Predictive Modelling for the classification of High Performing and Low Performing students.

Preprocessing

We developed a classification model to predict student performance from the information in the Open University Dataset. We used the studentVle_vle_info file which contained the data on the student's information such as their id, demographic data, and final results, as well as the information on the virtual learning environment that they interacted with.

We grouped this data by the activity type and the student id and totalled up the sum of clicks. This shows each activity type the student interacted with and the total number of times the student interacted with that activity type. This also retains the demographic data on the student.

Grouping the data like this removed any missing values that were present. We checked for complete cases and all cases in our grouped data were complete. The variables which we included in the were: *activity_type, code_module, gender, region, highest_education, imd_band, age_band, num_of_prev_attempts , disability, final_result* and *sum_click* (the total clicks for each student interacting with each activity type).

The data was pre-processed in this way to ensure the information on the particular activity types that each student interacted with was retained and used in the classification models. Previously, the importance of activity types has not been included in models predicting student performance and it has been mentioned that further studies could include this data in predictive modelling (Waheed et al., 2020).

Encoding categorical variables

As many of the features of this data were categorical, and many machine learning models cannot work directly with categorical data, we use encoding for the categorical variables. We use Ordinal encoding for the variables where the categories had an order to them. These categories included *age_band, imd_band* and *highest_education.* For the variables where there was no order to the categories, we used one-

hot-encoding. This included *activity_type, code_module, gender, region* and *disability.*

Defining High performing and Low performing.

We created binary classification for the models and defined the students as those who were High Performing and those who were Low Performing. The students categorised as High Performing had a *final_result* of Distinction or Pass. The students categorised as Low Performing had a *final_result* of Fail. We chose to remove students with a *final_result* of Withdrawn in line with (Aljohani, Fayoumi and Hassan, 2019).

Splitting the data

We split the data into a training, test and validation set, with a split of 60%, 20%, 20% respectively. The training set is used to train the models, the testing set is used to test model performance, and the validation set is used to show how the final model would perform on unseen data. Splitting data in this way is done to prevent overfitting (Bilal *et al.*, 2022) of the model to the training set and allows the performance of the model on unseen data (out of sample) to be assessed.

Resampling

There was a large difference in the training set between the numbers of the groups High Performing (79968) and Low Performing (29273). We chose to resample the data using a synthetic minority oversampling technique (SMOTE), in this case we used SVM-SMOTE. SMOTE has previously been used as the resampling technique with the Open University dataset (Bilal *et al.*, 2022). We chose SVM-SMOTE as it has been shown to have better performance compared to other resampling methods when performed on similar data to measure student performance (Ghorbani and Ghousi, 2020).

Feature Selection

We first applied a Random Forest model. This model used five hundred decision trees to show the feature importance for each variable. The feature importance of each variable was plotted to visualise which features were the most important in the classification of the

students. From this, all features with an importance greater than 0.014 were selected to use in the classification models. This selected the top twenty most important features. As encoding the categorical data resulted in the dataset containing many features, the feature selection approach was used to reduce the number of features from fifty to twenty. We chose to use Random Forest for feature selection as it has been seen that feature selection with a Random Forest model can improve the performance of classification model used for predicting student performance (Deepika and Sathyanarayana, 2019). Feature selection is a common process when using data with high dimensionality and can improve speed and performance of machine learning models as well as help prevent overfitting (Li *et al.*, 2018).

Scaling the data

Some classification algorithms such as k-nearest neighbours require all features of the data to on the same scale in order to perfrom well. Scaling prevents features with much larger values, such as the *sum_clicks* from having more influence in the model. We used Normalisation as an approach to scaling the data as it is a method previously used in other studies using the Open University Dataset to build machine learning algorithms (Tomasevic, Gvozdenovic and Vranes, 2020; Waheed *et al.*, 2020).

