Report

Dataset

When deciding how to approach the dataset it is important to notice various properties about the data as they can affect how you manipulate the data. For example, many students have seemingly not completed all the assessments in their module, including their final exam. Therefore it is important to consider the weighted average of all completed and uncompleted exams to get a fuller picture of a student's performance. When calculating student marks it is also important to be aware that over 3,000 students have enrolled in 2 or more courses and so may have assessment marks relating to each course. Also in cases where a row of the studentAssessment table has an empty score I have decided to set the score equal to zero as a default value. I use a 70% random selection as the training set.

Feature Selection

In order to build a model, I first created a comprehensive dataset of all conceivable features, including a number of categorical and numerical data:

Categorical	**Numerical**
Module - combination of course and start date for student	Coursework Score
Age range	Exam Score
Region	Normalised Coursework Score
Highest education attained	Normalised Exam Score
Imd band (index of multiple deprivation)	Num of previous attempts
Disability	Sum of vle interactions
Gender	Studied credits

Categorical features can be transformed into either numerical ones (age range, imd band) or into boolean features. Disability and gender are already binary features allowing a very direct replacement. For region, highest education and module choice I chose to encode these as dense vectors, assigning a unique binary number to each unique value in the feature and expressing it over several boolean features.

Coursework and exam scores are simply weighted averages of each student's assessments across a module. The normalised versions of each feature is the result of an additional step, a normalised score in each assessment is the standard deviation from the mean

result from all students who completed the assessment. This is different from normalising all of the features to have 0 mean and unit variance which is the step applied to all non-boolean features once the total dataset has been generated.

Figure 1

When deciding which features to use in the final model my first step was to examine the variance of features. Figure 1 shows the variance of boolean features (numerical features are all normalised and have unit variance). The low variance of disability and particularly education bit 2 suggest that these features are unhelpful. EduBit2 actually encodes whether or not the student has a Post Grad qualification which over 99% of students lack, hence the low variance.

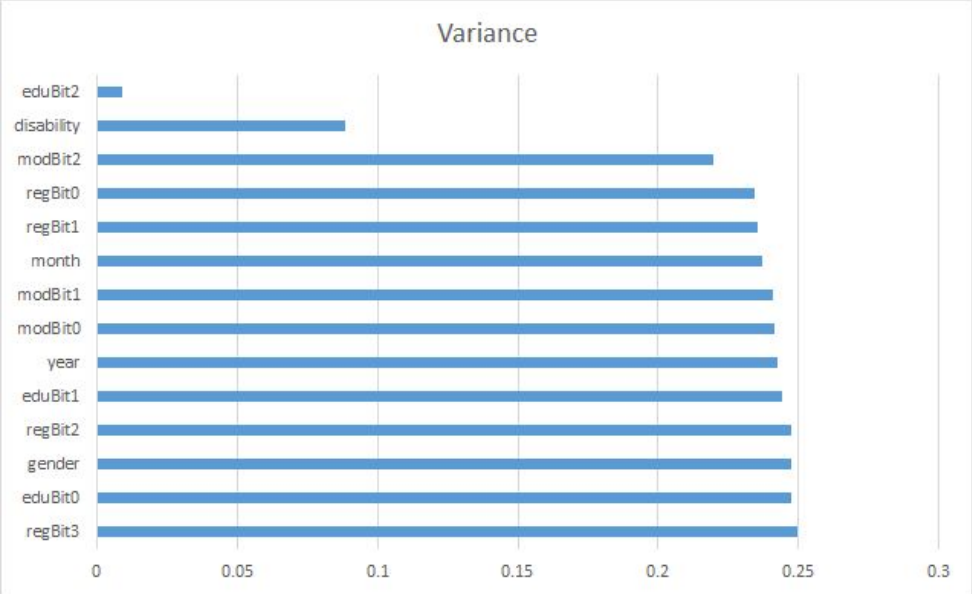


Figure 2

I then used a Random Forest model to calculate feature importances over the whole training set. Figure 2 shows that the most useful features are exam score, studied credits, imd band, normalised exam score, vle interactions, coursework score and normalised coursework score. Subsequent random forest models built using only the n best features support this conclusion as improvements in performance are very slight after n = 7, as shown by figure 3.

