Learning Analytics Using OULAD Data

1.0 Data Preparation

An understanding of the data was gained by examining the database schema**[1]** and creating a data dictionary:

name	type	example_value
code_module	object	AAA
code_presentation	object	2013J
id_student	int64	11391
gender	object	M
region	object	East Anglian Region
highest_education	object	HE Qualification
imd_band	object	90-100%
age_band	object	55<=
num_of_prev_attempts	int64	0
studied_credits	int64	240
disability	object	N
final_result	object	Pass
date_registration	float64	159
influence	float64	1648
sum_click	float64	4.76531

Figure 1.0 – data dictionary

Final exam results were then removed from the assessment data as this would make it possible to calculate each student's final grade exactly, removing the need for estimation. Also, students who withdrew from the course were removed from the data since their data would not help predict a final result.
StudentInfo was identified as the table containing the most relevant data to the students and new features were created for the mean number of clicks a student made on the VLE as well as the weighted average of each student's score on all other assessments. Additionally, the feature date_registration was merged into StudentInfo.
To visualise the final_result feature I then plotted a chart depicting the percentage of records with each value:

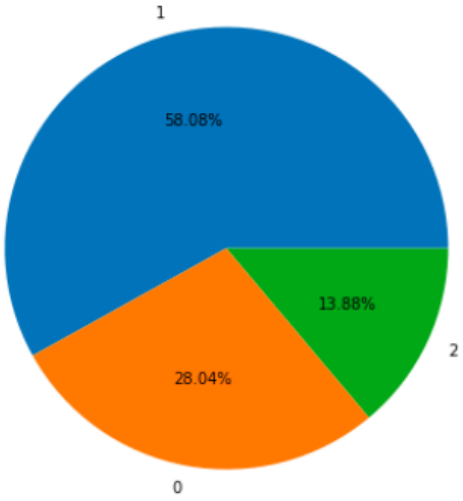

Figure 2.0 – Percentage of samples with each value for final_result

With only 13.88% achieving a distinction, predicting a pass/fail result was decided due to the limited data on distinctions.
The few samples missing data for some features were removed from the data set as opposed to filling the gaps with an average due to their high range.
A mapping dictionary was then used to convert ordinal categorised data into numerical representation meanwhile nominal categorised data was replaced using One-Hot encoding. This made all data in the table numerical at the cost of adding many new features.
'Influence' was the feature with the highest correlation to 'final_result' and with the highest range of any feature, it was an excellent candidate for stratified sampling.

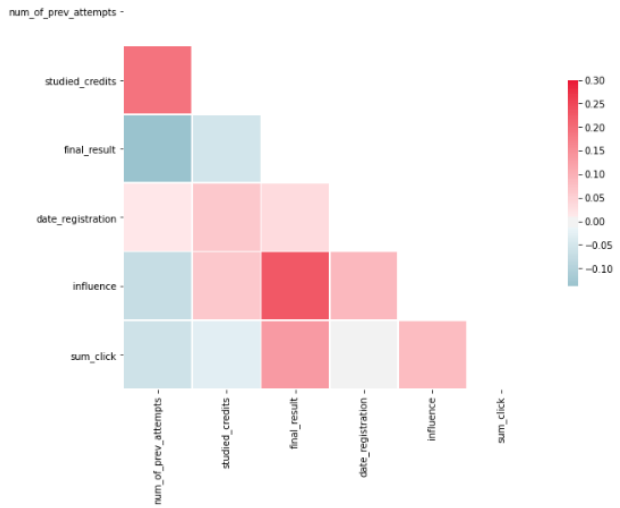

Figure 3.0 – Correlation matrix

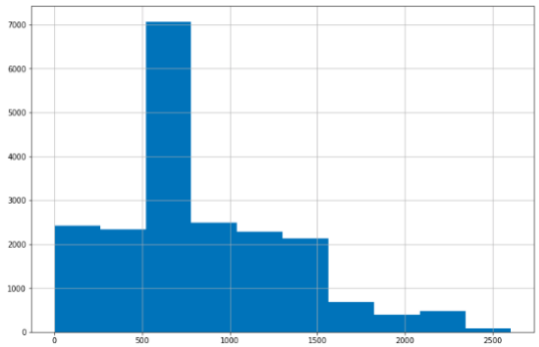

Figure 4.0 – Histogram of values for influence

Using the above histogram, influence was split into categories from which the stratified split could be achieved.

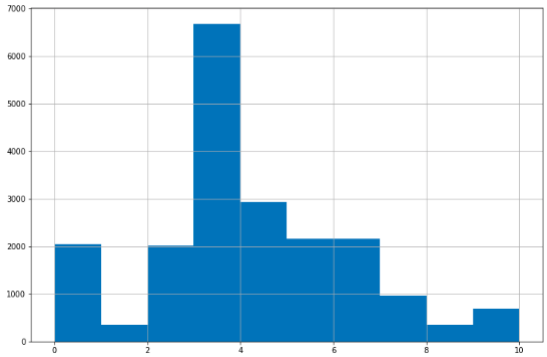

Figure 5.0 – Histogram of categories for influence

Finally, a minmax scaler was fit on the training data and used to transform both data sets to make the range of each feature 1.0.
The data cleaning phase then concludes with stratified, scaled datasets for testing and training.

2.0 Training and Evaluation

To explore how different models naturally fit the data, both binary classifiers and various regressors were tested to find root mean squared error (RMSE) and accuracy score as together these describe a models margin of error as well as how often it is incorrect. These metrics were taken again after cross-validating each model on five splits.