Results

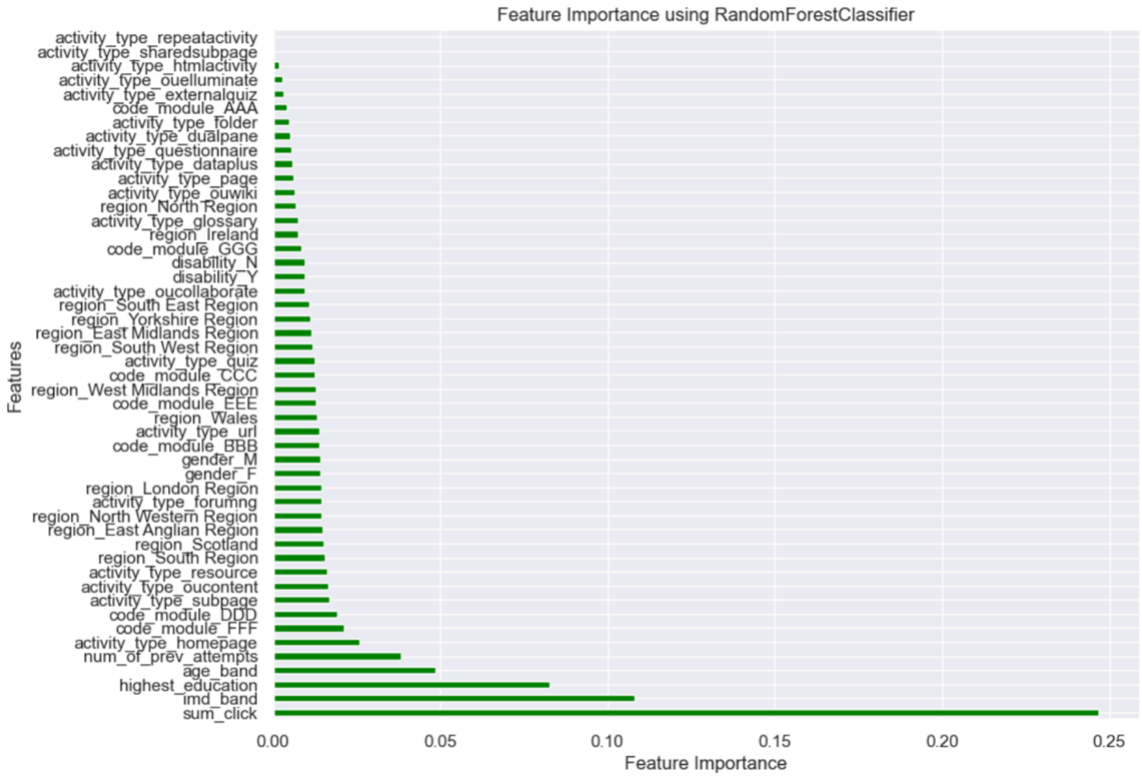

Figure 1. Plot of the feature importance for each variable.

The bar chart in Figure 1 shows the how much each feature contributed to the classification. The larger the mean decrease in impurity, the greater that feature's importance. This was produced by performing a random forest model which used the Gini Impurity as the metric for measuring the quality of the split. The total value of all the feature importance adds up to 1. From Figure 1, we can see that the most important features
are *sum_clicks, imd_band, highest_education, age_band, num_of_previous_attempts,* and *activity_type_homepage.* This shows
the *sum_clicks* to be the most important feature.

This corresponds with our earlier findings that the interaction with the virtual learning environment and final result of the student are linked. It also suggests that certain demographic features are important in the classification. We can see that interaction with particular activity types of the virtual learning environment is involved in the classification.

Random Forest model

From the set with the most important twenty features, we performed a Random Forest model for classification. This was performed using the RandomForestClassifer from sklearn.

```
Classification Report:
                precision    recall  f1-score   support

High Performing       0.80      0.86      0.83     26751
 Low Performing       0.51      0.41      0.45      9663

       accuracy                           0.74     36414
      macro avg       0.66      0.63      0.64     36414
   weighted avg       0.72      0.74      0.73     36414
```

```
Text(0.5, 1.0, 'Confusion Matrix showing the percentage of test data in each group')
```