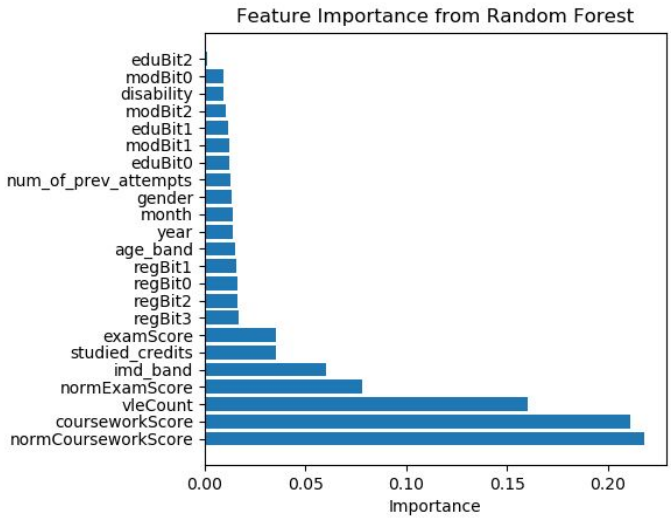
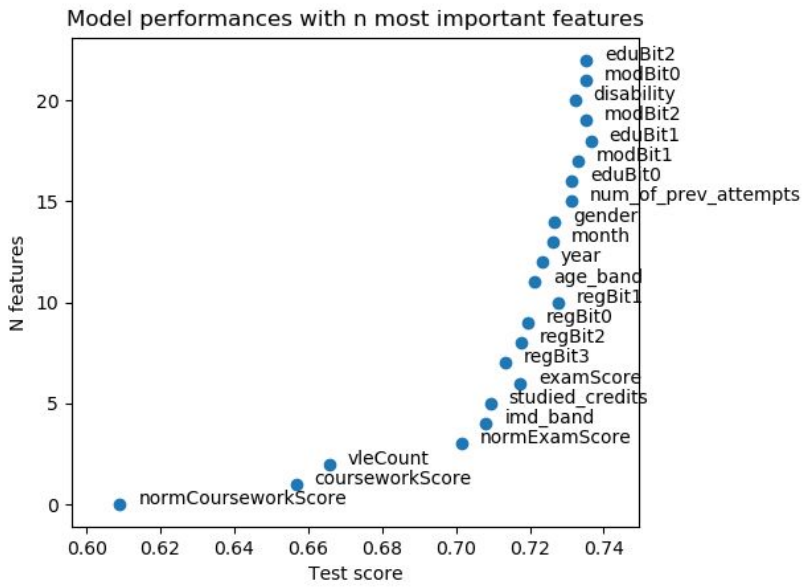


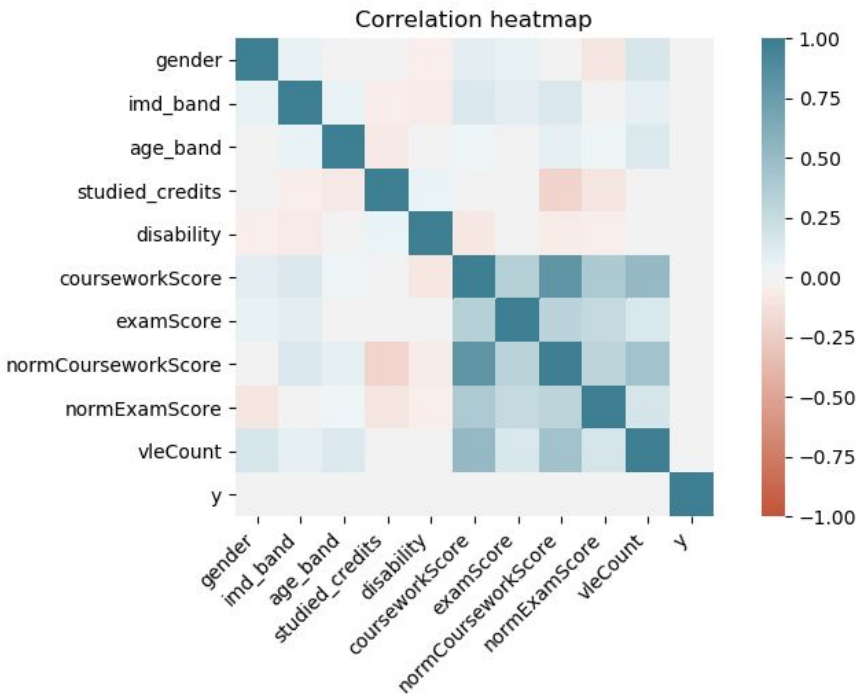
Figure 3



A more complete feature selection can be done by including a correlation heatmap. In the models for figures 2 and 3 inputs are mapped on to a categorical output with 4 possibilities (Withdrawn, Fail, Pass, Distinction). In order to generate correlations I used a binary value instead

