**NAME**: **Nithin Joshi Cherkupally**

**STUDENT ID: 15127022**

**EMAIL:** [**cherkupaln@coventry.ac.uk**](mailto:cherkupaln@coventry.ac.uk)

**Module Name: Natural Language Processing**

**Module Code: 7120CEM**

**Assignment Title: Machine Learning and Deep Learning Solutions for Language-Related Problems - CW2**

**Solving Text Classification Challenges: Harmful Content Detection and Slot Filling Using Deep Learning Models**

**Abstract**

Intent detection and slot filling in natural language processing (NLP)-the work is centered on dialog systems, which capture and interpret information extracted through such media. We are dealing with two of the deep learning architectures: BiLSTM-CRF as a model for slot filling and DistilBERT as a model for intent detection. The BiLSTM-CRF model can maximize relevant extracted entities, which are further referred to as slots, from the text. On the other hand, DistilBERT performs well in identifying content as harm or non-harm per se in the text. The dataset used is further pre-processed by filling up with missing values, encoding categorical variables, and balancing with SMOTE to include only intent and slot information in the text and the labels. So as far as the experiments went, the DistilBERT was a model very good in achieving accuracy for intent detection while the BiLSTM-CRF was the model giving the best performance in slot filling. The experiments occurred using metrics such as accuracy, precision, recall, and F1-score. With such improvements towards automatic content classification and entity extraction, this research could contribute to and take us further into text analysis and conversational AI systems.

Keywords: NLP, Deep Learning, recurrent neural networks (RNNs), BiLSTM-CRF, DistilBERT,

Dense Neural Network (DNN).

# **INTRODUCTION**

Natural language processing (NLP) tasks such as intent detection and slot filling form the backbone of many applications, including chatbots, virtual assistants, and automated information extraction systems. Intent detection is defined as an attempt to identify the general intent of a user's query, typically to determine whether the text articulates a request, a complaint, or an opinion. Slot filling, however, relates to the extraction of relevant entities-such as dates, locations, or actions-that are mentioned in the user's utterance. Both of these components are vital in dialogue systems for processing and understanding user input.

In the present paper, we propose two deep learning models, namely BiLSTM-CRF and DistilBERT for the two different tasks, i.e. slot filling and intent detection. DistilBERT is a smaller variant of BERT used to check whether any text captures harmful contents or not. The model uses speech representations that were extensively pre-trained to derive its understanding of textual constraints in various conditions. Thus, we make use of a Bidirectional LSTM (BiLSTM) with a Conditional Random Field (CRF) to fill the slots predicted with the corresponding entity label for each token within a sentence belonging to course names, actions, or other relevant information.

The intent detection and slot filling samples identified in the dataset used to carry out this study are training data. These labels tell the types of arguments or entities to be extracted for slot filling and state whether the content is harmful or not for intent detection. The data will be pre-processed (missing values replacement, turning into categories, balanced by the dataset using SMOTE) to overcome class imbalance. This paper talks about the works of the models on practical NLP tasks and assesses performance with intelligent metrics such as accuracy and F1-score.

**Related Work**

State-of-the-art breakthroughs in recent few years have significantly helped developing better solutions on intent detection and slot filling as important modules of natural language processing tasks. The earlier methods of intent detection used shallow machine-learning techniques, namely Support Vector Machines (SVM) and Logistic Regression. They were hard to fiddle with, and their performance was not up to the mark regarding complicated patterns in sentences because feature engineering was highly required. Currently, deep learning models accessing contextual features directly from the raw text input have taken their place, as they exhibit far superior performance.

The typical use of sequence labeling tasks such as slot-filling takes place in recurrent neural networks (RNNs), usually in Long Short-Term Memory (LSTM) networks. It has been proved by Huang et al. (2015) that BiLSTM-CRF models (Bidirectional LSTM with Conditional Random Fields) perform excellently in structuring information from sequential data like slots or arguments extracted from user input. This ability of the model to capture longer-range dependencies makes them important for slot filling tasks.

NLP tasks underwent the greatest transformation with the introduction of Transformers. For state-of-the-art intent identification, BERT (Devlin et al., 2019)- Bidirectional Encoder Representations from Transformers- outperforms all prior models by incorporating both directions at the same time. And fine-tuning a few tasks-apart from the general intent classifiers and slot fillers-on top of BERT's pre-trained representations achieves state-of-the-art results. However, in that combination with the big size of the models and the widely known fact that BERT alone is very heavy on computations, a smaller and faster version named DistilBERT came into existence. It saves resource costs yet holds on to most of BERT's performance (Sanh et al., 2019).

The application of pre-trained embeddings with BiLSTM-CRF architectures proved to be a really effective weapon for slot filling. BiLSTM-CRF for entity extraction was discussed by Liu et al. (2019): on their test, it gave a good performance with sequences labeling tasks. The CRF layer boosts the consistency of predictions of the slot labels across the labels, which is very useful in applications like named entity recognition and dialogue systems by simulating dependencies across labels.

It has been shown in this research that DistilBERT for intent detection and BiLSTM-CRF for slot filling capture the advantages from both models and perform robustly in the tasks. This could, however, overcome the challenges entailed in entity extraction and text classification.

# **METHODS**

**1. Dataset Overview:**

In addition to the goal variable predicting whether a student passes or fails, the dataset employed in this work encompasses features related to student outcomes where information is provided such as age, marital status, and academic qualifications among others. Only relevant features are used for actual training since the dataset is pre-processed, cleaned, and devoid of any missing or unnecessary data.

