USING SUPERVISED LEARNING TO PREDICT STUDENT DROPOUT

Diptarko Chowdhury

16TH OCTOBER 2024

Problem Statement

Students dropping out of enrolled courses negatively impact educational institutions financially and reputationally – both in terms of academic prestige and student satisfaction. This study is undertaken to assess whether existing data can be used to predict which students are more at risk of dropping out, so that measures can be taken, and resources allocated, to avoid this.

The Data Set & Exploratory Data Analysis

The dataset (Figure 1) used for initial analysis consisted of 25,059 entries with eight features plus the target variable *'CompletedCourse'*; a zero in this field indicates a dropout.

#	Column	Non-Null Count	Dtype
0	CentreName	25059 non-null	object
1	DateofBirth	25059 non-null	object
2	Gender	25059 non-null	object
3	CourseLevel	25059 non-null	object
4	IsFirstIntake	25059 non-null	bool
5	CompletedCourse	25059 non-null	object
6	CreditWeightedAverage	22763 non-null	float64
7	ProgressionUniversity	25059 non-null	object
8	UnauthorisedAbsenceCount	24851 non-null	float64

Figure 1:Description of data set

Upon removing all null values, the *'DateofBirth'* column was replaced by 'Age' which transformed this information about the student into a usable, continuous variable. Subsequently using `sklearn`, all continuous variables were standardised via the `StandardScaler()` function; all categorical variables encoded using `OneHotEncoder()`, and the target variable was binarized using `LabelBinarizer()`.

```
Column 'CentreName' has 19 unique entries.
Column 'Gender' has 2 unique entries.
Column 'CourseLevel' has 4 unique entries.
Column 'CompletedCourse' has 2 unique entries.
Column 'ProgressionUniversity' has 39 unique entries.
```

Figure 2: Number of unique entries for categorical features

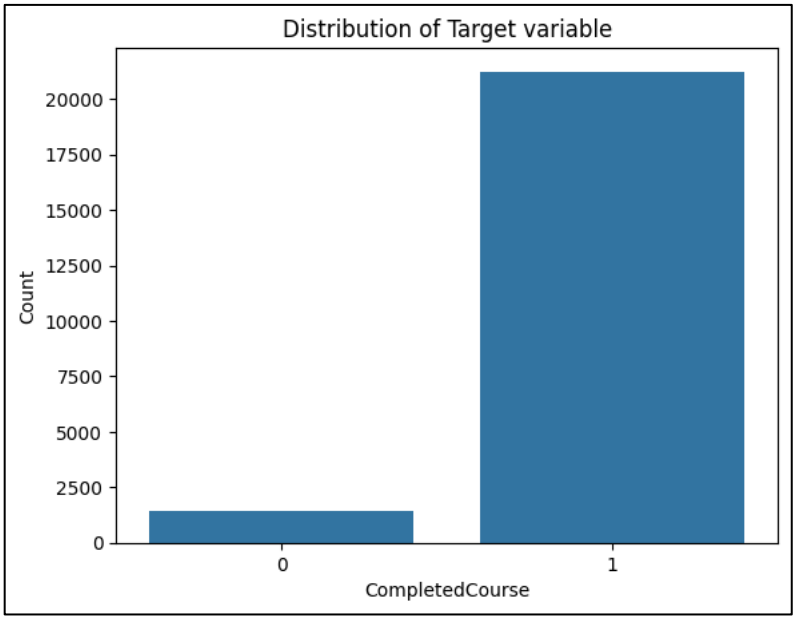


Figure 3: Distribution of target variable

In the process of encoding categorical variables, two features (*'CentreName'* and *'ProgressionUniversity'*) were seen to contain many unique values. For further model refinement, experimentation in group binning should be conducted for these features.

Initial exploration of the target variable 'CompletedCourse' showed a very imbalanced distribution, there were far more students completing the course than dropping out. Therefore, accuracy will not be a reliable metric when it comes to judging a model's effectiveness, and techniques such as stratified splitting should be used when making training/validation/test subsets of the data.

XGBoost Model

The first model used to predict student drop-out was the extreme gradient boosting (XGBoost) ensemble technique. In this technique, the base learner is constructed by initialising a prediction (or probability in binary classification) of

$$p = 0.5$$

for every entry. This equates to a log-odds of zero for all entries:

$$\text{log-odds} = \log\left(\frac{p}{1-p}\right) = 0$$

A second base learner is then formed by fitting decision trees to the residuals. XGboost considers all possible splits and chooses the one that maximises gain as given by the Similarity weight.

$$Similarity\ Weight = \left(\frac{(\sum \text{Residuals})^2}{\sum p \times (1-p) + \lambda}\right)$$

The penalisation parameter, λ, prevents any single node from having an excessively large weight thus limiting the total tree complexity and helping to avoid overfitting. Subsequent decision trees are built recursively, with residuals recorded at each level, and trees adjusted to maximise fit. Once the full tree is built, the model prunes nodes if they are deemed to be below a given threshold (preventing overfitting).