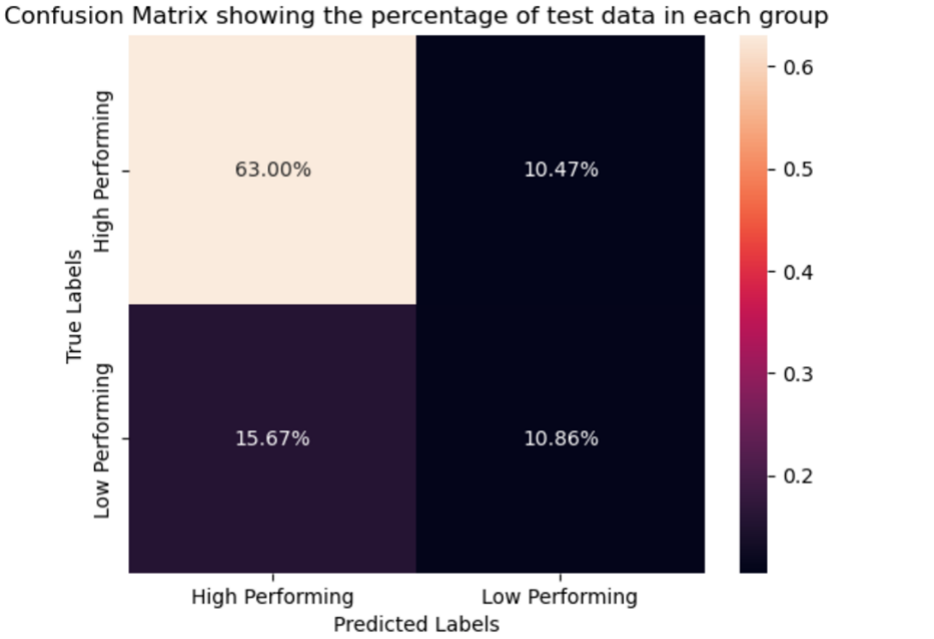

Figure 2. The performance of the Random Forest model. Table showing the Precision, Recall, F1 score and Accuracy. The confusion matrix showing the percentages of the test set that were classified into each group.

From Figure 2, we can see the performance of the Random Forest model on the test data. The confusion matrix shows the percentage of test data that is categorised into each group.

Using the Precision value as the metric for measuring the performance of our classification models, we can see that this model has a precision of 80% regarding true positives (high performing classified as high performing). This means that 20% of the students which were classified as high performing were in fact low performing. From the confusion

matrix, we can see that 15.67% of the test data was classed as High performing when the true label is Low performing. It also shows that 63% of the data received a true label of High Performing.

Decision Tree

```
Classification Report:
                precision    recall  f1-score   support

High Performing       0.79      0.80      0.80     26751
 Low Performing       0.43      0.41      0.42      9663

       accuracy                           0.70     36414
      macro avg       0.61      0.61      0.61     36414
   weighted avg       0.69      0.70      0.70     36414
```

```
Text(0.5, 1.0, 'Confusion Matrix showing the percentage of test data in each group')
```

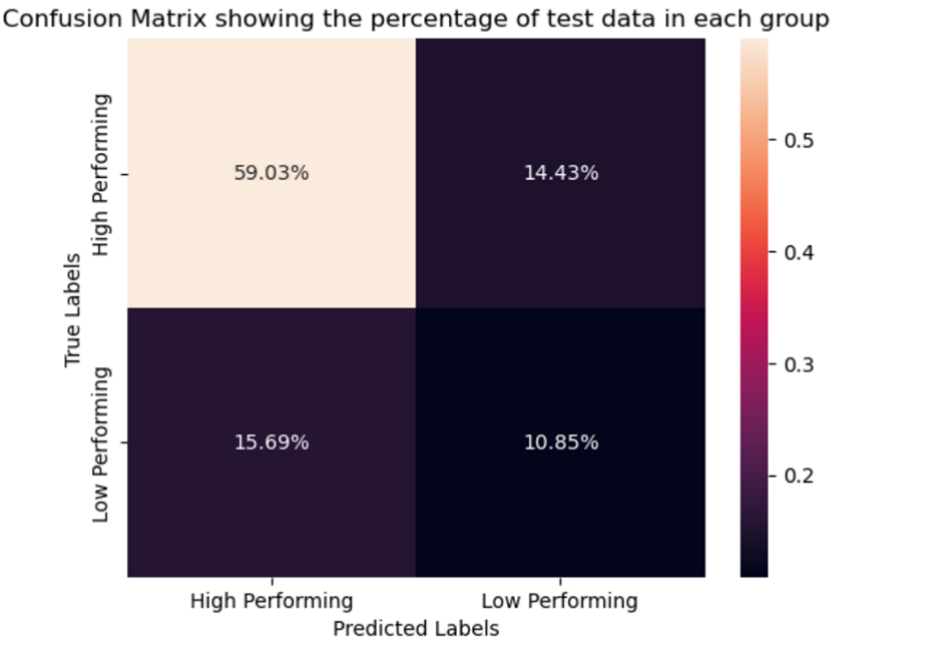

Figure 3. The performance of the Decision Tree model. Table showing the Precision, Recall, F1 score and Accuracy. The confusion matrix showing the percentages of the test set that were classified into each group.