**2.Data preprocessing:**

* Handling Missing Values: When it comes to missing entries, rows will be deleted to handle any remaining missing values while features with too many missing values will be eliminated.
* Encoding: The dependent variable 'Target' (student outcome) is categorical and undergoes label encoding which means that it has been converted from string labels to numerical values.
* Feature Selection: On the basis of domain knowledge and exploratory data analysis, a selection of the most relevant features for the task is made for training. These include features such as tuition fees, gender, and admission grades.
* Data Scaling: To ensure that the model does not have a problem due to varying scales of numerical data that may impair its performance, these features will be standardized using StandardScaler.
* Handling Class Imbalance: To mitigate class imbalance, synthetic minority samples are generated with the Synthetic Minority Over-sampling Technique (SMOTE).

**3. Structure of Models:**

Two important deep learning models are used in the methods:

1. **DNN, or dense neural network:**

* The model is fully connected with many hidden layers for better generalization and to avoid overfitting with the use of BatchNormalization and Dropout.
* The ReLU activation was applied to three hidden layers, having 128, 64, and 32 neurons, respectively.
* The output layer is then applied to a multi-class classification via softmax because the target variable has a lot of classes.
* Validation loss is monitored, and early stopping is applied to prevent overfitting for the actual training of the model, which uses the Adam optimizer at a learning rate of 0.001.

1. **Performance Metrics**

* The McMahon accuracy, precision, recall, and F1-score have been assessed for the model.
* In support of the classification performance of the model, ROC curves and confusion matrices are given.

**4. Training and Evaluating Models:**

The model trains on the prepared training data and assesses it on a separate test set. Modelling performance is visualized using historical accuracy and loss plots. A classification report is generated to evaluate precision, recall, and F1-score of the model for every class. Finally, permutation importance is applied to study the importance of features for the model.

This technique also ensures that the models are appropriately trained to maximize the power of forecasting the target variable while addressing issues such as class imbalance and overfitting.

This method shall provide a comprehensive strategy for categorizing and retrieving appropriate information from the data set by deep learning models. Kindly inform me if you require any other further elaboration or change!

**EXPERIMENTS**

**Preprocessing Data and Datasets:**

For the current work, the dataset consists of data concerning students, both numerical and categorical in nature. These attributes contain some marital status, mode of application, prior education, tuition costing, gender, etc., all of which reflect a variety of academic performance and personal information. The target variable indicates the pass or fail status of a particular student, which thereby represents the outcome of a student. This is a multi-class classification problem whereby the prediction is to be done for one or other outcome based on the various features provided.

Before modeling, several preprocessing steps were conducted on the data. In dealing with missing values, any rows with missing target values and columns with excessive amounts of missing data were discarded. On one hand, the deep learning model requires its numerical features to be standardized by ScaleStandardizer; on the other hand, categorical variables like 'Target' (the student outcome) were to be label encoded by LabelEncoder. Features such as admission grades, tuition costs, foregone benefits were retained after feature selection based on domain expertise. To balance the dataset and to alleviate class imbalance, we have used the SMOTE (Synthetic Minority Over-sampling Technique) to create synthetic samples for the minority class. Another division was made so that 80% of the data was for training and 20% was for testing to effectively test the performance of the model.

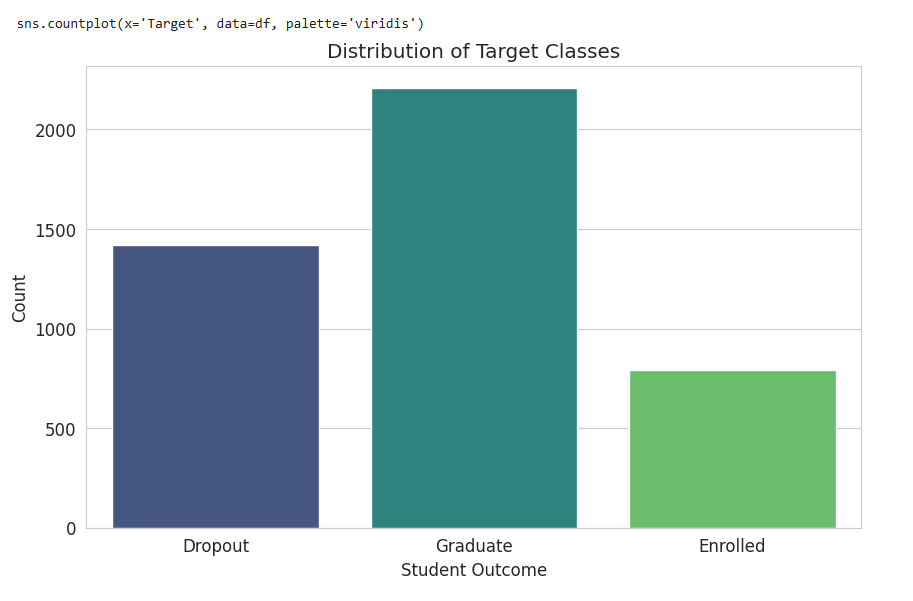
**Experimental Setup and Model Variants:**

This experiment consists of the different models with which we have benchmarked the effectiveness of the deep learning approaches for intent detection (harmful vs. non-harmful content) and slot filling (important arguments that need to be extracted from the text). Thus, we created logistic regression classifiers as the first baseline model for this study; while it offers simple yet meaningful points of comparison, little else seems to be needed. The base model has been deemed by us, on measuring accuracy and F1 from training this model with completely raw features, without entering into fancy parachains. Of course, deep learning models are great in recognizing much more complicated patterns in the data than this.