Upon fitting XGBoost on a training set of the total data (80% split with target variable stratified), the model predicted outcomes for the remining test set with the following results (Figure 4).

```
XGBoost Model Accuracy: 0.9700308505949757
XGBoost Precision: 0.9803784162578837
XGBoost Recall: 0.9877618263120734
XGBoost AUC: 0.9794570022989328

Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.71      0.75       289
           1       0.98      0.99      0.98      4249

    accuracy                           0.97      4538
   macro avg       0.89      0.85      0.87      4538
weighted avg       0.97      0.97      0.97      4538
```

Figure 4: Performance indicators and classification report of the baseline standard XGBoost model

The model performs well overall, with all values close to 1. However, the use-case of this model requires the recall to be high for True label = no (*'CompletedCourse'* = 0). In this capacity, the recall score is 0.71, which is suboptimal. To rectify this, class weight is applied to the XGBoost model which gives more importance to predicting class 0 using the inbuilt `scale_pos_weight` parameter. This produces a 0.90 recall for the 0 class (Figure 5).

```
XGBoost Model Accuracy (Weighted): 0.9543851917144116
XGBoost Precision (Weighted): 0.9926864943929791
XGBoost Recall (Weighted): 0.9583431395622499
XGBoost AUC (Weighted): 0.9797123035666441

Classification Report (Weighted):
              precision    recall  f1-score   support

           0       0.59      0.90      0.71       289
           1       0.99      0.96      0.98      4249

    accuracy                           0.95      4538
   macro avg       0.79      0.93      0.84      4538
weighted avg       0.97      0.95      0.96      4538
```

Figure 5: Performance indicators and classification report of the baseline, weighted XGBoost model

This model was then tuned by testing optimal combination of its hyperparameters (learning rate, maximum tree depth, and number of estimators), to give the following best settings and results (Figure 6).

```
Best hyperparameters found: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 100}
Tuned XGBoost Model Accuracy: 0.9431467606875276
Tuned XGBoost Precision: 0.9952841896252171
Tuned XGBoost Recall: 0.9437514709343375
Tuned XGBoost AUC: 0.9816443681843315

Classification Report (Tuned):
              precision    recall  f1-score   support

           0       0.53      0.93      0.68       289
           1       1.00      0.94      0.97      4249

    accuracy                           0.94      4538
   macro avg       0.76      0.94      0.82      4538
weighted avg       0.97      0.94      0.95      4538
```

Figure 6: Optimal hyperparameter values, performance indicators, and classification report of the tuned, weighted XGBoost model

The model has improved further, increasing the class 0 recall to 0.93. At this stage two new features were added to the dataset (and standardised), *'AttendancePercentage'* and *'ContactHours'*, with results shown (Figure 7).

```
Accuracy with new features: 0.9363155575143235
Precision with new features: 0.9962406015037594
Recall with new features: 0.9355142386443869
AUC with new features: 0.9807713762896378

Classification Report (Weighted, Tuned, New Features):
              precision    recall  f1-score   support

           0       0.50      0.95      0.65       289
           1       1.00      0.94      0.96      4249

    accuracy                           0.94      4538
   macro avg       0.75      0.94      0.81      4538
weighted avg       0.96      0.94      0.95      4538
```

Figure 7: Performance indicators and classification report of the tuned, weighted XGBoost model with two added features.

This had the effect of increasing the relevant recall further to 0.95 – thus making it the best model for catching students at risk of dropping out. The confusion matrices for all four models are shown in Figure 8; each alteration made to the model enhanced its suitability for this use case.

