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 there is a discrete number of outcomes/labels rather an a continuous value.
- In this case it is binary classification problem as there are 2 possible outputs labels, pass and fail

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**

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
Number of features: 31
Graduation rate of the class: 0.67%
```

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 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 - K Nearest Neighbours (KNN)

KNN can be used for regression and classification. In this project I am using it for classification. I chose to use this classifier as it is simple, well understood and fast. I am using the default N of 5 as the training set is of sufficient size to allow for this.

Pros

- Training time is very low and constant for increasing samples
- Prediction time is fast

Cons

- Prediction time increases with an increasing number of neighbours
- Doesn't work well with a large number of features
- Space requirements increase exponentially with the number of features

	N Samples	Train Time	Pred Time Train	F1 Train	Pred Time Test	F1 Test
KNeighborsClassifier	100	0.0006	0.0016	0.8477	0.0014	0.7368
KNeighborsClassifier	200	0.0009	0.0035	0.8600	0.0021	0.7914
KNeighborsClassifier	300	0.0011	0.0075	0.8598	0.0029	0.7761

Model 2 - Support Vector Machines (SVM)

An SVM model is used for classification. I chose it as it works well out of the box and also is good for a large number of features

Pros

- Works well with lots of features
- Works well on large training sets

Cons

- Training time is very slow and increases exponentially for increasing samples

	N Samples	Train Time	Pred Time Train	F1 Train	Pred Time Test	F1 Test
SVC	100	0.0016	0.0011	0.8553	0.0010	0.8153
SVC	200	0.0041	0.0032	0.8813	0.0016	0.7867
SVC	300	0.0083	0.0064	0.8811	0.0024	0.8027

Model 3 - Decision Tree

A decision tree can be used for classification and regression. In this project I will be using it for classification. I chose it as it is easy to understand, fast and is suited for binary classification problems.

Pros

- Fast to train
- Fast to predict
- Requires very little space

Cons

- Will overfit with small sample sets

	N Samples	Train Time	Pred Time Train	F1 Train	Pred Time Test	F1 Test
DecisionTreeClassifier	100	0.0009	0.0003	1.0000	0.0003	0.7442

DecisionTreeClassifier	200	0.0015	0.0005	1.0000	0.0004	0.6126
DecisionTreeClassifier	300	0.0023	0.0008	1.0000	0.0003	0.7402

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?

In 1-2 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?

Recommendation

After reviewing 3 different classifiers, KNN, Decision Tree and SVM with the student data I have taken into account training time, prediction accuracy and space required.

The KNN model worked well with the average f1 test score of 0.7681 on the testing data, placing it in the middle with the Decision Tree average F1 at (0.6990) and the SVM average F1 (0.8016). The KNN suffers from the curse of dimensionality which meach that for each feature there is a significant increase in resource required for computing and space, and due to this I do not recommend using this model.

The SVM was shown to be the most accurate out of the box with an average F1 test of 0.8016. While this is a good thing, the amount of time to train the model is significant. It took approx 4 times as long to train on 300 samples than the Decision Tree and 8x as long as the KNN classifiers. Due this being resource heavy I do not recommend using this classifier.

The last model to review was a Decision Tree. It's average F1 accuracy was the lowest of the classifiers at 0.6990, which I believe to be caused by bad out of the box parameters. Parameter tuning can fix this. As for the resources required, the classifier was very quick to train and is incredibly quick to predict. It also requires very little storage space. Another benefit is that this classifier will allow us to extract vital information about the data we are reviewing, such as a list of the most important features relating to a pass/fail grade. Due to the above I select this classifier to parameter tune.

Description: Prediction with a decision tree

A decision tree is a very simple classifier to understand. It is essentially a tree where each node is a binary question and the leaf is outcome. For example in relation to the supplied data, the classifier may have derived the first and most important question to be "Does the student have the internet at home?", and depending on the outcome it will ask the next most important question along that branch until it reaches a leaf, which contains the pass or fail label.

Final F1 Score

After parameter tuning the decision tree classifier with GridSearchCv using 3 folds, I received an F1 score of 0.8110, which is better than all the previous test scores.

I tuned 2 parameters, max_depth from 1-5 and min_samples_split from 1-15.

GridSearchCv selected the best parameters to be min_samples_split of 1 and max_depth of 1.

Supplementary

After parameter tuning the grid search choose the best max_depth of 1, which indicates that there is 1 feature that stands out as the major influence of a pass or fail grade. We can also use the classifier as a tool to extract this information. (Code supplied in notebook)

Feature Importance max_depth 1
1.000 - failures

Feature Importance max_depth 2
0.717 - failures
0.142 - schoolsup
0.141 - absences

Feature Importance max_depth 3
0.497 - failures
0.196 - absences
0.138 - schoolsup
0.097 - goout
0.071 - studytime

So in summary the school should focus on the students who have failed before, as this is the key indicator whether the student will pass or not.