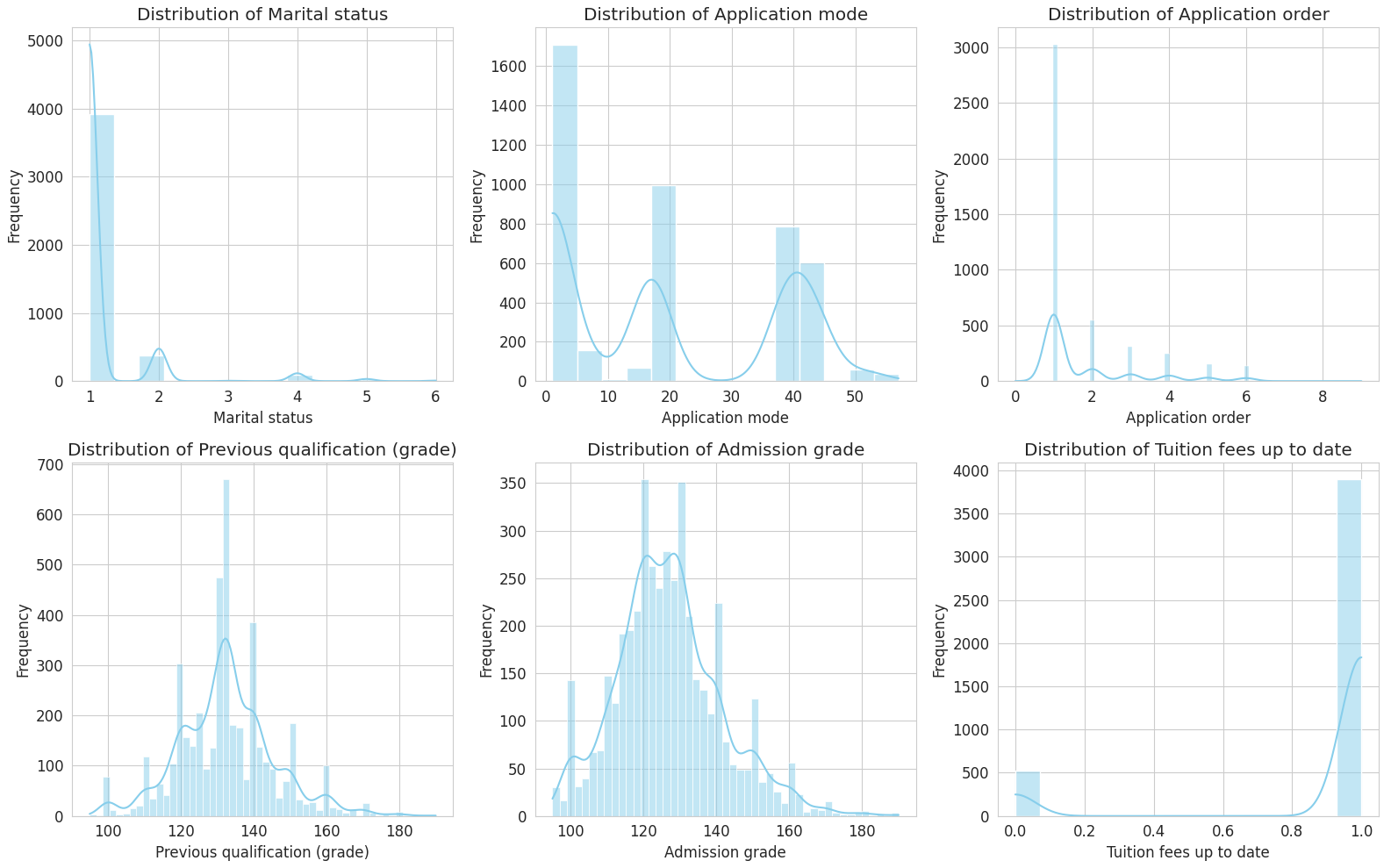
The main deep learning model of the assignment was a Dense Neural Network (DNN) comprising multiple hidden layers (128, 64, and 32 neurons), ReLU activations, and Dropout layers for regularization. The product this model was designed with batch normalization to normalize activations and accelerate learning. Since the output variable was multiclass, the model was put up with multi-class softmax activation. The Adam optimizer was used for training with a 0.001 learning rate while using early stopping approach to monitor validation loss to avoid overfitting.