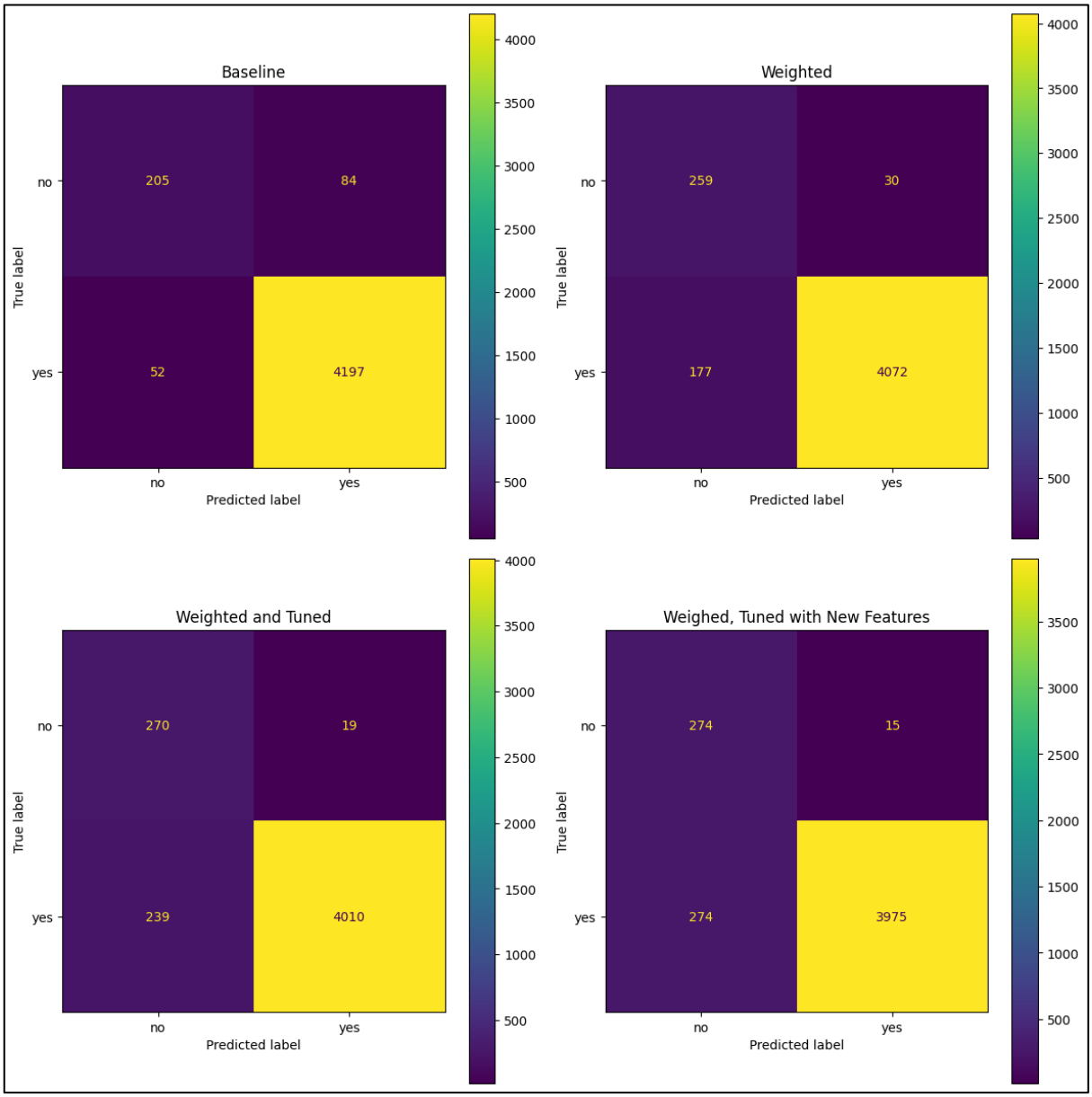


Figure 8: Confusion matrices for each of the four XGBoost models constructed, with the false positives reducing from 84 to 30 to 19 to 15.

The importance of features as determined by the built in `plot_importance` feature inn XGBoost, and via the Shapley Additive Explanations (SHAP) are shown in Figure 9.