From Figure 3 we can see how the Decision Tree model performed. It produced a Precision of 79% and an Accuracy of 70%. As expected, the Decision Tree model performs less well than the Random Forest Model.

K-nearest neighbours

After scaling the data using Normalisation, a K-nearest neighbours model (KNN) was developed using Euclidian

distance. A value for k (number of neighbours) was calculated through testing the accuracy of the model using a sample data. K=25 was used for the final model.

```
Classification Report:
                precision    recall  f1-score   support

High Performing       0.79      0.85      0.82     26751
 Low Performing       0.46      0.36      0.40      9663

       accuracy                           0.72     36414
      macro avg       0.62      0.60      0.61     36414
   weighted avg       0.70      0.72      0.71     36414
```

```
]: Text(0.5, 1.0, 'Confusion Matrix showing the percentage of test data in each group')
```

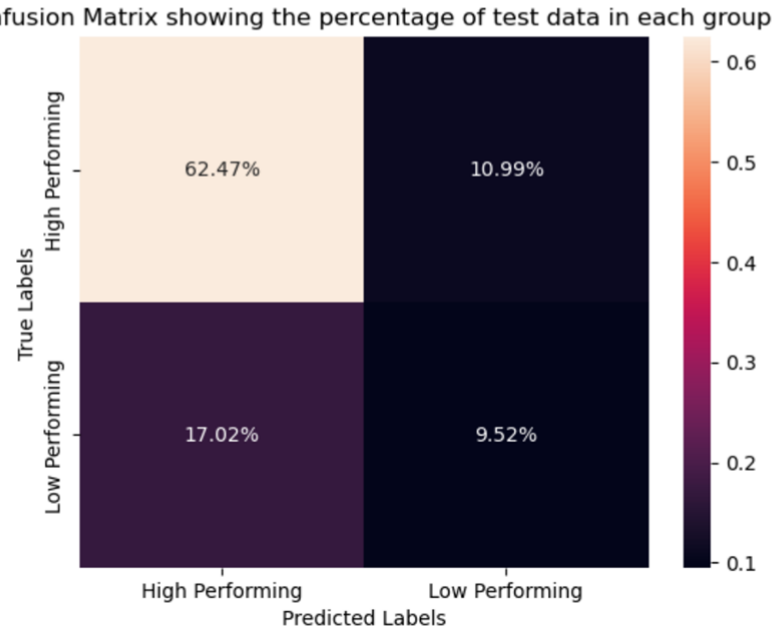

Figure 4. The performance of the K-nearest neighbours model. Table showing the Precision, Recall, F1 score and Accuracy. The confusion matrix showing the percentages of the test set that were classified into each group.

Figure 4 shows that the KNN model produces a Precision score of 79% and an Accuracy of 72%.

Validation

The Random Forest model performed the best out of the tested models. It produced the highest Precision score of the models; therefore, it was chosen to be run with the validation set.

```
Classification Report:
                precision    recall  f1-score   support

High Performing       0.80      0.86      0.83     26814
 Low Performing       0.52      0.41      0.46      9600

       accuracy                           0.75     36414
      macro avg       0.66      0.64      0.65     36414
   weighted avg       0.73      0.75      0.74     36414
```

```
|: Text(0.5, 1.0, 'Confusion Matrix showing the percentage of test data in each group')
```

