2. Background

2.? Logistic Regression

Logistic regression is a supervised machine learning model, a variant of the linear regression model in which the outcome variable is dichotomous [1]. Unlike linear regression, logistic regression is used for classification, in which the model finds the boundary line of the classification between the two classes [2]. This is represented by the conditional probability $P(Y|X)$ where X is a real number (input features) and the random variable Y takes the value of 1 or 0 [3]. The input variable X can be desribed as a set of features $(x_0, x_1, x_2, ..., x_n)$ which are multiplied by a set of associated weights $(w_0, w_1, w_2, ..., w_n)$, and summed together to produce the dot product [2]. Therefore:

$$z = w_0x_0 + w_1x_1 + w_2x_2 + ... + w_nx_n$$
$$z = w \cdot x = w^Tx$$

This is then passed to a function which converts z to a value between 0 and 1. The Sigmoid function is the most widely used function for this [2], and the distribution is shown in Figure 1. The Sigmoid function is given by:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

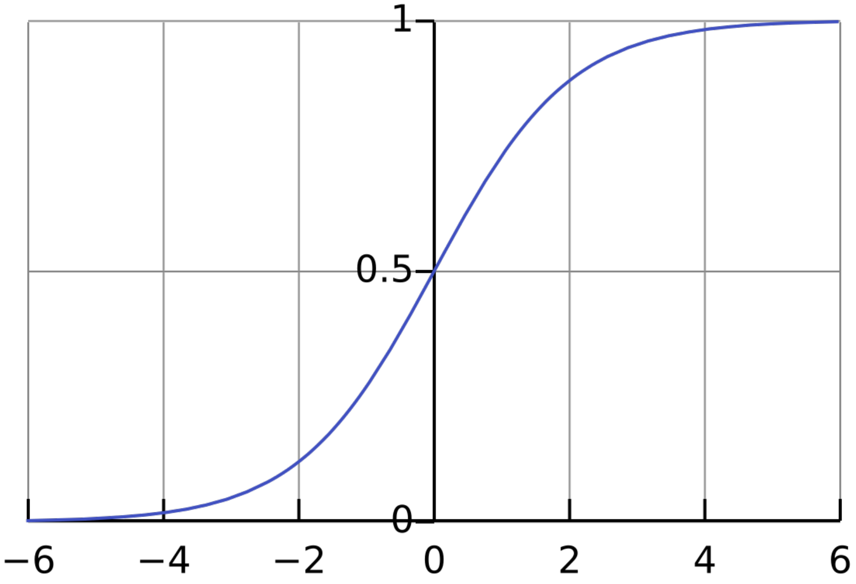

Figure 1: Sigmoid function.

After being passed through the Sigmoid function, the data point is then given the prediction of belonging to the class 0 or 1. Therefore, if $\sigma(w^Tx) \geq 0.5$ then the outcome is class 1, and if $\sigma(w^Tx) < 0.5$ then the outcome is class 0. The weights of the logistic regression model are found during training the model using **discuss Logistic Regression cost function**.

While standard logistic regression gives a binary outcome variable, it can also be used for multi-class problems. This is done using techniques such as one-vs-rest (OvR). The one-vs-rest model trains separate binary classifiers for each class, with each classifier distinguishing one class from the rest **Source? Find other techniques**.

3. Data Preparation

3.1. Exploratory Analysis

The initial dataset consisted of 4424 records with 36 independent variables, with no missing values. The dataset was processed using the *pandas* library in Python.

- **Note: Should I include means of variance tables despite not all variables being in the final MI dataset?**

Each student entry has one of 3 Target variables: Graduated, Dropout, or Enrolled (where the student took another three years to complete the course). The distribution of the Target variable is shown in Figure 2, which shows that it is imbalanced, where the Graduated Target makes up 50% of the data records. This was addressed using the Synthetic Minority Over-sampling Technique (SMOTE), an oversampling algorithm which generates synthetic data using *k* nearest neighbours [4]. The SMOTE implementation from *imbalanced-learn* in the *sci-kit learn* library was used.