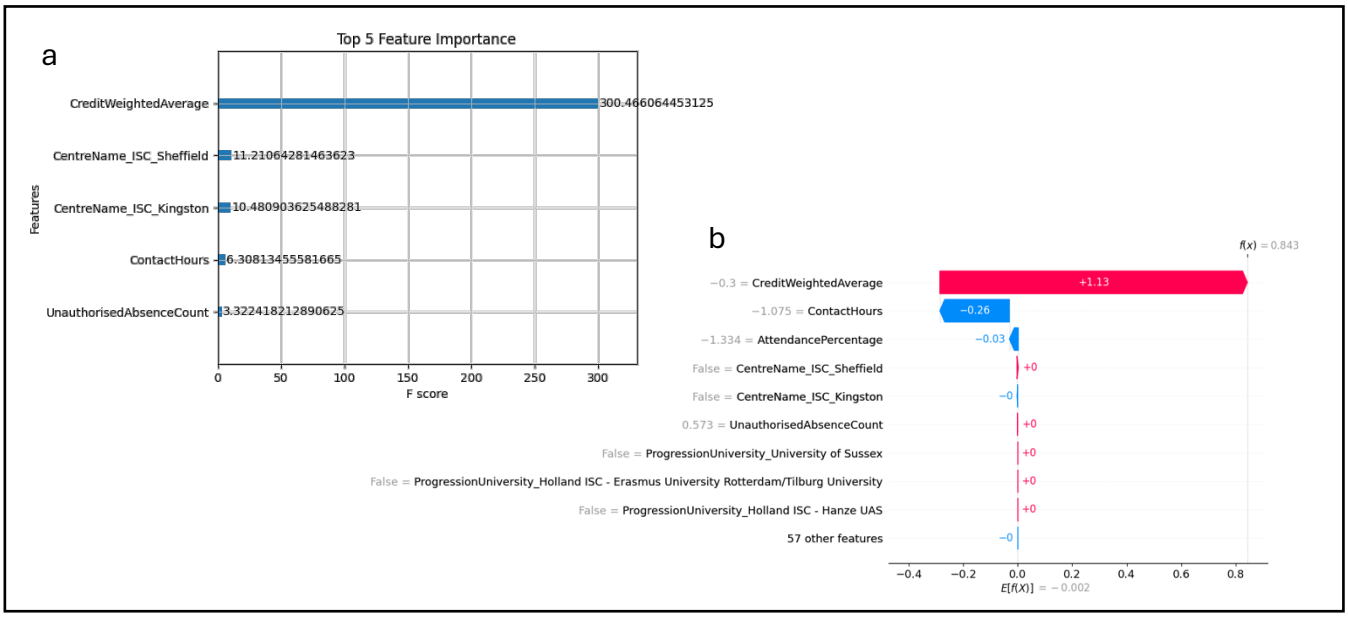


Figure 9: (a) Feature importance for the weighted, tuned model with extra features as given by plot_importance (b) Feature importance for the same model given by SHAP

Neural Network Method

The second method used to predict dropouts was a neural network.

An initial network was constructed with the structure shown in Figure 10, with two

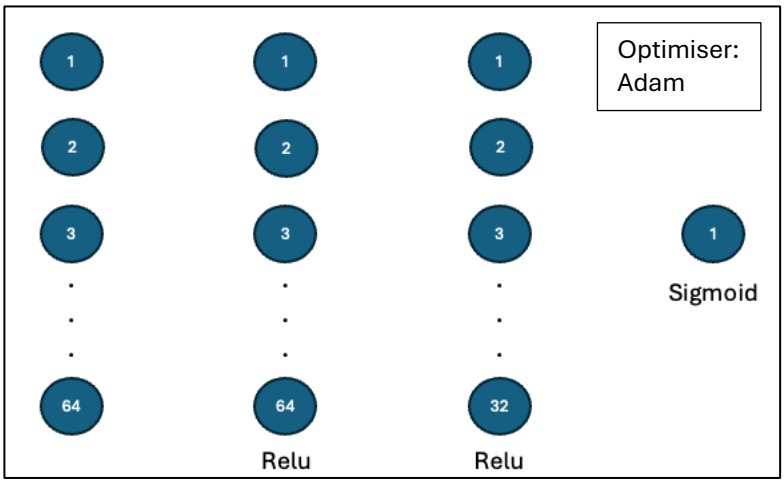
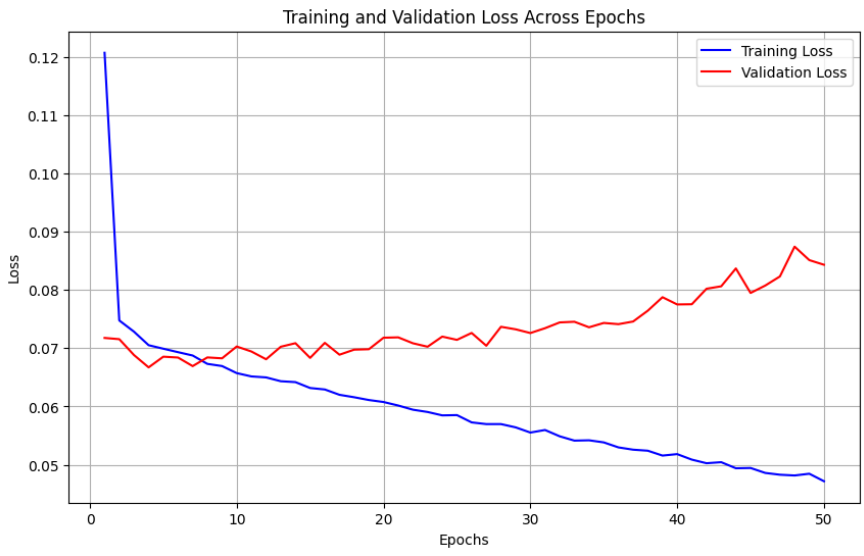


Figure 10: Structure of the initial neural network

hidden layers of 64 and 32 neurons with relu activation, and an output neuron with sigmoid activation to reflect the binary classification required. This model was trained over 50 epochs, an Adam optimizer and with a batch size of 32 to produce results shown in Figure 11.