(Withdrawn/Fail vs Pass/Distinction). Figure 4 shows the resulting heatmap when less significant features are removed to declutter the graph. As expected exam and coursework scores are strongly positively correlated with their normalised counterparts, as are vle interactions to a lesser extent. This suggests that a simpler model could be produced by removing the unnormalised score features. Indeed a model with its only 2 features being normalised coursework and exam scores gets a score of 0.674, compared to 0.657 when using the 2 most important features as per figure 2. The model with 4 features (normalised assessment scores and unnormalised scores) achieves a score of 0.700 which is very close to the model with normalised scores and vle interactions as its 3 features (0.693).

Figure 4



Finally, I trained a random forest on the dataset but added three ‘noise’ columns. Noise is normally distributed noise of unit variance around a zero mean and small and large noise are boolean variables with variances 0.005 and 0.250 respectively. It is interesting that the noise feature is interpreted by the model as being more important than all but three other features, as figure 5 shows.

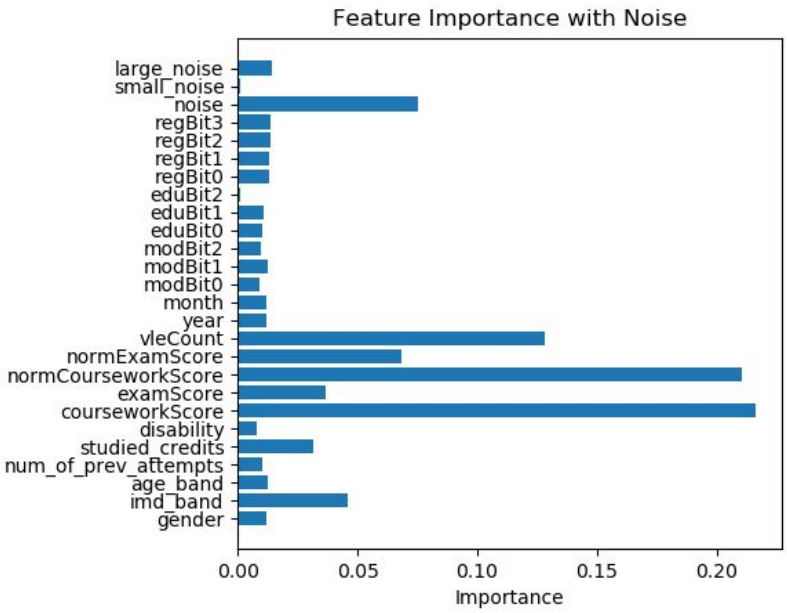


Figure 5

Model Selection

I chose Random Forests and Logistic Regression as my two models. Random Forests is suitable for solving the problem as a classification problem with 4 outcomes. Logistic Regression is better suited for a binary decision so for this model I predicted output between two classes (Withdrawn/Fail and Pass/Distinction). Because the output is unbalanced performance can be slightly improved by setting class weights to balanced in the Logistic Regression

model. Using the feature selection data from the previous section we can build a model to predict a simple pass/fail outcome. Based on scoring from the test set this model achieves an accuracy of 89.3% and 89.4% for unbalanced and balanced models respectively. These models use the 5 best features (excluding the highly correlated unnormalised assessment scores). This model produces this confusion matrix:

	Predict Fail	Predict Pass
Is Fail	10437 - true neg	1609 - false pos
Is Pass	679 - false neg	10090 - true pos

Hyperparameters

A simple hyperparameter of a random forest is the number of decision trees, which affects performance on the test set as shown:

Num. of trees	Score
1	0.646
11	0.688
21	0.692
31	0.700
...	
101	0.704

Between 31 and 101 trees progress is very slow and incremental, compared to the large increases from 1 to 31. This makes a forest with 31 trees the most time-efficient random forest model, being quicker to train at a minimal cost to performance compared to larger models.

Conclusion

In conclusion, the two models perform relatively well at their respective tasks. It is to be expected that linear regression is more accurate at binary classification since the problem is inherently less challenging with less room for error. This explains the difference in accuracy. For the logistic regression model it has precision of 0.862 and recall of 0.937 as calculated from the confusion matrix. These are good values and suggest that the model performs well when features are selected well and that logistic regression is a good model.