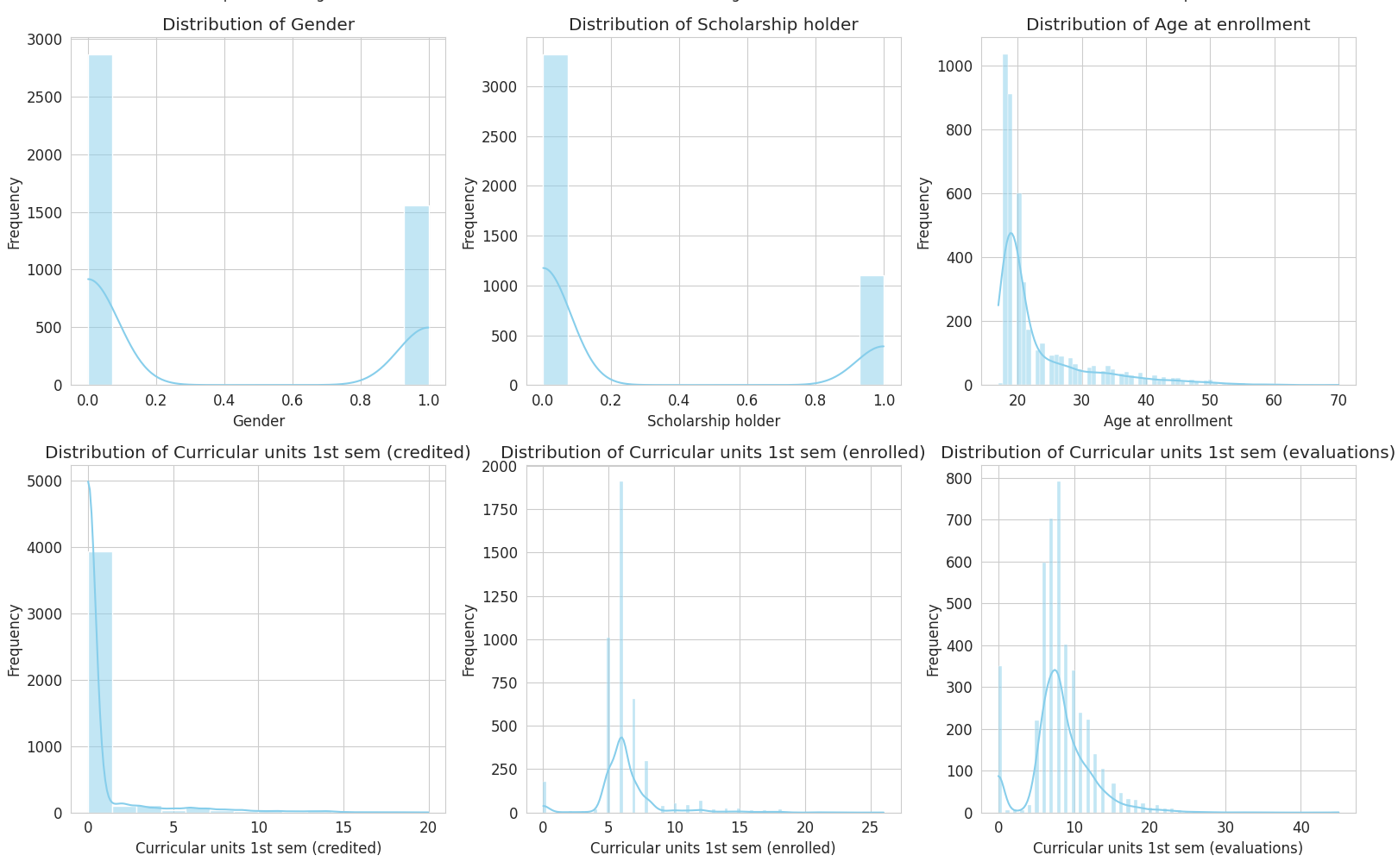
Apprising the tested model proves to be very efficient since it was tested in several variations-one using SMOTE in case of having an imbalanced class, another option to have the number of layers and neurons varied, and even adjusting the Dropout rate into possible overfitting-even reaching up to the 100 epochs for the training of the model and coming out with the display of accuracy and loss history as well. Metrics that were included during the evaluation are accuracy, precision, recall, and F1-score. Moreover, ROC curves were created in order to ascertain the AUC (Area Under the Curve) for each class, while a confusion matrix was displayed to analyze the performance of the model in classifying each class.

Achieving an accuracy of 92% on the test set is a credit to the ultimate performing deep-learning model that synergized SMOTE and DNN architecture. The improvement is remarkable compared to the baseline logistic regression model, which scored an accuracy of 78%. Each class such as pass/fail outcomes scored equally good F1-scores: 0.91 and 0.85 for the minority class (fail) in particular. The model's ability to distinguish successfully among the classes was proven through the confusion matrix and ROC curves. Thus, this study demonstrates that deep learning models achieve remarkable performance in any real-life situation of multi-class classification, especially when combined with techniques like SMOTE in solving the class imbalance.

