Assignment Report

Student Dropout and Academic Success Prediction using Machine Learning

Module Code:
Convenor Name:
Student Name:
Student Number:
Date: 20 Feb, 2024

Actual Hours Spent: 40+ hrs (I work in 2 hour sessions, and take 5 min break every 30 mins. I needed 19 such sessions, and some extra work in between.)

Abstract

This report presents a predictive comprehensive analysis of student dropouts and academic success in higher education institutions via machine learning (ML). Utilizing a dataset from the UCI Machine Learning Repository by ***Realinho et al. (2021)***, I applied three distinct machine learning models: Logistic Regression, Random Forest Classifier, and a Deep Learning model implemented with TensorFlow. The study aims to proactively identify students at risk of dropping out, allowing for timely intervention strategies.

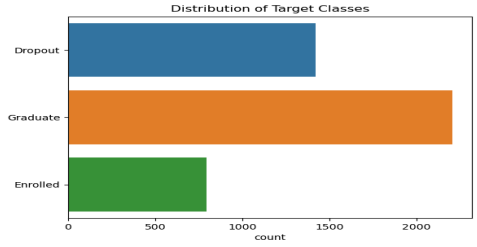
Background and Problem Addressed

The challenge in the educational sector to predict student dropouts and successes is critical. Proactive identification of at-risk students can lead to timely interventions, improving educational outcomes. This insight is not just a matter of statistical interest but a tool for informed decision-making, policy formulation, and resource allocation for the university authorities and student counselors. By identifying students at risk of dropping out, institutions can come up with strategies, allocate support resources more effectively, and ultimately foster a more inclusive and successful educational environment.

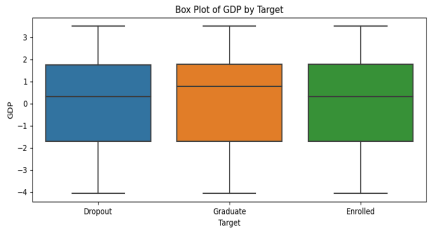
This study leverages machine learning technologies to analyze factors (demographic, academic, and socio-economic) to address these challenges, contributing to the burgeoning field of educational data mining. I hope that my model can help universities make better decisions so students can have the best possible chances of getting an education.

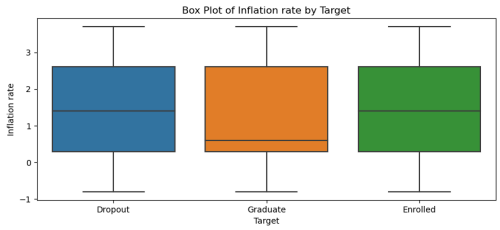
Exploratory Data Analysis

The dataset comprises **4424 instances with 36 features**, including demographic, academic, and socio-economic factors. Initial explorations reveal a balanced representation of features like Marital Status, Gender, and others, with insights gained from distribution plots and correlation analyses.

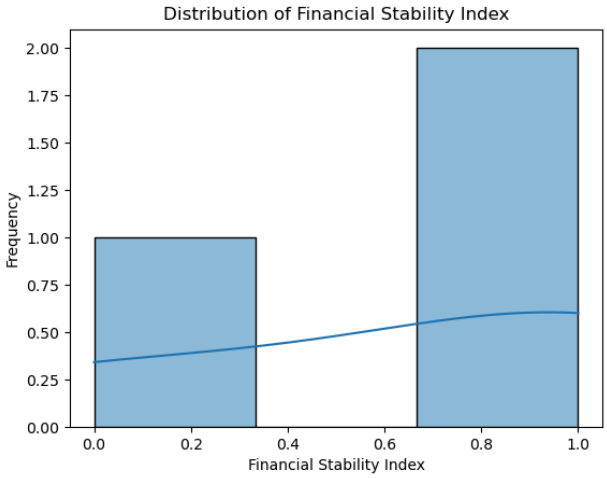


This figure shows the imbalanced nature of our dataset, and as we can see, the Enrolled class has the least example, and our models have a harder time correctly finding this class.