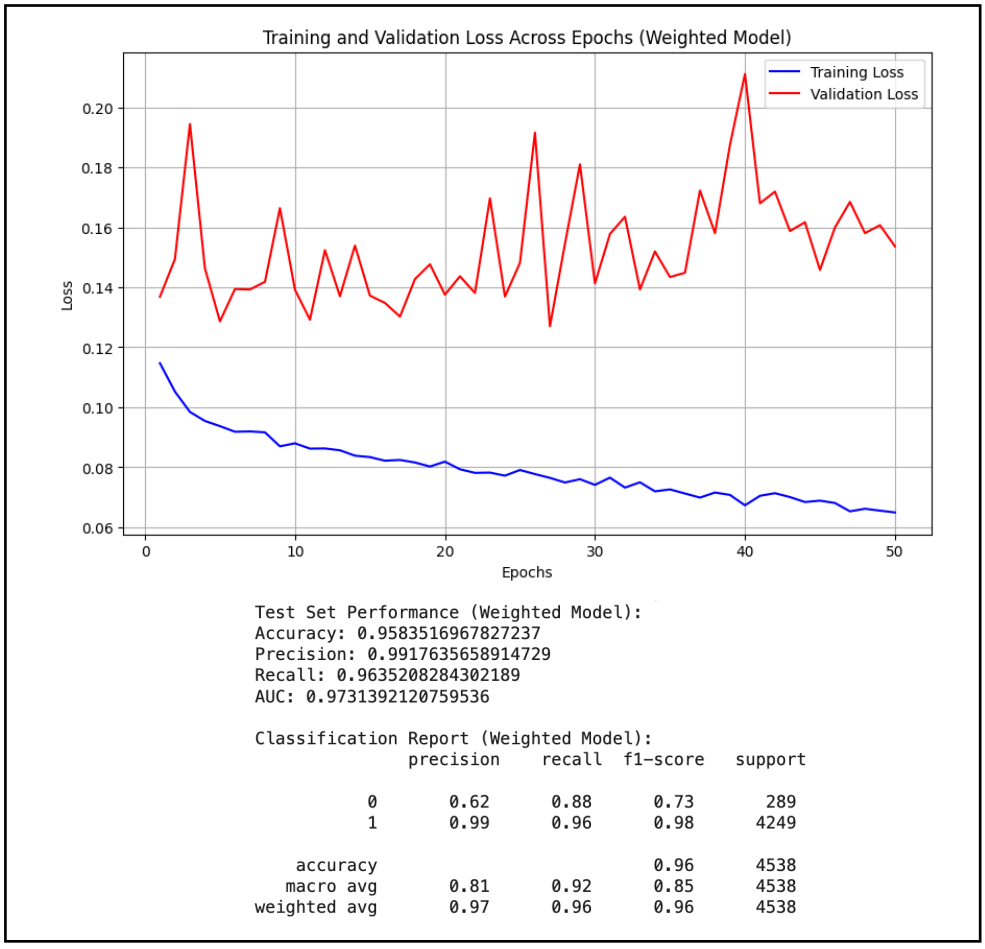



```
Test Set Performance:
Accuracy: 0.9731159100925518
Precision: 0.9808902353763691
Recall: 0.9905860202400565
AUC: 0.9786552667389272

Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.72      0.77       289
           1       0.98      0.99      0.99      4249

    accuracy                           0.97      4538
   macro avg       0.91      0.85      0.88      4538
weighted avg       0.97      0.97      0.97      4538
```

Figure 11: Training and validation loss across epochs, performance indicators, and classification report for initial neural network

This neural network suffers from overfitting beyond 10 epochs, and although performs highly on all four indicators, the recall for the 0 class is low at 0.72.

