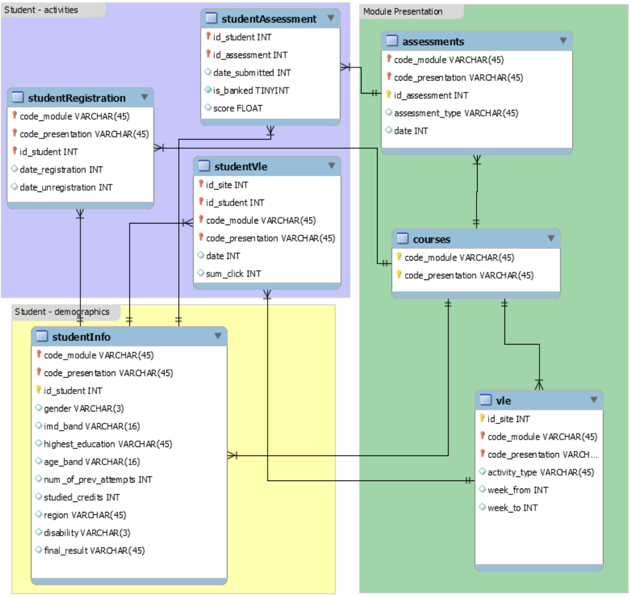
The Data Gathering



Shown left is the schema diagram of the OULAD (Open University Learning Analytics dataset) database used for this report. The information, deemed pertinent to create a decent model, was gathered across the files "studentInfo.csv", "studentAssessment.csv", "assessments.csv" and "studentVle.csv". This information was then saved in a new file "studentInfoEdited.csv".

This file was a copy of "studentInfo.csv" with no "id_student" column and two additional columns: "Total_no_of_material_use", representing the number of times material for a module was accessed by a student, and "Average_Assesment_Score", self explanatory.

"Average_Assesment_Score", for a row in "studentInfo.csv" was calculated from averaging "score" values in the rows of a panda dataframe, formed from merging "assessments.csv" and "studentAssessment.csv" according to "id_assessment", where all rows matched the "id_student", "code_module" and "code_presentation" of the row in "studentInfo.csv".

"Total_no_of_material_use" was found the same way, except for the panda dataframe being "studentVle.csv" and the calculation being the sum of "sum_click" for the rows.

Creating "studentInfoEdited.csv" took 15 minutes on the computer used for this report and so to save time this process will only run in "classifier.py" when "studentInfoEdited.csv" isn't a file in the "Data" folder.

The Data Preparation

The first stage of data preparation was to deal with any null values. The only column to contain them was "imd_band" and in only 1,111 rows. Since prediction of these missing values was too difficult and the number of them so small, compared to the total 32593 rows, they were just deleted from the data.

Stage two was to replace columns that were string values with One Hot Encodings. The columns were: 'code_module', 'code_presentation', 'region', 'highest_education', 'imd_band', 'age_band', 'gender' and 'disability'.