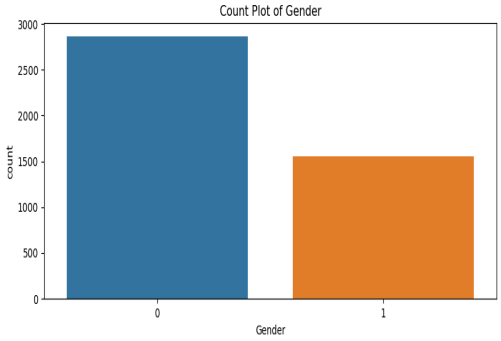


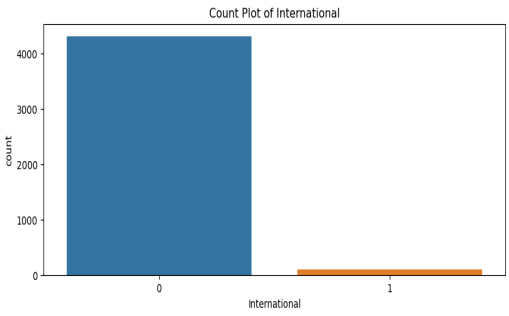


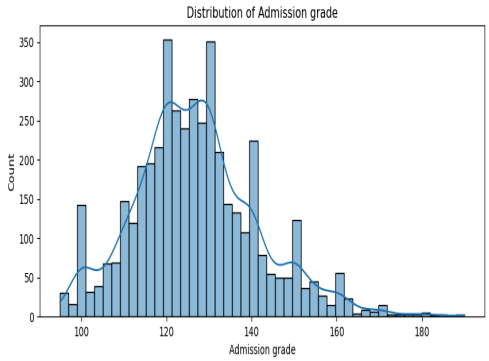
These figures show that current economic factors (GDP, Inflation) affect students' decisions. Then, we developed a feature that condenses these factors.We called it the *Financial Stability Index*.

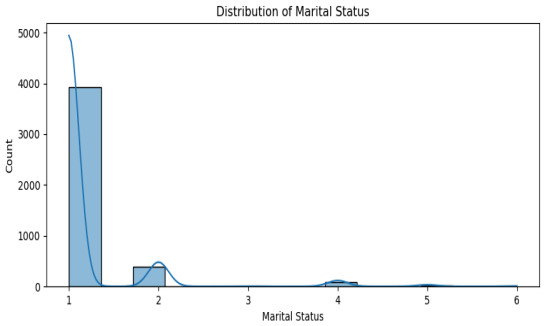


These figures show the imbalance of gender, international/national students, marital status and the admission grades. ***Lenis, Wong. (2023), and Ioanna et al. (2009)***

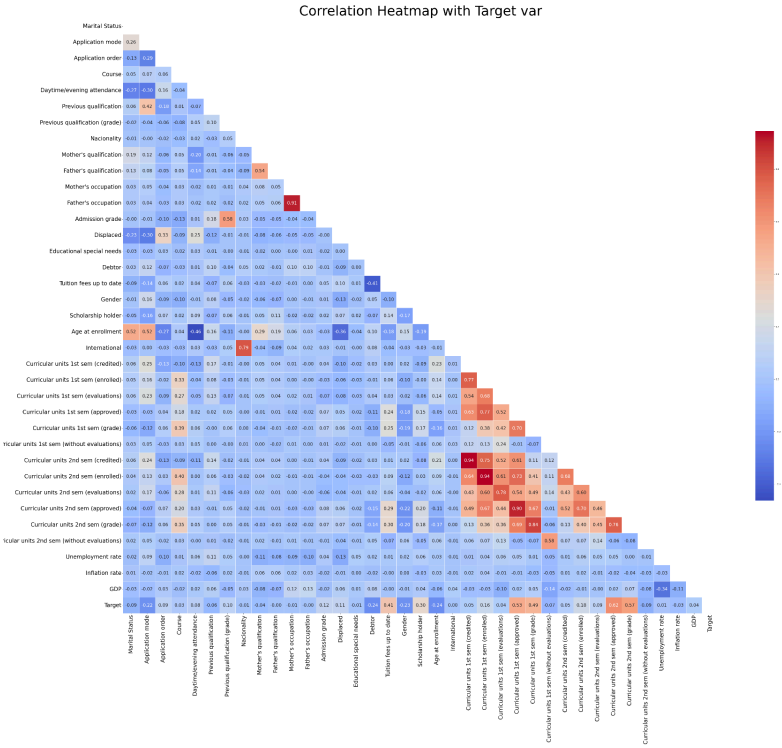








The Heatmap Below shows which factors are Correlated to the target.



Data Pre-processing and Feature Selection

- Initially, I transformed all categorical variables were using one-hot encoding to convert them into a format that machine learning models can interpret more effectively.

- For continuous variables, I normalized to bring all variables to a common scale, eliminating any bias that might arise from varying scales.