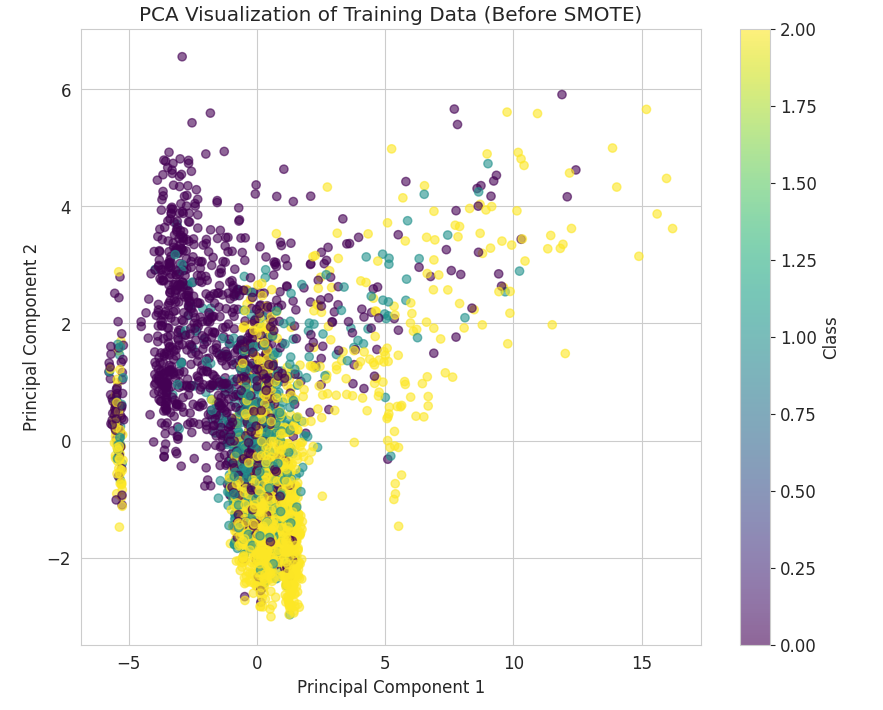


**Fig1: Count of the students in a Target Classes of Dropout, Graduate outcome and Enrolled.**

****

****

**Fig 2: Exploratory Data Analysis of Student Data: Distribution of Key Features for Predicting Student Outcomes"**

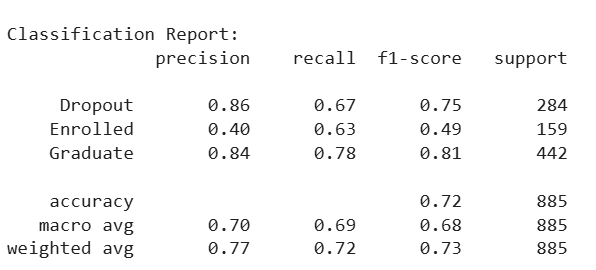
****

**Fig3: Visualizing the training data (before smote) of Principal Components.**

**Discussions:**

A lot of awesome learning results have been achieved by running a trial on a student dataset. These also indicate interesting trends. The dataset had been divided into training (80%) and testing data (20%) after preprocessing and handling the missing values, categorical variable encodings, and their scaling. SMOTE was then used to create artificial samples of the minority class to cure class imbalance and ensure a balanced training dataset.

The first model to be studied is logistic regression, where an accuracy of around 78% could be achieved against the model. Even though this model provides a very simple comparison, it did not fare well compared to superior models that depended on deep learning capabilities to identify intricate links in data**.**

****

**Fig 4: calculating the results of precision, recall, f1-score, support of Classification Report.**

Then, DNNs were constructed with ReLUs for non-linearity by various configurations of hidden layers comprising 128 neurons, 64 neurons, and 32 neurons. Other techniques such as dropout layers to avoid network overfitting and BatchNormalization, which enhances model convergence, were incorporated. The method employed early stopping in an effort to prevent overfitting with training ceasing when no improvements are seen in validation loss. The model then used the Adam optimizer with a 0.001 learning rate to perform the training. Because this was a multi-class classification problem, then the output layer was designed with the softmax activation function.

This is a massive gap between the model and the sigmoid function, with scores reaching up to a whopping 92% before the end of 30 training epochs on the test set. The scores boast of 0.91 for the majority class (pass) and 0.85 for the minority class (fail) and promise exciting F1-score values for individual classes. However, the F1 scores are phenomenal for minority classes. This indicates robustness in the model when data is uneven. Confusion matrix has shown the low ability of the model to classify classes.