To rectify this, added weighting is given on the 0 class using the `compute_class_weight` method from `sklearn`. This model gives results as in Figure 12.




```
Test Set Performance (Weighted Model):
Accuracy: 0.9583516967827237
Precision: 0.9917635658914729
Recall: 0.9635208284302189
AUC: 0.9731392120759536

Classification Report (Weighted Model):
              precision    recall  f1-score   support

           0       0.62      0.88      0.73       289
           1       0.99      0.96      0.98      4249

    accuracy                           0.96      4538
   macro avg       0.81      0.92      0.85      4538
weighted avg       0.97      0.96      0.96      4538
```

Figure 12: Training and validation loss across epochs, performance indicators, and classification report for the weighted initial neural network.

Recall on 0 class has increased to 0.88 – a step in the right direction. The validation loss curve is erratic however and may indicate overfitting. To optimise the neural network, hyperparameter tuning is undertaken, which opted for a network as in Figure 13, which had results as can be seen in Figure 14 – tuning increased the 0-recall class further to 0.90.

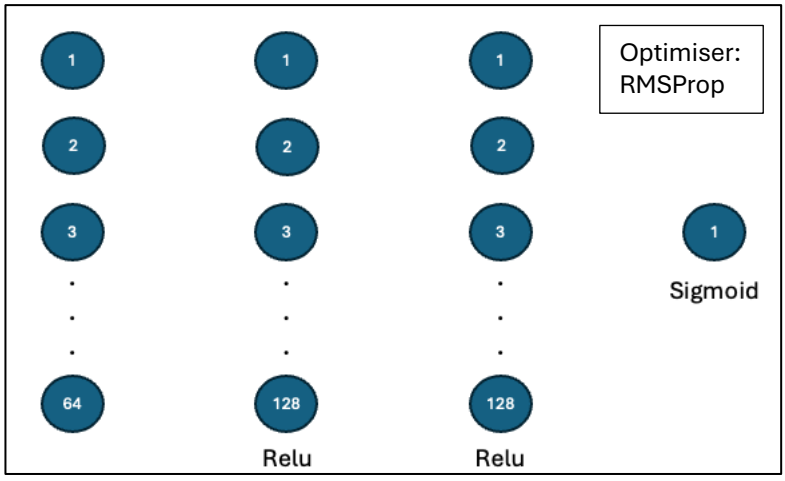
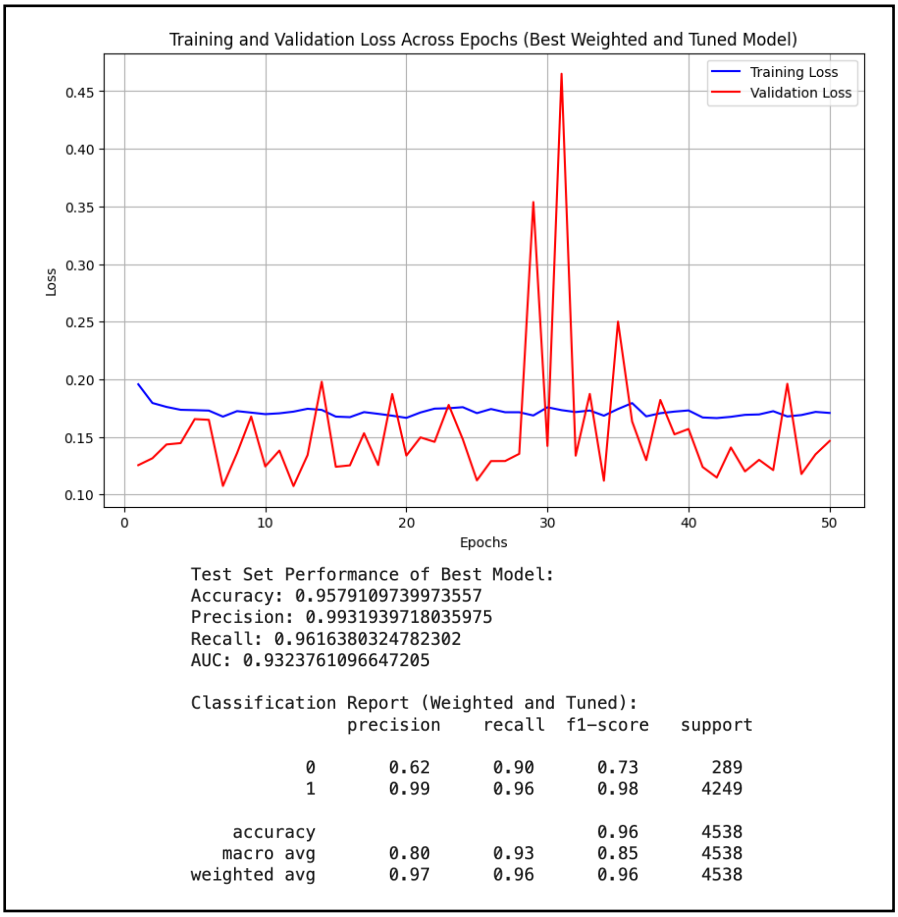