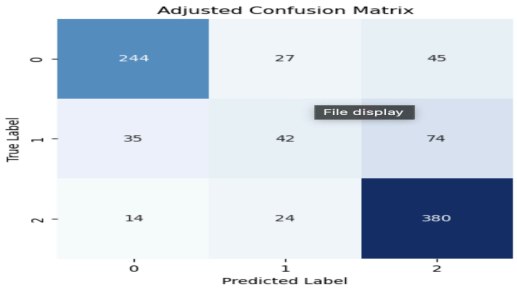
- I checked the dataset for missing values, and, found none, so, I proceeded without the need for imputation strategies.

- Feature Engineering : There were 36 features, and some were correlated to each other. So, instead of letting the model figure out the overlapping pattern, I introduced some new features such as 'Approval Rate' and 'Financial Stability Index'. The details are in my jupyter notebook. They were improved over time as I built the models.

Machine learning models

Model 1: Logistic Regression

- **Summary:** It was the easiest model to build, so it was my first choice as a baseline model. To my surprise, it did pretty well compared to some other complex models.
- **Parameters:** max_iter=500.
- **Rationale for using the Model:** Selected for its simplicity and interpretability.
- **Training and Evaluation:** Achieved an accuracy of 75.25%, with a weighted precision of 73%.
- **Analysis:** Overall balanced performance but slightly lower compared to other 2 models.



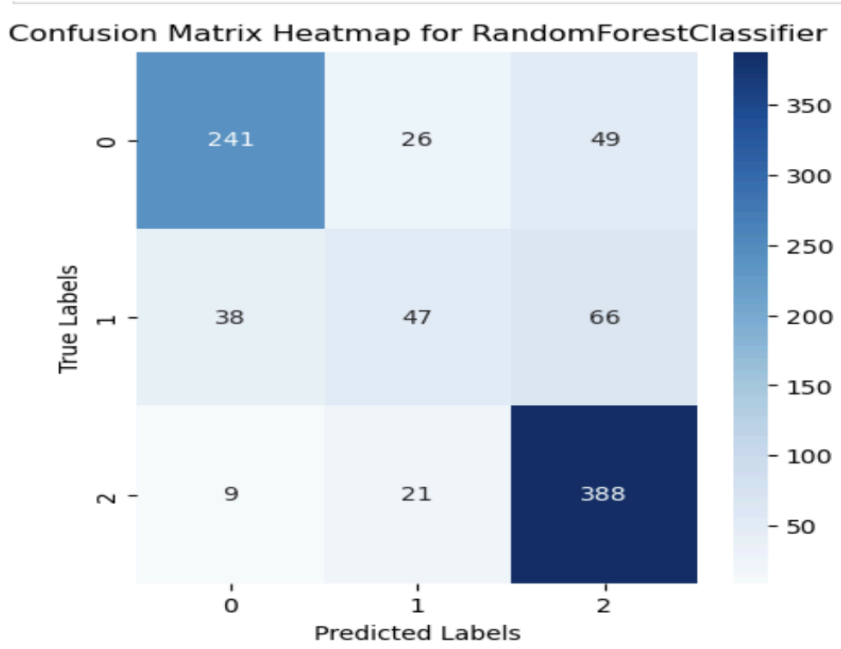

```
Adjusted Accuracy :  0.7525423728813356
Precision :  0.7340833505891269
Recall :  0.752542372881356
F1-Score :  0.7363081777186452
ROC-AUC Score: 0.8694260862515525
Classification Report :
              precision    recall  f1-score   support

     Dropout       0.83      0.77      0.80       316
    Enrolled       0.45      0.28      0.34       151
    Graduate       0.76      0.91      0.83       418

    accuracy                           0.75       885
   macro avg       0.68      0.65      0.66       885
weighted avg       0.73      0.75      0.74       885
```

Model 2: Random Forest Classifier

- **Summary:** I made a Decision Trees model, and although very easy to build, it showed promising results, so I finally did a RFC model for its robustness against overfitting and ability to capture nonlinear relationships. Results similar to Logistic Regression indicate that the data aren't nonlinear in nature ***Leo, Breiman. (2001).***
- **Parameters:** Used default parameters.
- **Rationale for using the Model:** Robustness against overfitting, nonlinear pattern capturing capabilities.
- **Training and Evaluation:** Achieved the highest ROC-AUC score of 0.8714, indicating superior class differentiation.
- **Analysis:** The best overall performance. Some details here -