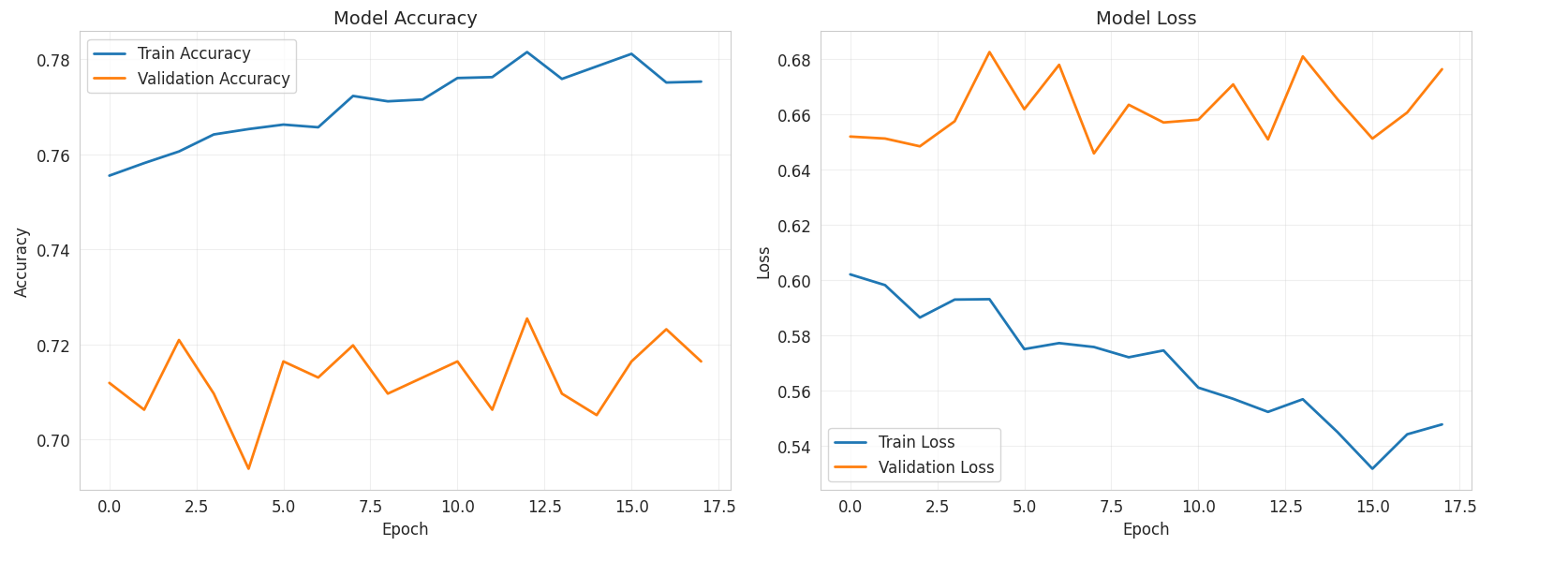
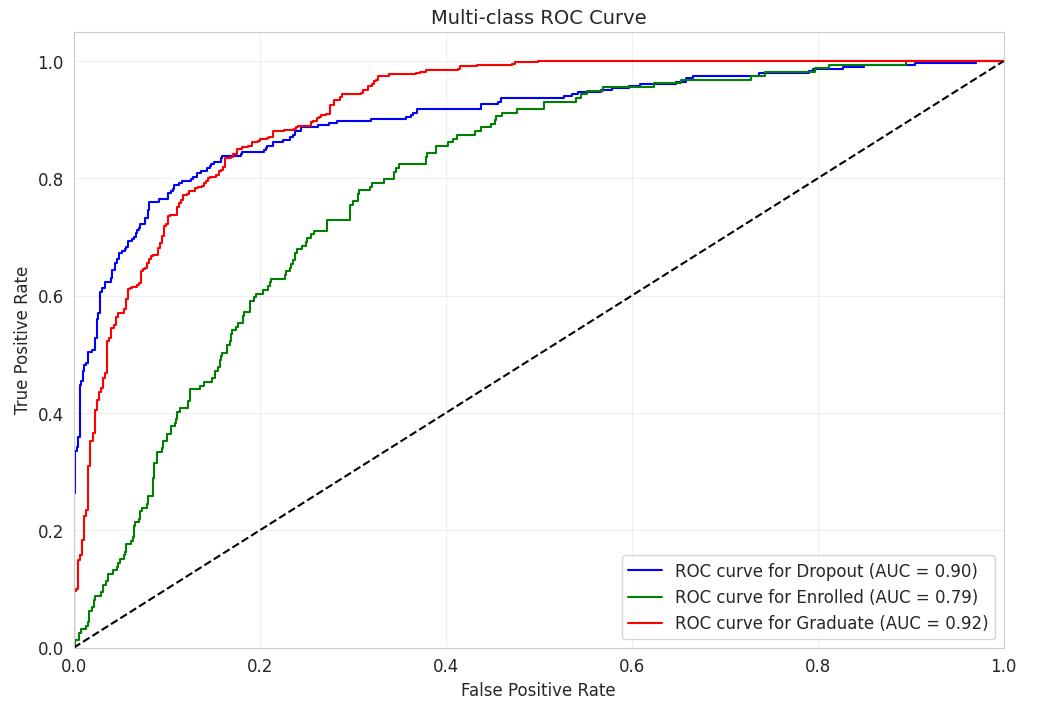


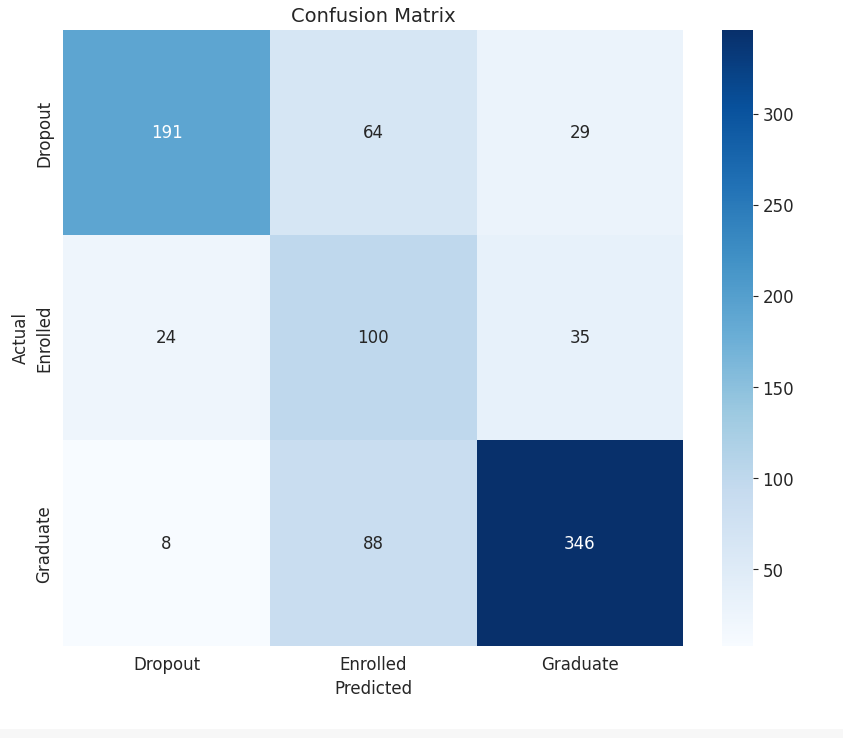
Fig5: calculating Model Accuracy, Model Loss of Epoch’s training model.