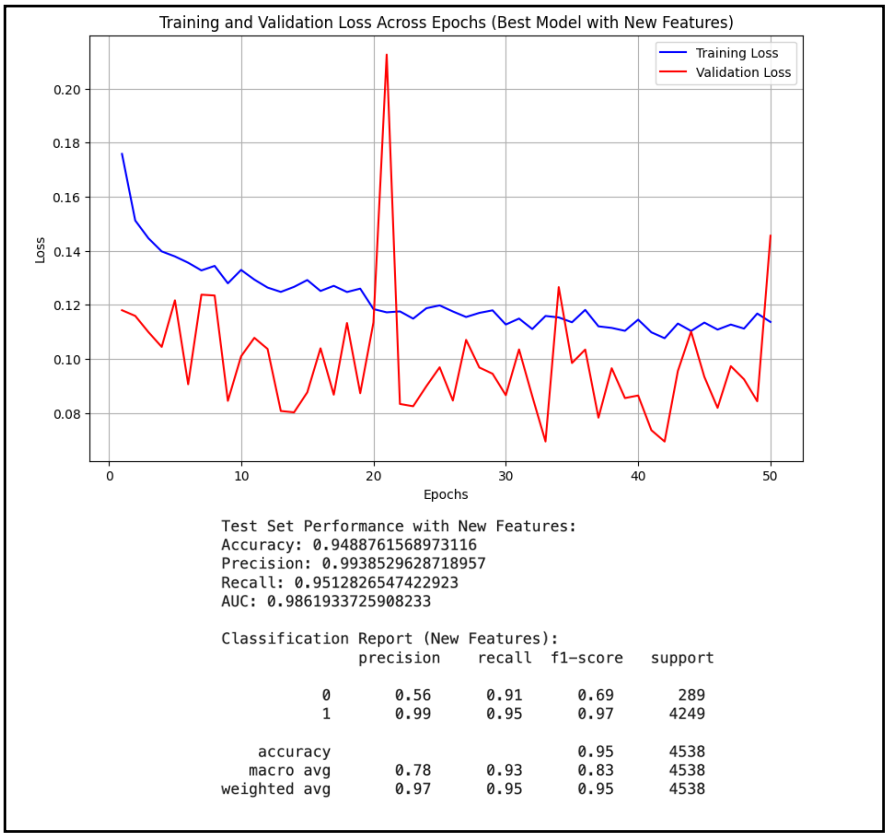
Figure 13: Structure of Tuned Neural Network




```
Test Set Performance of Best Model:
Accuracy: 0.9579109739973557
Precision: 0.9931939718035975
Recall: 0.9616380324782302
AUC: 0.9323761096647205

Classification Report (Weighted and Tuned):
              precision    recall  f1-score   support

           0       0.62      0.90      0.73       289
           1       0.99      0.96      0.98      4249

    accuracy                           0.96      4538
   macro avg       0.80      0.93      0.85      4538
weighted avg       0.97      0.96      0.96      4538
```

Figure 14: Training and validation loss across epochs, performance indicators, and classification report for the weighted, tuned neural network.

At this stage, as with XGBoost, *'AttendancePercentage'* and *'ContactHours'* were added back into the dataset, and the best performing Neural Network model was then fit giving results as in Figure 15.




```
Test Set Performance with New Features:
Accuracy: 0.9488761568973116
Precision: 0.9938529628718957
Recall: 0.9512826547422923
AUC: 0.9861933725908233

Classification Report (New Features):
              precision    recall  f1-score   support

           0       0.56      0.91      0.69       289
           1       0.99      0.95      0.97      4249

    accuracy                           0.95      4538
   macro avg       0.78      0.93      0.83      4538
weighted avg       0.97      0.95      0.95      4538
```

Figure 15: Training and validation loss across epochs, performance indicators, and classification report for the weighted, tuned neural network with new features.

As can be seen from Figures 15 and 16, the confusion matrices for all neural network models constructed, each alteration to the model resulted in a better 0-class recall and fewer false positives.