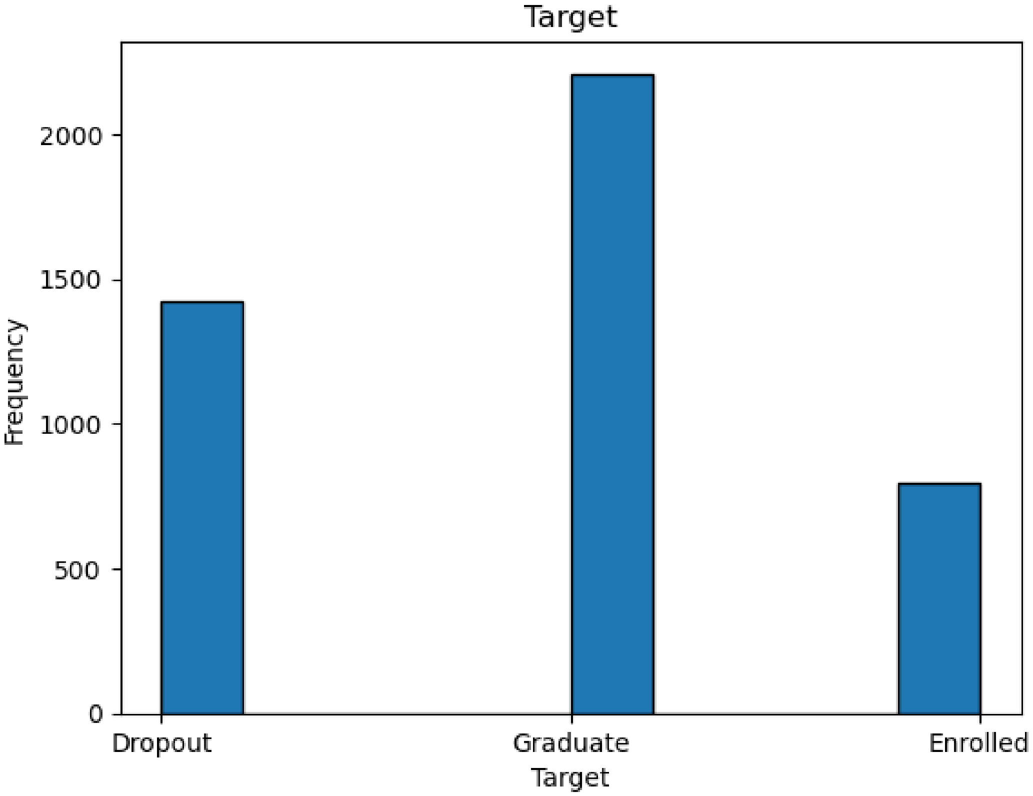

Figure 2: Count of Target variable.

3.2. Feature Engineering

Mutual Information (MI) is a filter-based feature selection method which measures how mutually dependent two variables are [5]. This method was used to filter out redundant variables, and was opted for since MI can be used for both discrete and continuous variables. The MI score with target was computed for each variable using *sci-kit learn*. Each feature is assigned a scoring value, with the resulting features being organised in descending order based on the scores and are assigned rankings for the features [5]. A limitation of this method is that while it measures a feature's importance by it's correlation with the target, it assumes independence of features from each other [6]. Therefore, choosing the highest scoring variables from MI can lead to redundant features can be selected.

To address this, a pairwise MI matrix was computed. The values were normalised to a range between 0 and 1, presented in Figure 3. Redundant variables were removed by checking each MI pairwise score according to a set threshold (0.99) and removing the variable with the lower MI score with the target. The final selected predictors resulted in 17 independent variables.