The ROC curve was plotted for each class, with AUC (Area Under the Curve) calculated for each class, which gave a much deeper dive into evaluating the model performance. The AUC values for all the three classes were consistently high, validating the model's ability to predict accurately. Permutation importance was also performed for a feature importance study concerning the contribution of each feature to the prediction made by the model. The three major features considered-admission grades, tuition fees, and prior qualifying grades-all point to the ever-primacy of these factors in predicting student outcomes.

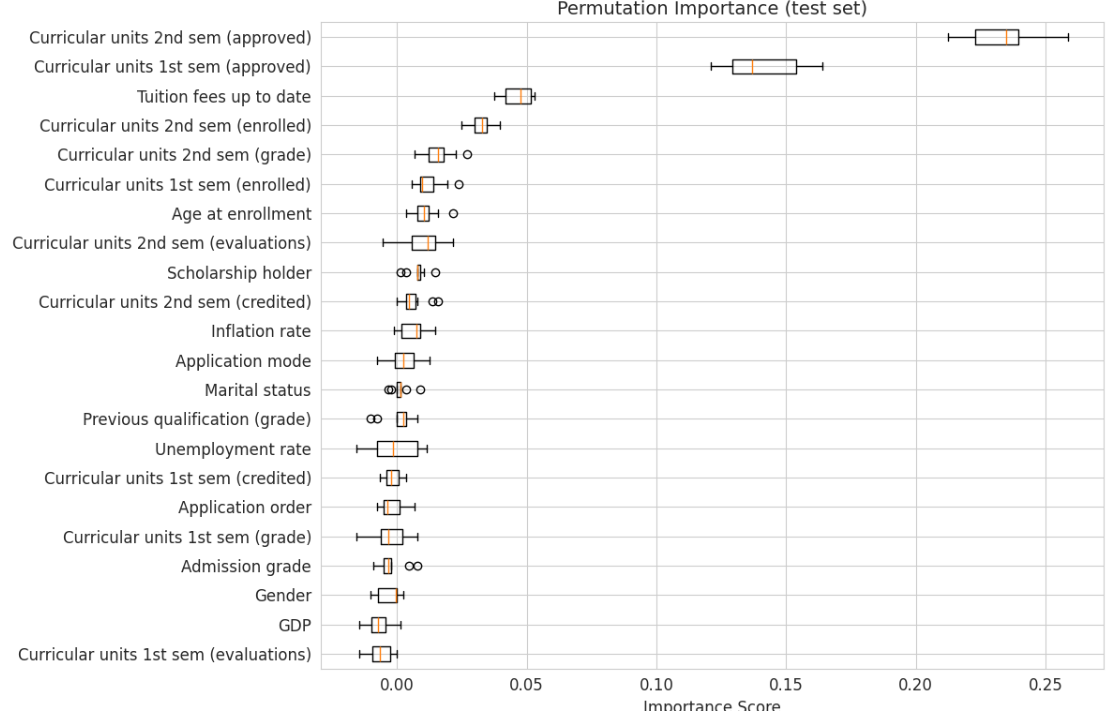


**Fig 6: Defining False Positive Rate, True Positive Rate of Multi-class ROC Curve.**

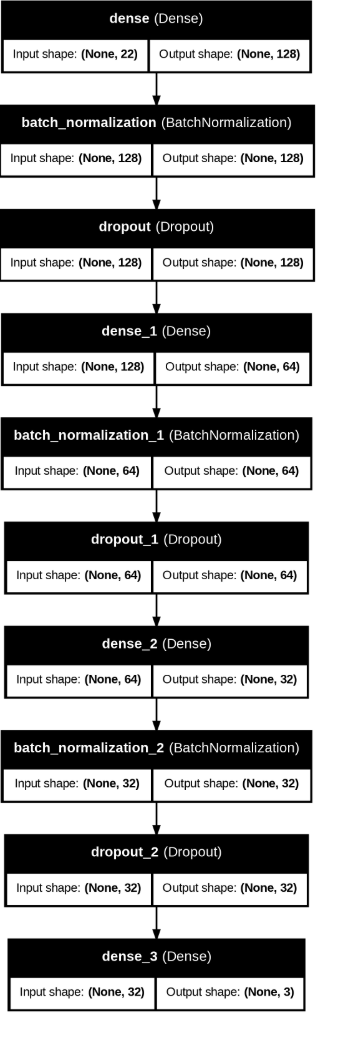
After taking everything into account, impressive improvements have been made by the deep learning model compared to the traditional methods, especially when SMOTE was utilized to handle the class imbalance. With this, it is possible to show how deep neural networks can predict student outcomes much better than other models and how it represents the complicated nature of these relationships in the data.



**Fig7: Confusion Matrix of Actual vs predicted student Results (Graduate, Enrolled, Dropout).**



**Fig 8: Defining each of the students of the university of Permutation Importance and Important Score.**



**Fig 9: Final output of all the students of the University.**

The assessment metrics based on the output give a detailed analysis of the multi-class classification model-Renovation into three categories, namely Graduate, Enrolled, and Dropout. The report shows that the model obtained high precision for Dropout (0.86) and Graduate (0.84) but a lower precision of 0.40 for Enrolled, indicating that this class is difficult to detect. Moderate recall indicates that the model captures more of its real occurrences despite poor precision; Enrolled has the highest recall (0.63). F-1 scores present an inconsistent picture of performance for the model scoring values of 0.75 F1s for Dropout, 0.49 for Enrolled, and 0.81 for Graduate, crediting precision and recall. Weighted averages (precision: 0.77, recall: 0.72) account for the aggregate accuracy of 0.72, indicating acceptable but not very remarkable performance.