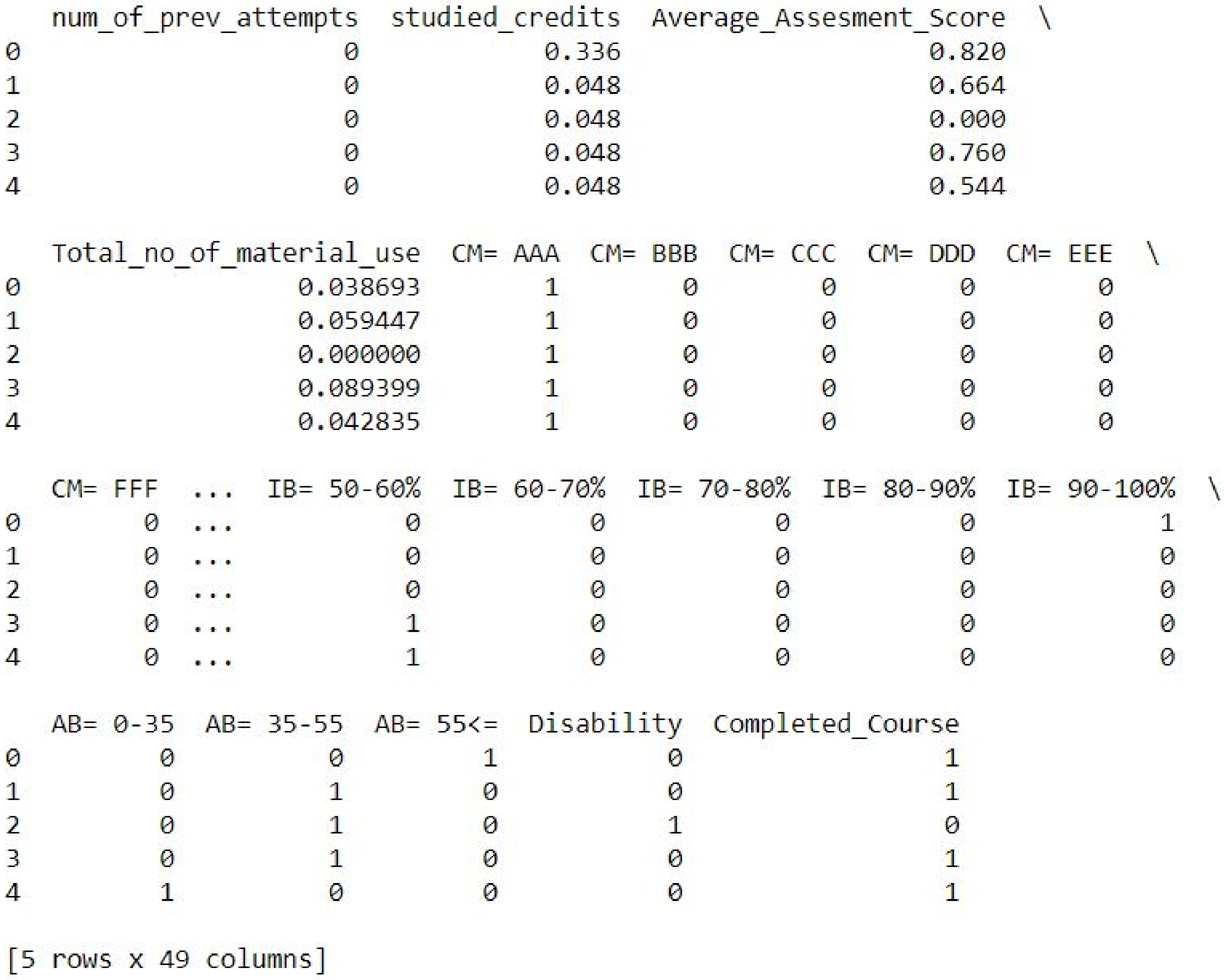
Columns 'gender' and 'disability', remained single columns but their information became binary. 'disability' either stayed the same name or was changed to 'No Disability'. 'gender' became either 'Male' or 'Female'. Both name outcomes depend on how a system handles encoding.

Stage three was to normalise columns that had large ranges. These were 'Total_no_of_material_use', 'studied_credits' and 'Average_Assesment_Score'.

Finally, 'final_result' was changed and renamed 'Completed_Course'. This was to make grade prediction a binary classification, for reasons discussed in more detail in section three. The four categories were redefined. 'Distinction' and 'Pass' became 1's to show completion of the course, and 'Withdrawn' and 'Fail' became zeros to show incompletion.

The Models Chosen

The prediction models used were Logistic Regression and Decision Tree. The reason was that predicting final grades could be easily framed as a binary classification, and both models work very well for this.



```
   num_of_prev_attempts  studied_credits  Average_Assesment_Score  \
0                     0            0.336                    0.820
1                     0            0.048                    0.664
2                     0            0.048                    0.000
3                     0            0.048                    0.760
4                     0            0.048                    0.544

   Total_no_of_material_use  CM= AAA  CM= BBB  CM= CCC  CM= DDD  CM= EEE  \
0                  0.038693        1        0        0        0        0
1                  0.059447        1        0        0        0        0
2                  0.000000        1        0        0        0        0
3                  0.089399        1        0        0        0        0
4                  0.042835        1        0        0        0        0

   CM= FFF  ...  IB= 50-60%  IB= 60-70%  IB= 70-80%  IB= 80-90%  IB= 90-100%  \
0        0  ...           0           0           0           0            1
1        0  ...           0           0           0           0            0
2        0  ...           0           0           0           0            0
3        0  ...           1           0           0           0            0
4        0  ...           1           0           0           0            0

   AB= 0-35  AB= 35-55  AB= 55<=  Disability  Completed_Course
0         0          0         1           0                 1
1         0          1         0           0                 1
2         0          1         0           1                 0
3         0          1         0           0                 1
4         1          0         0           0                 1

[5 rows x 49 columns]
```

Figure 1: The first five rows of the data after the preparation.

The data initially had four distinct possible values for ’final_grade’. This was changed as described at the end of section 2. Having a one in the ’Completed_Course’ column shows a pass or distinction in the course, and zero showing incompletion, that being a fail or withdrawal from the course.

Of course, the detail of a student’s exact grade banding is lost in this change to the classifications, but doing so creates two groups that have a lot of data, and having more data available in each group helps improve a model.

Therefore, this approach intends to see if it is even possible to roughly predict a students’ performance based on the data collected in section one, and not to get an exact prediction of a student’s grade banding.

Model Comparison

Using the sklearn method ’train_test_split()’, the data was split with 20% for testing and 80% Training, since there was enough data that even 20% was a large enough sample to test on reliably.

The first measurement used to compare the models was the ’accuracy score’. Model A, logistic regression, had a score of 83% when rounding up and Model B, decision tree, had 81%. As the data had a nearly even split between the two classes, this metric is useful but does not give enough detail alone.