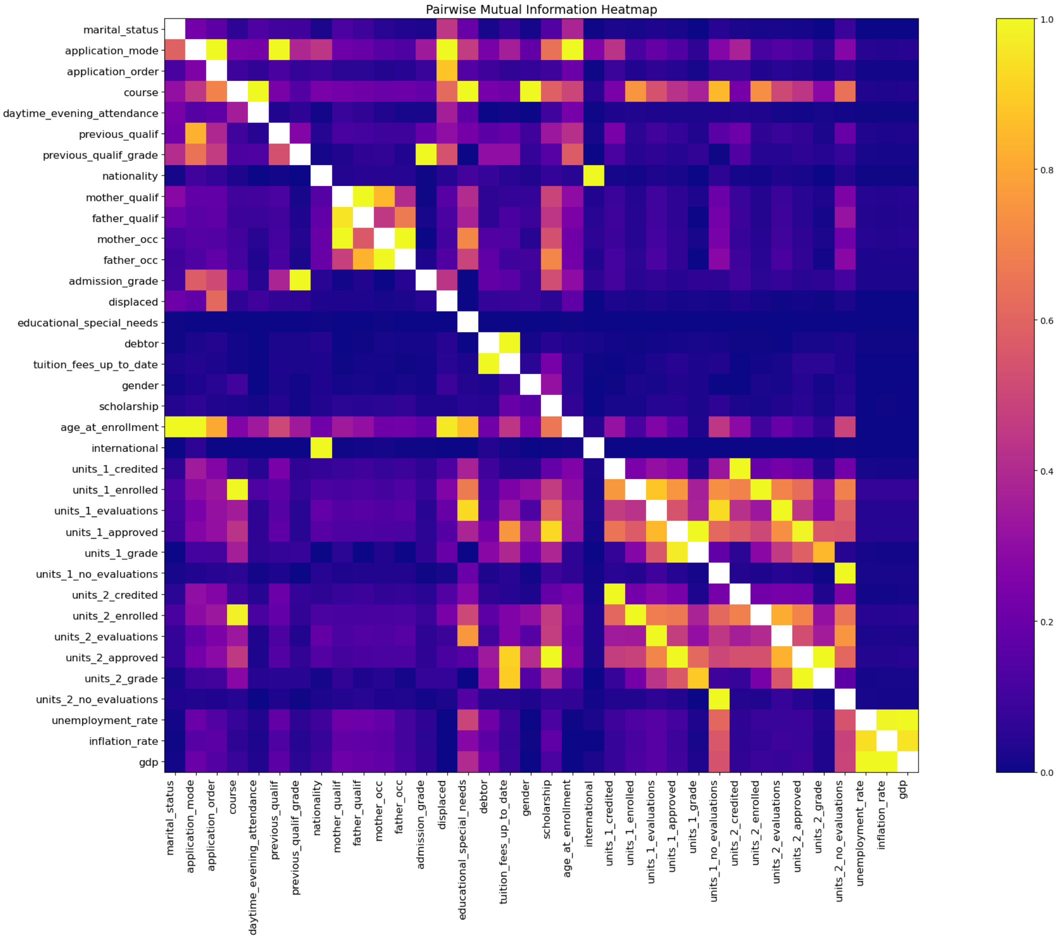

Figure 3: Heatmap of pairwise Mutual Information scores.

Principal Component Analysis (PCA) was then performed on the dataset for dimensionsality reduction [7]. **Explain PCA process briefly.** The final resulting dataset consisted of 8 independent variables.

4. Experiments

4.? Logistic Regression

The multi-class logistic regression model was implemented using the *sci-kit learn* library in Python. The dataset was split into training, validation, and test sets using a 70/20/10 split to ensure that the model was trained on a representative sample and tested on unseen data. Feature scaling was applied using **StandardScaler?** to ensure that the coefficients of the logistic regression were appropriately scaled.

The LogisticRegression class was utilised - 'multinomial' option? OvR or cross-entropy loss? Solver lbfgs. Model evaluation was performed on the test set using metrics the accuracy, precision, recall, F1-score metrics, and a confusion matrix to assess classification performance. The *scikit-learn* classification_report and confusion_matrix functions were used for this.

Table 1 shows the results of the classification report and Figure 4 shows the confusion matrix. The overall model accuracy is 61%. Results show that the Dropout target has the highest Precision and F1-Scores, indicating that the model has the best results for predicting this class over others. The model struggles most in predicting the Enrolled target, with an F1-score of 0.54.

Target	Precision	Recall	F1-Score
Graduate	0.61	0.59	0.60
Dropout	0.74	0.67	0.70
Enrolled	0.51	0.57	0.54

Table 1: Classification report for Logistic Regression model.

