COMP2231: Software Methodologies Machine Learning Coursework

gpqk41

Word count: 1,087

1 Introduction

This report investigates the use of two supervised Machine Learning methods to carry out learning analytics on the OULAD dataset and predict a student's final result. The methods chosen are Support Vector Machine Classification and Random Forest Classification. The aim is to determine the ability of these models to predict whether a student will dropout during a course. This is arguably the most insightful indicator for course leaders, facilitating early intervention and targeted guidance for struggling students.

2 Data Cleansing

The data provided was cleansed to remove missing values and correct semantic inconsistencies. Figure 1 shows how the vle dataset contained two columns that were only 17.6% populated; these columns were dropped. The studentInfo dataset also contained incomplete columns; the *imd_band* column was missing 1111 entries. These values were imputed as zero for convenience. This dataset also contained inconsistencies between nine students' *final_result* and the presence of a *date_unregistration* value. Within the assessments dataset, the combined weights of assessment by type for modules CCC and GGG did not sum to 200% as detailed in the data description. This was corrected by altering/imputing the weights for the relevant assessments.

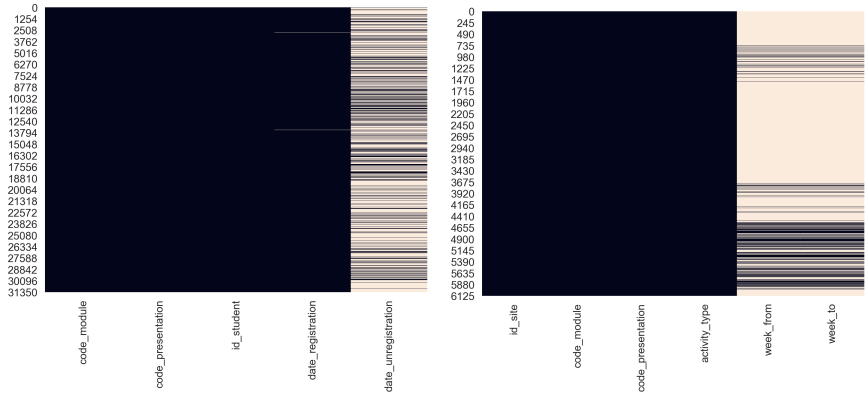


Figure 1: vle and studentRegistration heatmaps

3 Feature Engineering

The studentAssessment dataset was not particularly informative so it was merged with the assessments dataset to calculate each student's average mark by assessment type by module. This yielded two new features: *avg_CMA* and *avg_TMA*. Figure 2 illustrates the relationship between each of these variables and *result_class*.

Furthermore, the studentVle dataset required significant manipulation to produce data about each student. It was merged with the vle dataset to generate forty additional features which represented the total number of interactions each student had with each type of activity (*abc_sum_click*) over how many unique days (*abc_count*). Finally, the engineered features were added to the cleansed studentInfo dataset.

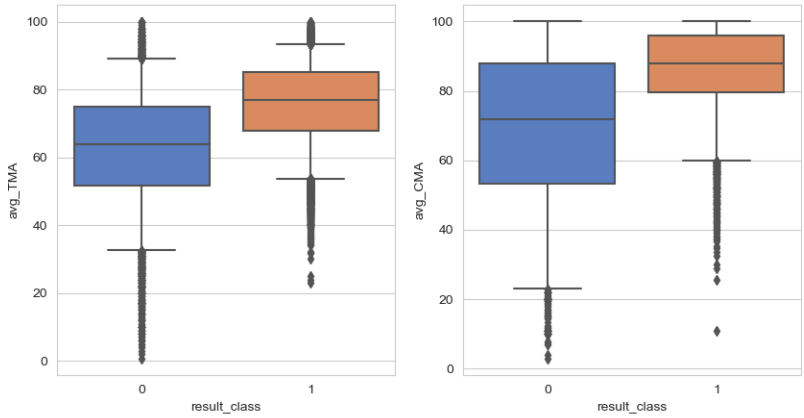


Figure 2: Distribution of CMA and TMA marks by result_class

4 Data Preparation

Predicting whether a student will fail/drop-out of a course necessitated reframing the task as a binary classification problem. This was achieved by generating a *result_class* feature which was set to one when a student's *final_result* was 'Pass' or 'Distinction', and zero otherwise.

Since the implementation of the chosen models require strictly numerical data, it was necessary to encode certain columns using a ColumnTransformer. Ordinal encoding was used for ordinal features (e.g. *imd_band*) and one-hot encoding for categorical features (e.g. *region*). After transformation, the data was normalised using the StandardScaler. This was done to allow the RFC's feature importance measures to be used reliably despite the features varying in scale or number of categories. Both steps were added to pipelines for each model, improving the readability of the workflow and enforcing each step on both the training and test sets.

5 Models

5.1 Support Vector Machine Classification (SVC)

The SVC method works by constructing a hyperplane in multidimensional space that best separates the input data into two classes. The algorithm seeks to select a hyperplane with the largest margin between support vectors (the points close to the hyperplane). As the prepared data was not linearly separable, the Radial Basis Function kernel was used which transformed the data into a higher dimensional space.

5.2 Random Forest Classification (RFC)

RFC is an example of Ensemble learning. It involves training a group of Decision Tree classifiers (each on a random subset of the training set) and making predictions based on the most popular classification across the cohort. Each tree within the forest only considers a subset of the features, reducing the training time of the model.