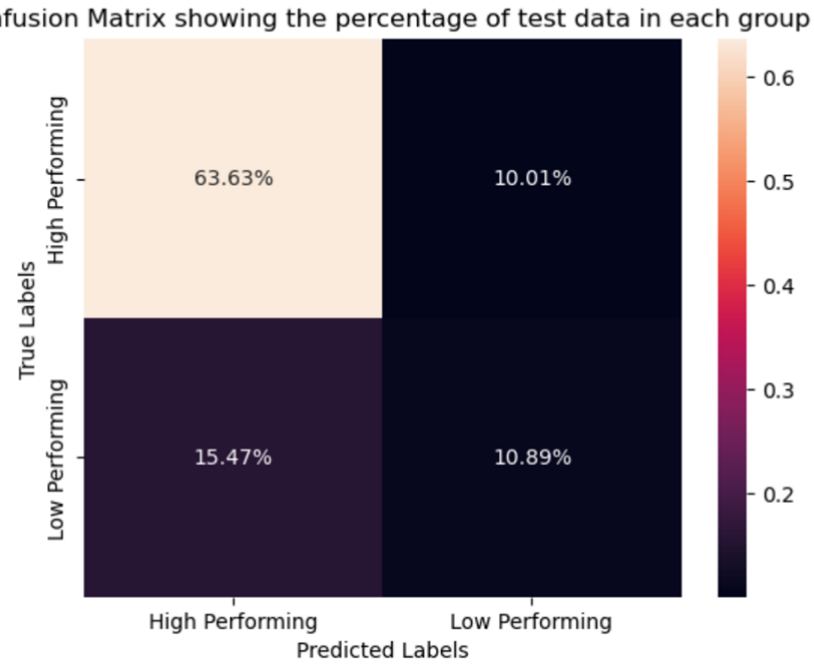

Figure 5. The performance of the Random Forest with the validation set. Table showing the Precision, Recall, F1 score and Accuracy. The confusion matrix showing the percentages of the test set that were classified into each group.

From Figure 5 we can see how the Random Forest model performed with the out of sample data. This shows how the model will perform on unseen data. The model shows a similar performance on the validation as it did with the test set. It produced a Precision of 80% with the validation set, it produced a Precision of 80% with the in-sample test set. With the validation set the model produced an accuracy of 75% and with the in-sample test set it produced an accuracy of 74%.

The purpose of these classification models would be to predict students who could be at risk of failing (Low Performing). This could enable targeted support to be provided to them to help them succeed in their courses. If these students were to be classified as ‘High Performing’ by mistake, this could mean they do not get supported adequately and could remain at risk of failing. Therefore, the priority of these models is to minimise the number of false positives which would be students that

are ‘Low Performing’ that get classified as ‘High Performing’. This is why we choose the Precision value as the main metric for evaluating the performance of the models. Precision is the metric used when the goal is to minimise false positives, a higher value of Precision means fewer false positives in the model.

Future work

Future work could look at comparing different feature selection processes on the results of the models. The work could compare which particular features and the numbers of features retained from the different selection processes and how these affect the models’ performance. It could work towards optimising the preprocessing methods to get optimal performance out of the classification models.

For example, looking at whether the model performance differs using Principal Component Analysis (PCA) for dimensionality reduction and selecting the number of components to retain based on Eigen values.

Aljohani, N.R., Fayoumi, A. and Hassan, S.-U. (2019) Predicting At-Risk Students Using Clickstream Data in the Virtual Learning Environment. *Sustainability* [online]. 11 (24), p. 7238. Available from: https://www.mdpi.com/2071-1050/11/24/7238 [Accessed 3 August 2023].

Bilal, M., Omar, M., Anwar, W., Bokhari, R.H. and Choi, G.S. (2022) The role of demographic and academic features in a student performance prediction. *Scientific Reports* [online]. 12 (1), p. 12508.

Ghorbani, R. and Ghousi, R. (2020) Comparing Different Resampling Methods in Predicting Students’ Performance Using Machine Learning Techniques. *IEEE Access* [online]IEEE Access. 8, pp. 67899–67911.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J. and Liu, H. (2018) Feature Selection: A Data Perspective. *ACM Computing Surveys* [online]. 50 (6), pp. 1–45. Available from: https://dl.acm.org/doi/10.1145/3136625 [Accessed 2 August 2023].

Talla Padmavathi College of Engineering, Deepika, K., Sathyanarayana, N., and Nagole Institute of Technology and Sciences (2019) Relief-F and Budget Tree Random Forest Based Feature Selection for Student Academic Performance Prediction. *International Journal of Intelligent Engineering and Systems* [online]. 12 (1), pp. 30–39. Available from: http://www.inass.org/2019/2019022804.pdf [Accessed 2 August 2023].

Tomasevic, N., Gvozdenovic, N. and Vranes, S. (2020) An overview and comparison of supervised data mining techniques for student exam performance prediction. *Computers & Education* [online]. 143, p. 103676. Available from: https://www.sciencedirect.com/science/article/pii/S0360131519302295 [Accessed 6 July 2023].

Waheed, H., Hassan, S.-U., Aljohani, N.R., Hardman, J., Alelyani, S. and Nawaz, R. (2020) Predicting academic performance of students from VLE big data using deep learning models. *Computers in Human Behavior* [online]. 104, p. 106189. Available from: https://www.sciencedirect.com/science/article/pii/S0747563219304017 [Accessed 4 July 2023].