely to fail in final exam. This sounds counter intuitive and may signify some other factor which is in play.
4. A student wanting to take higher education is more likely to pass the final exam. This probably stems from the fact that he is a more committed student
5. The presence of taking extra paid classes increases the chances of a student passing the final exam.

Some features play a significant role in helping predict the outcome student passing the final exam, while some other features play a lesser role. The model uses the value of these five features (as well as the other remaining 25 features) to calculate the odds of student passing vs failing the exam.

<u>F1 Score of Final Model</u>
The model of choice – Logistic Regression has a final F1 score on test set of 0.786 offering a marginal improvement over baseline F1 score of 0.774. The baseline predictor is to say "No Intervention" always. The F1 score of this baseline model is explained in the beginning of Section 4.

I also tried running the notebook multiple times with different values of "random_state" while splitting the data in training and testing set. Each time, the performance of three models was not too far and also the F1 score of "Logistic Regression" provided a marginal improvement over the baseline model. My hypothesis though is that accuracy and F1 score of all three models (and esp the proposed model of “Logistic Regression”) will keep improving with larger data set.