6 Experimental Procedure

The prepared data was split into features (X) and labels (y), which were then divided into training and test sets using an 80:20 split, allowing the performance of the models to be evaluated after training. The performance of the models was determined by measuring the precision, recall and accuracy when predicting the labels of the test inputs. Recall is particularly important because it measures the ability of the model to detect all cases of a class; a model with a greater recall would be more effective at completing this task. The precision and accuracy of the models also indicated its performance compared to other models. Initially, both models were

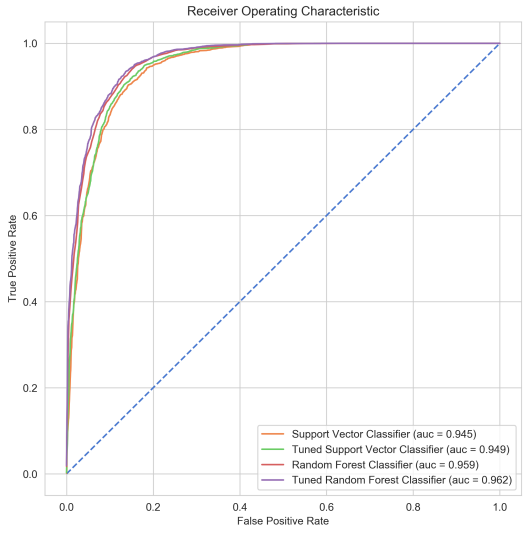
run using the default options on the prepared data. Then, their hyperparameters were tuned using Grid-Search Cross-Validation.

7 Performance Comparison

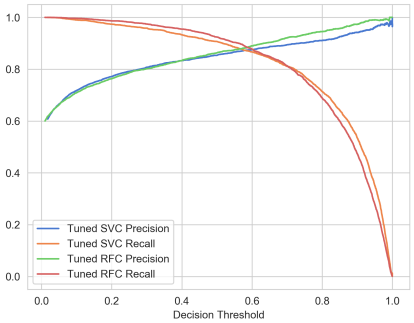
Table 1 shows the performance of each model before and after tuning. Although their performances were comparable, the tuned RFC performed marginally better according to the metrics used. Figure 3a also illustrates the relative performances of both models before and after tuning using an ROC curve. Again, the tuned RFC had the greatest AUC value which quantifies how well separated the probabilities from the positive class are from the negative class. Figure 3b illustrates the precision/recall trade-off which explains the values in Table 1.

Model	Hyperparameters	Recall	Precision	Accuracy
SVM	-	0.88	0.88	0.8780
Tuned SVM	gamma=0.01, C=10	0.88	0.88	0.8774
RFC	-	0.89	0.88	0.8842
Tuned RFC	n_estimators=800 min_samples_split=2 min_samples_leaf=2 max_features="sqrt" max_depth=80 bootstrap=False	0.90	0.89	0.8946

Table 1: Performance of each model



(a) ROC Curve for each model compared to a random predictor



(b) Precision/recall trade-off for tuned models

Figure 3: Quantitative measures of model performance

The time complexity of an SVC is between $O(n^2m)$ and $O(n^3m)$, where n and m represent the number of samples and features respectively, resulting in extremely slow training times for datasets with many instances. The training time for an RFC is $O(n^2\sqrt{m}.n_{trees})$, meaning training is faster. Moreover, the performance of an SVM is sensitive to the selection of the regularisation (C) parameter, which can cause over-fitting/selection bias

during tuning. An RFC does not suffer from over-fitting if the number of trees is large enough although this does introduce a time-performance trade-off. Additionally, the RFC is more well suited to slightly unbalanced data.

The RFC facilitates the extraction of feature importances. Figure 4 plots the relative importance of all features of the dataset. The three highest bars indicate that the model considered *avg_TMA*, *homepage_count* and *quiz_count* to be the most important. This indicates that vle/assessment data is far more useful than demographic data when predicting a student's result.

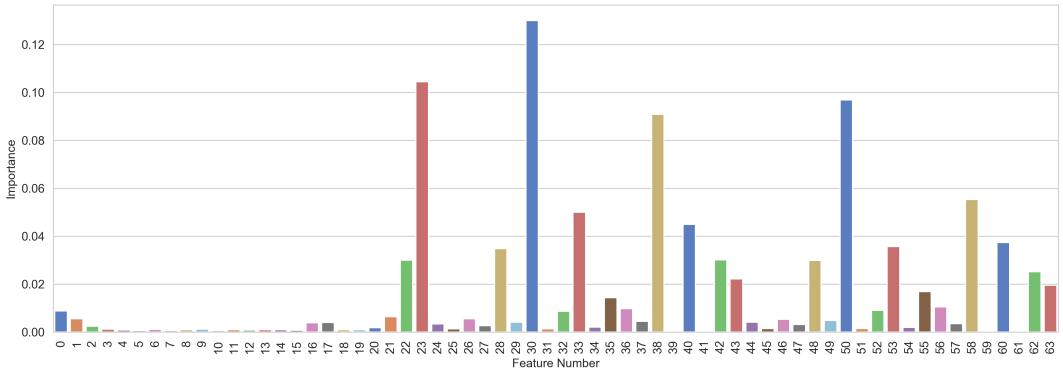


Figure 4: Feature importances from the RFC

8 Conclusion

This report aims to investigate the effectiveness of SVCs and RFCs at predicting a student's result. The tuned RFC achieved a recall of the negative class of 0.8946; this means the model identified 89.5% of all students due to fail prior to the exam. This model would allow academic staff to offer additional support to these students earlier, demonstrating the suitability of Machine Learning models for academic classification tasks. We can also conclude that, perhaps unintuitively, demographic data is not indicative of a student's performance.