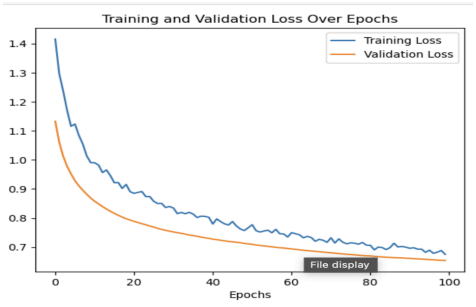


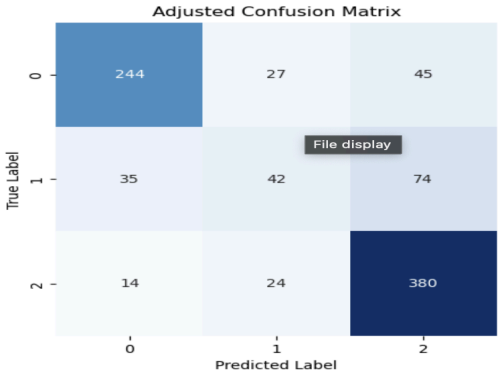

```
Random Forest Accuracy: 0.7638418079096045
Random Forest Precision: 0.7484338476360919
Random Forest Recall: 0.7638418079096045
Random Forest F1-Score: 0.7483592948498395
Random Forest ROC-AUC Score: 0.8712373920524056
Random Forest Classification Report:
                precision    recall  f1-score   support

     Dropout       0.84      0.76      0.80       316
    Enrolled       0.50      0.31      0.38       151
    Graduate       0.77      0.93      0.84       418

    accuracy                           0.76       885
   macro avg       0.70      0.67      0.67       885
weighted avg       0.75      0.76      0.75       885
```

Model 3: Deep Learning with TensorFlow

- **Summary:** The model's architecture, focused on scalability, incorporates a sequential layout with multiple layers, each designed to progressively refine the learning from the data.
    - Structure : A Sequential model is employed, a common choice for a stack of layers where each layer has exactly one input tensor and one output tensor.
    - Layers : Dense layers form the core of this model. These layers are fully connected, meaning each neuron in a layer receives input from all neurons of the previous layer, enhancing the model's ability to learn complex patterns.
    - Activation Function : I've used the very common 'relu' (rectified linear unit) activation function. It is known for its efficiency and effectiveness and helps in mitigating the vanishing gradient problem, which is crucial for DL.
    - Regularization : I've applied L2 regularization, aiming to prevent overfitting by penalizing large weights. This approach helps the model to learn smaller weights, leading to simpler models that generalize better.
    - Dropout Rate This is to stop overfitting. While training, half of the neurons in a layer are randomly deactivated (dropout rate of 0.5 implies that)
- **Parameters:** Sequential model with Dense layers, 'relu' activation, L2 regularization, and dropout rate of 0.5, Learning rate = 0.0001
- **Rationale for using the Model:** Adopted for its capability to model complex relationships.
- **Training and Evaluation:** The model has shown performance comparable to a Random Forest model, an encouraging sign given Random Forest's reputation for robustness. Achieving a ROC-AUC of 0.8663 is notable. This metric indicates good discriminatory ability.
- **Analysis:** I believe that with some proper tuning, maybe a different architecture, or hyperparameter readjustment could help DL models outperform the other ones. Here are the current results that I could push the model to (so far) -
- 




```
Adjusted Accuracy :  0.7525423728813356
Precision :  0.7340833505891269
Recall :  0.7525423728813356
F1-Score :  0.7363081777186452
ROC-AUC Score: 0.8694260862515525
Classification Report :
              precision    recall  f1-score   support

     Dropout       0.83      0.77      0.80       316
    Enrolled       0.45      0.28      0.34       151
    Graduate       0.76      0.91      0.83       418

    accuracy                           0.75       885
   macro avg       0.68      0.65      0.66       885
weighted avg       0.73      0.75      0.74       885
```

Performance Measures and Evaluation Strategies

I've embraced a multifaceted approach to evaluate their performance. One of my go-to tools is the **ROC-AUC Score**, which brilliantly showcases a model's knack for distinguishing between various classes. To get a well-rounded view, I also lean on **Accuracy**, **Precision**, **Recall**, and the **F1-Score**. They provide a comprehensive and balanced view of how well my models are performing. Then there's the **Confusion Matrix** which offers insights into the classification accuracy across different categories.

Model Limitations

- My Limitation : As I am new to building ML models, I know there is a lot to learn and these models can be significantly improved by seasoned professionals. I've tried my share of tuning and feature engineering, But I know it can be made better.

- Potential for Bias : if the data reflects historical biases, the models may inadvertently perpetuate these. For instance, if certain demographic groups historically had higher dropout rates, the model might unfairly flag students from these groups as at-risk.