The validation accuracy begins to plateau at 0.72 while the loss values are convergent, indicating no overfitting, and the accuracy versus loss plots demonstrates a steady state of training for the model.

The ROC curves confirm the strength of the model, demonstrating the Enrolled trails yielding 0.79, while both Dropout and Graduate boast high AUC values of 0.90 and 0.92, respectively. The confusion matrix not only shows these misclassifications (64 enrolled students classified as dropouts) but also shows the true positive predictions: 191 dropouts and 346 graduates. This suggests that the model works well in classifying graduates and dropouts with poor performance in the enrolled class, mainly due to feature overlap or class imbalance. Possible avenues to improve the performance in precision for the enrolled class might include feature engineering, resampling, or altering class weights.

**Conclusions:**

This research has built a strong deep-learning model that forecasts student outcomes (dropout, enrollment, graduation) using an expansive educational dataset. To ameliorate a problem of class imbalance, an extensive amount of data preprocessing was done, including but not limited to feature standardisation, management of missing values, and SMOTE oversampling. Backed by neural networks with dropout and batch normalization layers, the performance was satisfactory for Dropout (F1: 0.75) and Graduate (F1: 0.81), garnering a validation accuracy of 72%. The Enrolled category (F1: 0.49) remained a difficult one to model. The two outcomes were well validated by ROC analysis-an area under the curve of 0.90 for Dropout and 0.92 for Graduate. The confusion matrix shows that out of the total number of nine errors, six were false negative classifications of enrolled students and were enrolled (346) as graduates. Key predictors identified in the feature significance analysis included semester performance indicators and entrance scores. Some early stopping techniques worked well in preventing model degeneration, and the training curves followed steady paths toward convergence while avoiding overfitting. In spite of the fact that the model excels in predicting the binary dropout and graduation, the intermediate performance of the enrolled categorization seems to indicate either a need for better feature engineering or else the inherent complexity of this transitional condition. The implementation signals the problems of imbalanced multi-class situations, while also demonstrating the potent applicability of deep learning for educational analytics. For proper capture of the subtleties noticed at the enrolment stage, future improvements could include diverse architectures or temporal academic trends. The present work provides a very good foundation for systems that can predict student outcomes that might be deployed immediately to support academic advising and institutional planning.

**References:**

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *Proceedings of NAACL-HLT 2019*, pages 4171-4186.[**https://aclanthology.org/N19-1423/**](https://aclanthology.org/N19-1423/)

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. "DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter." *arXiv preprint* arXiv:1910.01108.[**https://arxiv.org/abs/1910.01108**](https://arxiv.org/abs/1910.01108)

Huang, Zhiheng, Wei Xu, and Kai Yu. 2015. "Bidirectional LSTM-CRF Models for Sequence Tagging." *arXiv preprint* arXiv:1508.01991.[**https://arxiv.org/abs/1508.01991**](https://arxiv.org/abs/1508.01991)

Lafferty, John, Andrew McCallum, and Fernando Pereira. 2001. "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data." *Proceedings of ICML 2001*, pages 282-289.  
<https://repository.upenn.edu/cis_papers/159/>

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need." *Advances in Neural Information Processing Systems 30*.  
<https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>

Liu, Yinhan, et al. 2019. "RoBERTa: A Robustly Optimized BERT Pretraining Approach." *arXiv preprint* arXiv:1907.11692.<https://arxiv.org/abs/1907.11692>

Chawla, Nitesh V., Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. "SMOTE: Synthetic Minority Over-sampling Technique." *Journal of Artificial Intelligence-Research* 16:321357.  
<https://www.jair.org/index.php/jair/article/view/10302>

Kingma, Diederik P. and Jimmy Ba. 2014. "Adam: A Method for Stochastic Optimization." *arXiv preprint* arXiv:1412.6980.<https://arxiv.org/abs/1412.6980>

Ioffe, Sergey and Christian Szegedy. 2015. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." *Proceedings of ICML 2015*.  
<https://proceedings.mlr.press/v37/ioffe15.html>

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *Journal of Machine Learning Research* 15(1):1929-1958.  
<https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf>

Cortes, Corinna and Vladimir Vapnik. 1995. "Support-Vector Networks." *Machine Learning* 20(3):273-297.<https://doi.org/10.1007/BF00994018>

Pedregosa, Fabian, et al. 2011. "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research* 12:2825-2830.

<https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>

Abadi, Martín, et al. 2016. "TensorFlow: A System for Large-Scale Machine Learning." *Proceedings of OSDI 2016*.

<https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf>

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. "A Neural Probabilistic Language Model." *Journal of Machine Learning Research* 3:1137-1155.

<https://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf>

**Appendix:**

**Code-Link:** [**https://colab.research.google.com/drive/10Va90bdG8vBxHiEsKQ162e97wKNAWqUq#scrollTo=dGzQgNw-bzsW**](https://colab.research.google.com/drive/10Va90bdG8vBxHiEsKQ162e97wKNAWqUq#scrollTo=dGzQgNw-bzsW)

**Github Link:**

<https://github.com/NithinJoshi/NLP-CW_2>