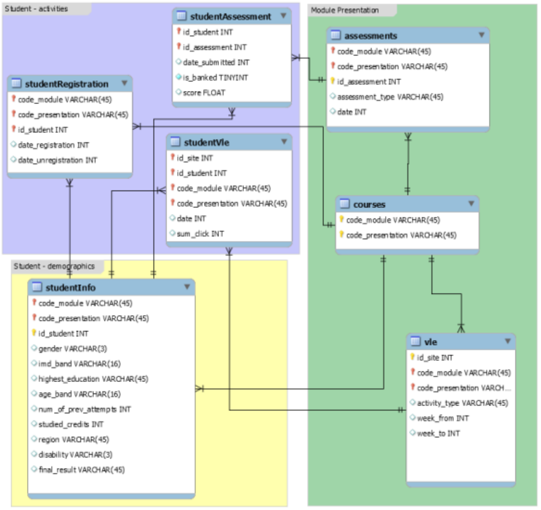
Machine Learning

Predicting Final Results

I have undertaken an investigation upon the use of two different machine learning algorithms, Random Forest Classification (RFC) and Logistic Regression Classification (LFC), to predict the final result that a student may obtain. The report will go in detail on how the data was prepared along with the experimental procedures used to obtain the results I have received. A comparison of the two machine learning algorithms will also be made which will be further discussed.

Data Preparation



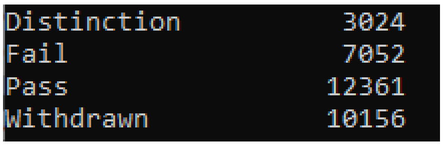
Shown above is the schema diagram of the OULDAD (Open University Learning Analytics Dataset) that was used for this investigation. Not all data sets were used as I had deemed some to be not necessary so that it would not flood the training models and cause false negatives.

Three datasets were used which were “studentInfo.csv”, “studentAssessment.csv” and “assessments.csv”. To prepare the data, a copy of “studentInfo.csv” was made with the “student_id” column removed but with the addition of “average_score”. The removal of “id_ student” was made as I had judged that it would have no impact when predicting the results. Whereas, the addition of “average_score” was done by calculating an average score using the values formed from the merging of “assessment.csv” and “studentAssessment.csv”. These two data sets were merged according to “id_assessment” and matched with “id_student” and “code_module”. Additional columns that had appeared from merging were dropped to keep the data set concise.

The next stage was to ensure that null values were taken care of. It was found that column “imd_band” was the only one that contained null values. To reduce the difficulty any rows that contained null values were removed from the data set completely.

Following from this, we had to ensure that there was no categorical data present within the data set. To combat this, we used One Hot Encodings to make categorical data into numerical values which in turn helps predictions become as accurate as possible.

Finally, we needed our label column, “final_result” to be a binary classification to guarantee that the data set can be trained using the methods described below. The first step in doing so was to eliminate two types of results. Upon further calculation, it was found that “Distinction” and “Fail” were the minority out of the four result types, therefore rows containing those values were discarded. “Pass” and “Withdraw” values were then changed to be 1 or 0 respectively.



```
Distinction        3024
Fail               7052
Pass              12361
Withdrawn         10156
```

Figure 1

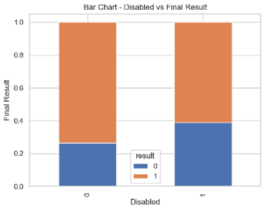


Figure 2

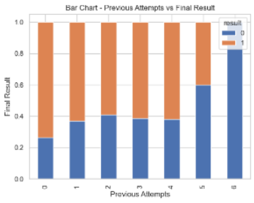


Figure 3

Logistic Regression Classification (LRC)

This model was used as the data that we are predicting had a binary classification of pass/withdraw therefore making it a good model to use. The LRC classifies data through a binary dependant variable using a logistic function. The benefit of this is that increasing one of the independent variables multiplicatively scales the odds of the given outcome.

The data was split into 20% and 80% for test and training data respectively. The maximum iteration parameter was set to 10000 and the model was fitted on the training data. The test data was used to make predictions and can be viewed in the experimental procedure section of the report.

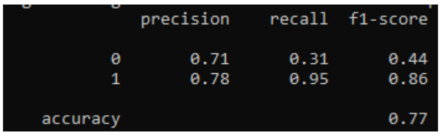
Random Forest Classification (RFC)

RFC works best when the predictions made by the individual trees have low correlations with each other. This model was used to compare how this simple algorithm works with multiple features and randomisation as opposed to LRC.

Similarly, to the LRC, the data was split into 20% and 80% for test and training sets, respectively. The model was instanced with 140 n_estimators and a maximum depth of five to allow more samples to be randomised and taken. Bootstrap was enabled to ensure the samples taken were at random increasing the precision of the predictions.

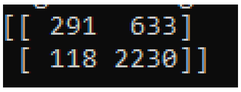
Experimental Procedure

To gauge the performance of the two methods, I had decided to compare the values provided by the classification report. The closer to 1 the values were the better the performance of the predictions.