ROC graphs, fig[2], were drawn. They continue to show that model A performed better by having a curve closer to the top left of the graph, indicating less trade-off between true-positive rate, sensitivity, and false-positive rate, specificity.

The AUC score, from the ROC graphs, confirms that A did better with a score of 0.8352 to 4 d.p compared to 0.8065 with B.

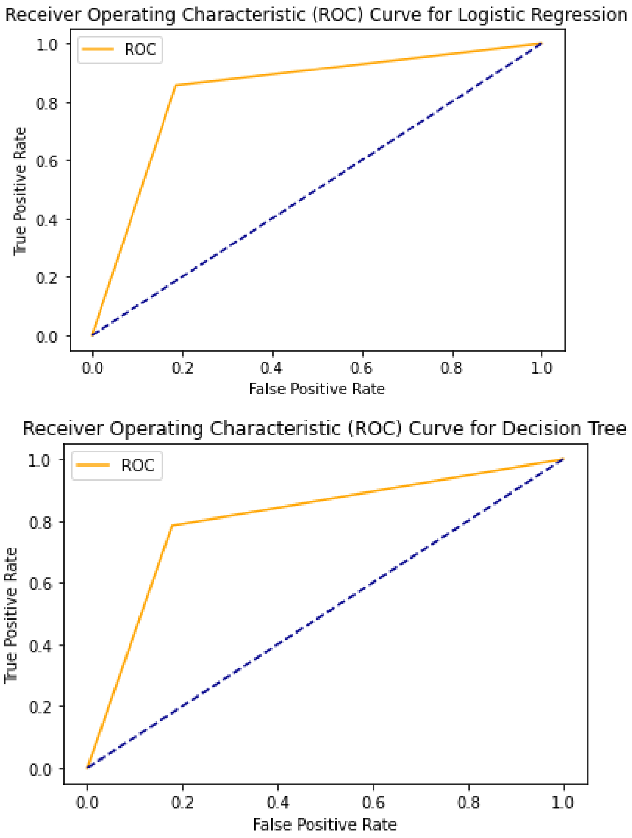


Figure 2: ROC graphs for both models.

Another accuracy measurement, much like 'accuracy score', is 'F1 score'. This measure is different because whereas accuracy measure was concerned about the ratio of true positives to total predictions, 'F1 score' has a more complicated formula that takes into account false negatives. Hence, giving this measurement a more rounded view of the model's accuracy.

False negatives are especially important in this case as accidentally failing a passing student would be far worse than accidentally passing a failing student. The score for A was 0.8394 and for B 0.8228. The higher the 'F1 score', the more precise the model.

Images of both model's confusion matrices have been included below to view the raw percentages of the predictions that were true positives, false positives, etc.

Analysis and Conclusion

Looking over all information in section 4, it's clear Logistic regression is the better model. Each measurement supports this and it also gives fewer false negatives than decision tree, as seen in the confusion matrices, which is very important for the model's application.

However, the decision tree model did have much fewer false positives, showing promise for the model perhaps if there was more data or variables to analyse.

In conclusion, both models did perform well, having good accuracy and not taking long to train. Although it is the Logistic regression model that is the better of the two.

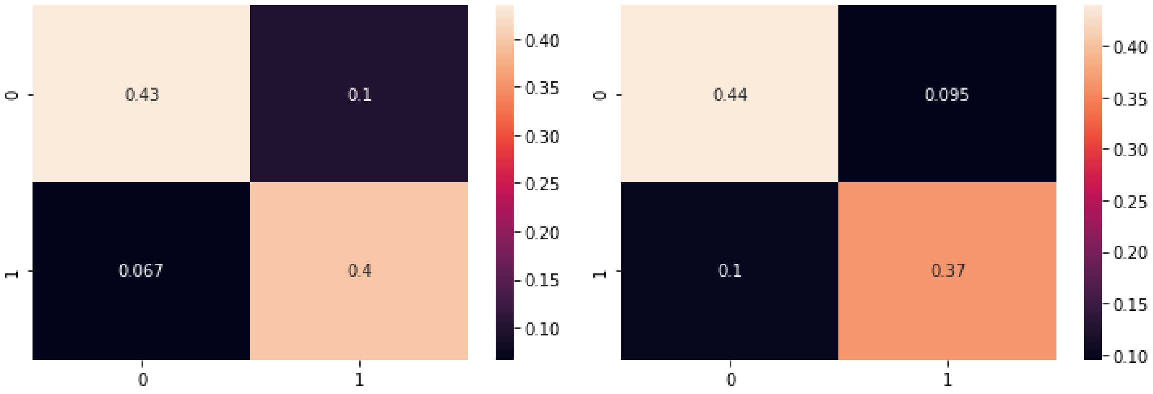


Figure 3: The confusion matrix for the logistic regression model, left, and for the decision tree model, right.