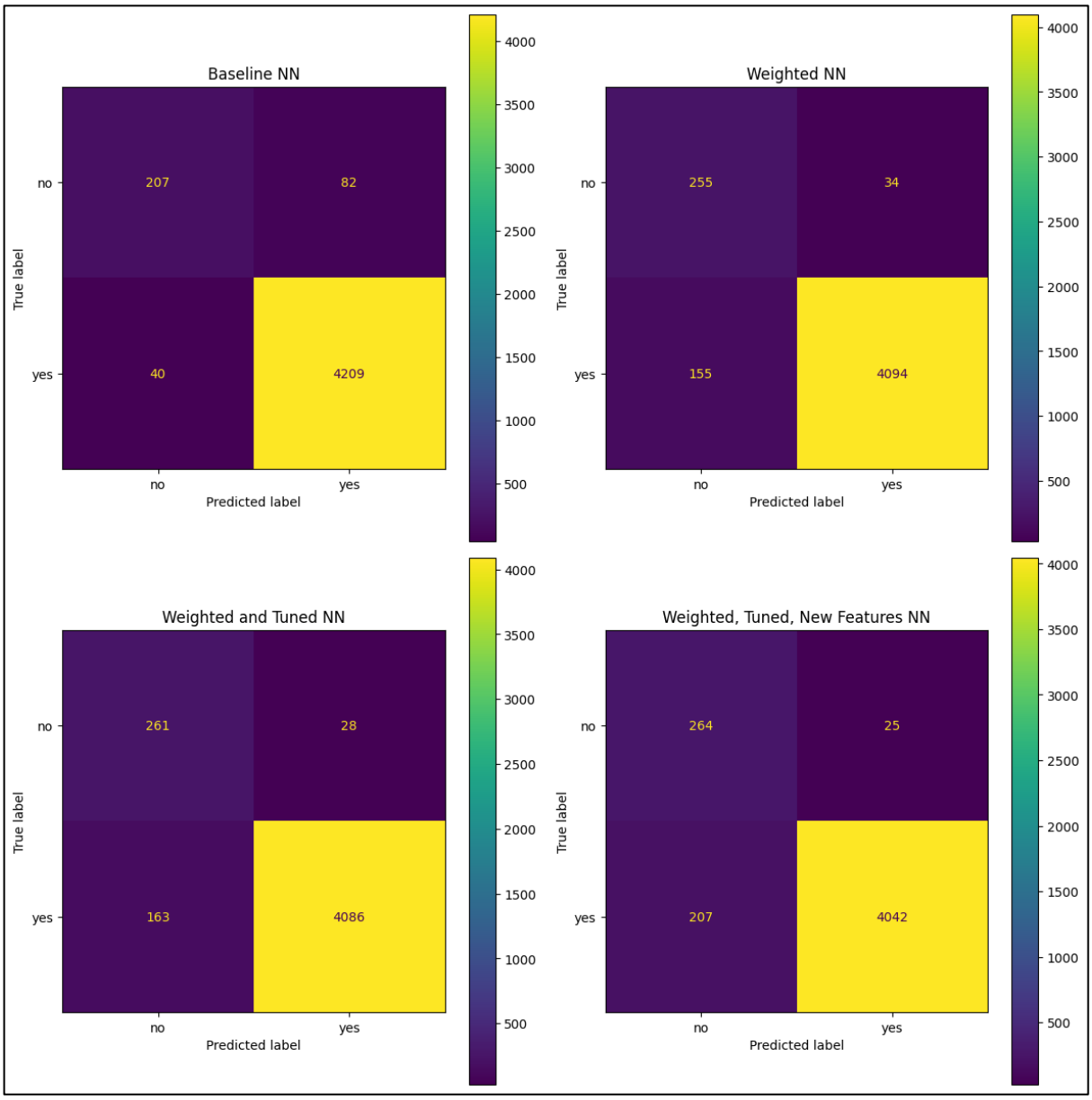


Figure 16: Confusion matrices for each of the four neural network models constructed, with the false positives reducing from 82 to 34 to 28 to 25.

Conclusion

Both models perform well at detecting students dropping out, and both improved upon implementing weighting for the 0-class, hyperparameter tuning, and adding the two new features. This suggests that experimentation with more features and/or further tuning may be beneficial for the models.

However, in this business case, where spotting students at high risk of dropping out is the goal, XGBoost is preferable; it boasts a higher recall rate (amongst other performance indicators), with only 15 false negatives in the best model. Furthermore, the validation loss curves for the neural network models fluctuate significantly indicating possible overfitting. A better NN model may be achieved with further hyperparameter tuning and/or regularisation but at this stage XGBoost is recommended.

If proactive action is to be taken, Figure 9b should be considered. This graph shows that an increase in contact hours is the biggest factor in reducing the probability a student will complete the course. The presents an opportunity for further analysis investigating why courses with more contact hours result in higher rates of dropouts.