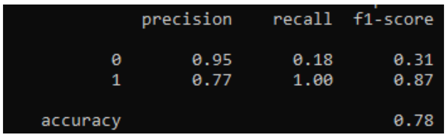
```
              precision    recall  f1-score

           0       0.71      0.31      0.44
           1       0.78      0.95      0.86

    accuracy                           0.77
```

Figure 4 – LRC



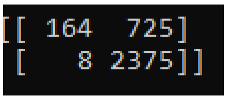
```
[[ 291  633]
 [ 118 2230]]
```

Figure 5 – Confusion Matrix for LRC



```
              precision    recall  f1-score

           0       0.95      0.18      0.31
           1       0.77      1.00      0.87

    accuracy                           0.78
```

Figure 6 - RFC



```
[[ 164  725]
 [   8 2375]]
```

Figure 6 - Confusion Matrix for RFC

In order to further test the models, I had experimented with changing the values for the random state on the LRC. It had a small impact on the values given within the classification report.

For the RFC, I had experimented with changing the depth of the tree and it also had small changes to the result found in the classification report however when changing the n_estimators, there was little to no change when looking at precision and recall.

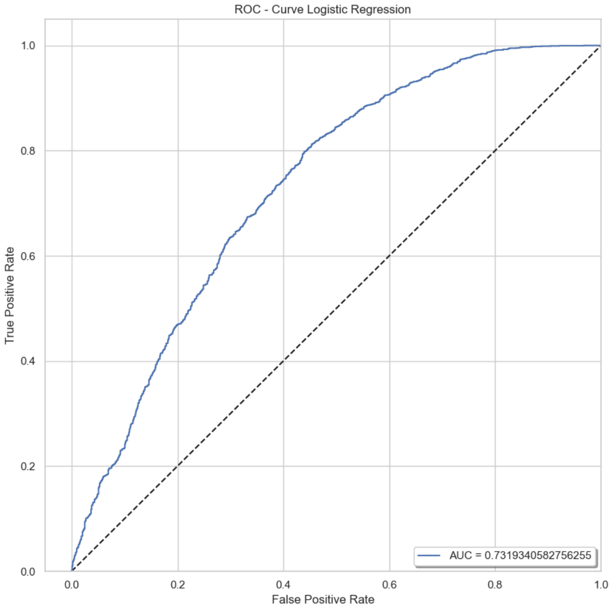


Figure 8

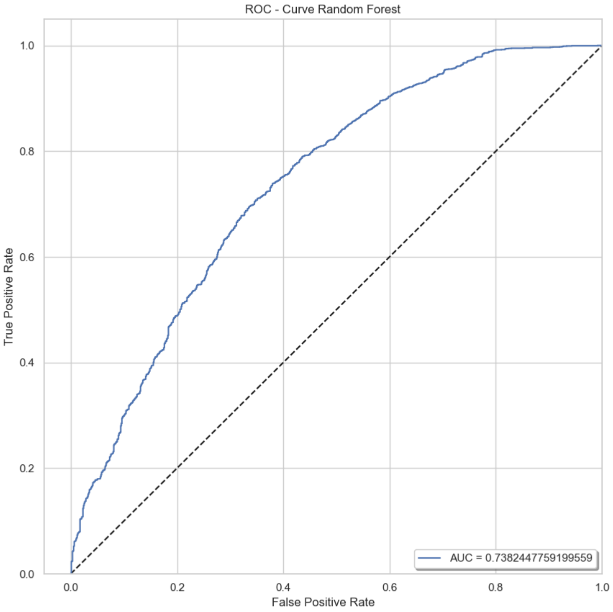


Figure 9

Analysis and Conclusion

There were quite poor results for recalling withdrawn when using both classification techniques to make predictions. This could be an indicator that there were many false positives lying within the data sets when being prepared. To combat this, I could have removed columns or added extra columns from data sets that I had not used such as the "studentVle.csv" or "Vle.csv". However, in terms of precision and accuracy both models were hitting similar numbers and quite high numbers as well if we do not take recall into account. This can be seen within figures 4 to 7.

In RFC, many techniques are used to enhance prediction such as bagging and random sampling. This meant that small changes to the training set could result in significant different tree structures. But random forest allows each individual tree to randomly sample with replacement causing little change in predictions when features are tampered with. While with LRC struggles to deal with categorical data and excels more with binary classifications.

The rules used by these two models are entirely different, RFC separates data repeatedly into subsets that achieve the best result where as LRC calculates the odds using logistics and determines the maximum likelihood that the prediction is correct. Even though these techniques were vastly different, I still wanted to try out both techniques and see the results first-hand as to which is greater.

In conclusion, after adjusting parameters and features to be used in the data set, the overall performance of the two models were similar. Since I had made it so that every feature was a binary classification, LRC performed slightly better on recall as opposed to RFC.