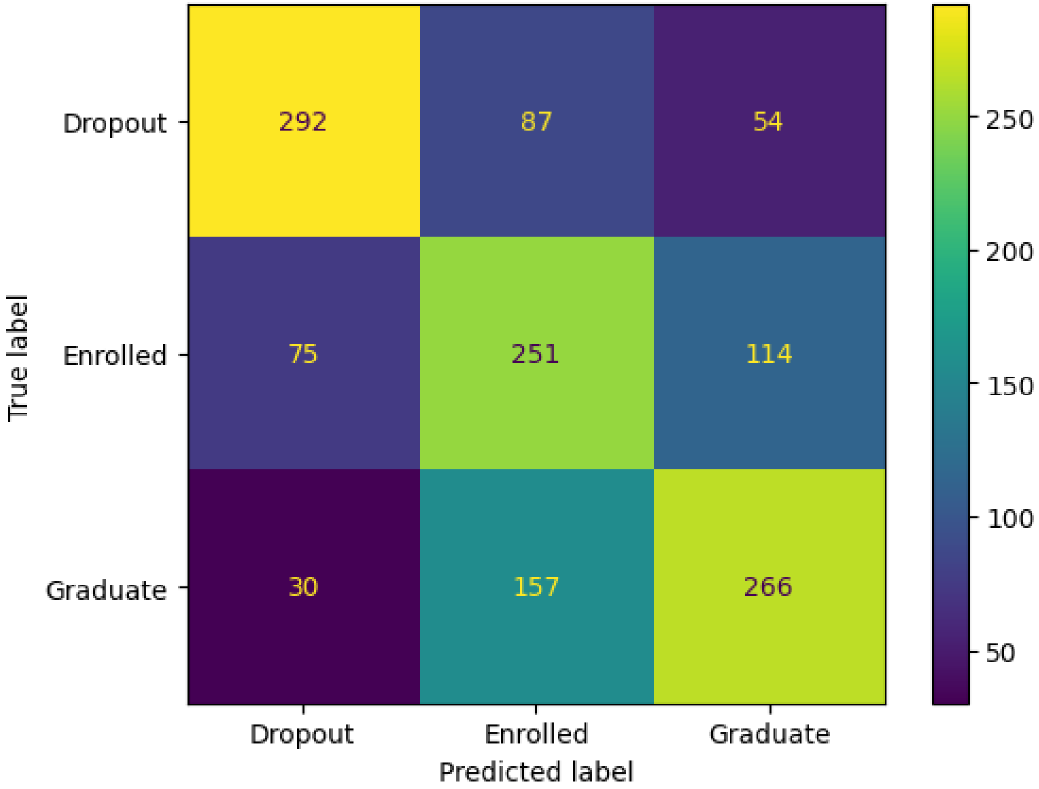

Figure 4: Confusion Matrix of the Logistic Regression model.

References

[1] D. W. Hosmer Jr., S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression* (Wiley Series in Probability and Statistics). Wiley, 2013, ISBN: 9780470582473.

[2] X. Zou, Y. Hu, Z. Tian, and K. Shen, "Logistic regression model optimization and case analysis," in *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, 2019, pp. 135–139. DOI: 10.1109/ICCSNT47585.2019.8962457.

[3] H. Li, "Logistic regression and maximum entropy model," in *Machine Learning Methods*. Singapore: Springer Nature Singapore, 2024, pp. 103–125, ISBN: 978-981-99-3917-6. DOI: 10.1007/978-981-99-3917-6_6. [Online]. Available: https://doi.org/10.1007/978-981-99-3917-6_6.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002, ISSN: 1076-9757. DOI: 10.1613/jair.953. [Online]. Available: http://dx.doi.org/10.1613/jair.953.

[5] B. Venkatesh and J. Anuradha, "A review of feature selection and its methods," *Cybernetics and Information Technologies*, vol. 19, no. 1, pp. 3–26, 2019. DOI: 10.2478/cait-2019-0001. [Online]. Available: https://doi.org/10.2478/cait-2019-0001.

[6] J. Li, K. Cheng, S. Wang, *et al.*, "Feature selection: A data perspective," vol. 50, no. 6, Dec. 2017, ISSN: 0360-0300. DOI: 10.1145/3136625. [Online]. Available: https://doi.org/10.1145/3136625.

[7] F. L. Gewers, G. R. Ferreira, H. F. D. Arruda, *et al.*, "Principal component analysis: A natural approach to data exploration," *ACM Comput. Surv.*, vol. 54, no. 4, May 2021, ISSN: 0360-0300. DOI: 10.1145/3447755. [Online]. Available: https://doi-org.ejournals.um.edu.mt/10.1145/3447755.