Model	Initial Score	Initial RMSE	Cross-Val Score	Cross-Val RMSE
Linear Regressor	0.174	0.406	0.155	0.171
SGD Classifier	0.767	0.481	0.763	0.237
Random Forest Classifier	0.822	0.422	0.811	0.189
SVC	0.779	0.470	0.764	0.236
Linear SVC	0.778	0.471	0.768	0.232
Logistic Regressor	0.778	0.471	0.770	0.230
Decision Tree Regressor	-0.271	0.504	-0.277	0.258
Decision Tree Classifier	0.746	0.504	0.744	0.259
Ridge Classifier	0.775	0.475	0.767	0.233

Figure 6.0 – Preliminary vs cross-validated results for tested models

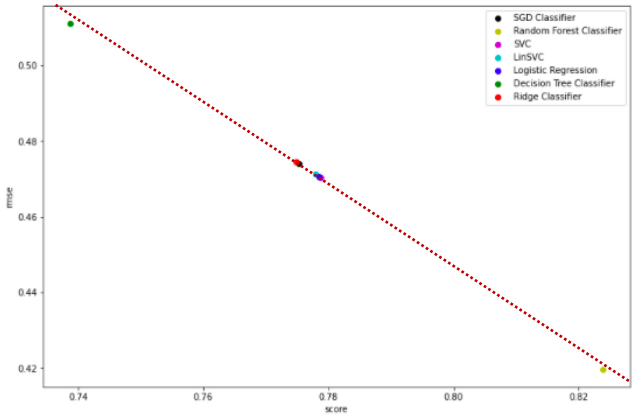

Figure 7.0 – RMSE vs score graph for models

When plotted, the RMSE and score metrics formed a trend of negative correlation with a lower accuracy inferring to a higher mean error. Cross-validation produced similar results to initial tests meaning the models were not overfitting and performing similarly on various data sets. The Logistic Regressor (LR) and Random Forest Classifier (RFC) models continued into the tuning and evaluation stage due to their scores being the highest when cross-validating.

3.0 Parameter Search and Selection

The chosen models were then tuned so as to find the combination of hyperparameters best suiting the data.

Grid search was chosen over random search and manual tuning as it provided a more time-efficient solution than manual tuning but would provide a more exhaustive solution to random search.

Two dictionaries were created consisting of possible values for the parameters of LogisticRegression and RandomForestClassifier respectively. These dictionaries were then passed into two, 5-fold grid search algorithms and fitted to the training data. Considering the multiple parameters, values and the grid search having a 'cv' of 5, in total, the RFC and LR were fitted 1,080 and 120 times respectively.

The best estimator found from each grid search was refitted to the training data so that they could be compared to the untuned models.

4.0 Comparison of Methods

Having tuned and refitted the selected models, they were then evaluated. Adhering to the metrics used for preliminary tests, the RMSE and accuracy score were calculated through cross-validation against the test set created earlier.

Model	Initial Score	Initial RMSE	Tuned Score	Tuned RMSE
2-Class Logistic Regressor	0.7698	0.2302	0.7711	0.2289
2-Class Random Forest Classifier	0.8109	0.1891	0.8111	0.1888

Figure 8.0 – Comparison of cross-validated metrics pre and post tuning

As shown, the increase in performance from tuning the hyperparameters was minimal indicating that the default parameters are well-suited to the data.

As a graphical representation of each models' performance, a confusion matrix and ROC curve were then generated, comparing the models' predictions to the actual results:

Random Forest Classifier	Actual True	Actual False
Predicted True	587	535
Predicted False	226	2724

Recall: 0.722
Precision: 0.523
Accuracy: 0.813
F-Measure: 0.607

Logistic Regressor	Actual True	Actual False
Predicted True	322	800
Predicted False	124	2826

Recall: 0.722
Precision: 0.287
Accuracy: 0.773
F-Measure: 0.411

Figure 9.0 – Confusion matrices for LR and RTC models

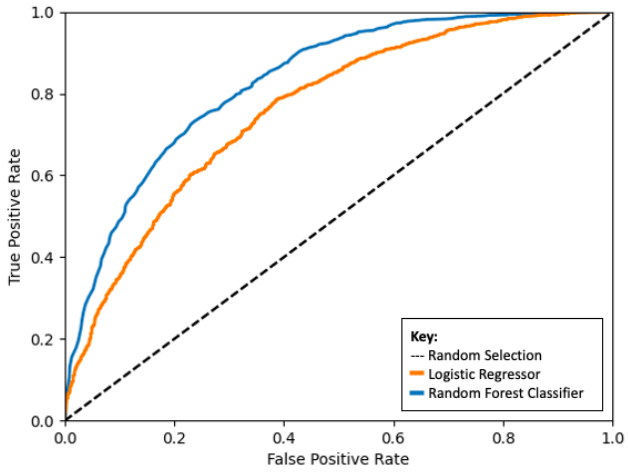

Figure 10.0 – ROC curves for LR and RTC models

5.0 Conclusions

In conclusion, the RFC was shown in both RMSE and accuracy score metrics to be superior to the LR model. This difference was further demonstrated in the confusion matrices and ROC curves for the two models, with the ROC curve for RFC being far closer to the upper-left corner of the graph (the ideal classifier). However, the LR model is still well-suited to this data and provides a far better-than-random performance as seen by the ROC curve curving away from the random selection line. The performance of these models has been acceptable given the removal of final exam scores from the dataset, with final

accuracies above 75%. Unsurprisingly, the best predictor of final_result was the weighted mean of assessment scores; however, the second best was the number of previous attempts made by the student. To improve, samples removed for containing null values could be replaced with averages while ensemble methods could be used in place of grid search hyperparameter tuning.

6.0 References

[1]
https://analyse.kmi.open.ac.uk/open_dataset