- Data Quality : Any missing nuances or unrecorded variables that significantly impact student success could skew predictions.

- While the models excel in identifying patterns within the dataset, they cannot ascertain causation, an important consideration when interpreting the results.

Results Comparison and Analysis

For this classification problem at hand, I crafted 8 models, each with its own pros and cons. I zeroed in on the top 3 for a more detailed study, finding that these models had a special knack for handling unseen test data. It was a close call, but the **Random Forest Classification** stood out, not just in ROC-AUC and accuracy but across various metrics.

Deep Learning, with its complex neural networks, hinted at a vast potential yet to be fully tapped. **Logistic Regression**, though a bit more modest in performance, still held its ground with admirable effectiveness. These 3 models were very close to each other. Check the confusion matrices.

Gradient Boosting Machines (GBM) and **Support Vector Machines (SVM)** were powerful and accurate. **The Naive Bayes Classifier**, with its simplicity, was surprisingly quick and accurate. **K-Nearest Neighbors (KNN)** and **Decision Tree**s were reliable and straightforward. All of these models show accuracy near **65-76%**, and ROC-AUC scores between **0.78** to **0.87**.

**However, every single model lacks in identifying 'Enrolled' classes. I knew this would be the case given the imbalanced nature of the dataset, but, even after working and building so many different models and different feature engineering, I couldn't improve it.*

Conclusion, Recommendations, and Future Work

Throughout this assignment, I've worked with lots of ML models for multiclass classification techniques and it's clear that ML is a powerful tool for predicting student outcomes in educational settings. Each model brought something unique to the table, with the Random Forest Classifier emerging as the most balanced one. For future works -

- **Further Feature Engineering:** Looking ahead, I'm excited to do an in-depth Analysis of feature impacts and model behaviors. We need better Feature Engineering to give the models a sharper edge.
- **Over/Undersampling for Class Imbalance:** *All of our models suffered in predicting the 'Enrolled' category. As expected from an imbalance dataset* ***Tahira et al. (2019).*** *I have tried weighted variations, but that didn't work very well for me. So, In future, I'd try sampling techniques, and explore and compare different techniques like Synthetic Minority Over-sampling Technique (SMOTE) or Adaptive Synthetic Sampling (ADASYN)* ***He et al. (2008)***

- **Model Tuning and Ensemble Methods:** A mix of hyperparameter tuning and Ensemble Methods could be the secret sauce for even better performance.
- **Advanced Model:** Something like Long Short-Term Memory (LSTM) networks could adeptly handle the sequential nature of academic data. ***Hochreiter et al. (1997)***
- **Cross-Validation Strategy:** To strengthen the robustness and interpretability of the models.
- **Ethical Considerations:** It's always important to consider the ethical implications of predictive modeling. I have to ensure fairness and make sure that my model avoids biases that could adversely affect certain student groups before they are put to use in the real world.

- **Practical Implications :** We can develop targeted support systems for students at risk, optimize our resources, and cater our educational policies to enhance student retention and success. With insights from these models, institutions can improve their academic performances and also foster a more supportive and equitable learning system.

References

1. Realinho,Valentim, Vieira Martins,Mónica, Machado,Jorge, and Baptista,Luís. (2021). Predict students' dropout and academic success. UCI Machine Learning Repository. https://doi.org/10.24432/C5MC89.
2. Leo, Breiman. (2001). Random Forests. 45(1):5-32. doi: 10.1023/A:1010933404324
3. Lenis, Wong. (2023). Model for the Prediction of Dropout in Higher Education in Peru applying Machine Learning Algorithms: Random Forest, Decision Tree, Neural Network and Support Vector Machine. 116-124. doi: 10.23919/FRUCT58615.2023.10143068
4. Ioanna, Lykourentzou., Ioannis, Giannoukos., Vassilis, Nikolopoulos., George, Mpardis., Vassili, Loumos. (2009). Dropout prediction in e-learning courses through the combination of machine learning techniques. Computer Education, 53(3):950-965. doi: 10.1016/J.COMPEDU.2009.05.010
5. Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.
6. Tahira, Alam., Chowdhury, Farhan, Ahmed., Sabit, Anwar, Zahin., Muhammad, Asif, Hossain, Khan., Maliha, Tashfia, Islam. (2019). An Effective Recursive Technique for Multi-Class Classification and Regression for Imbalanced Data. IEEE Access, 7:127615-127630. doi: 10.1109/ACCESS.2019.2939755
7. He, Haibo & Bai, Yang & Garcia, Edwardo & Li, Shutao. (2008). ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. Proceedings of the International Joint Conference on Neural Networks. 1322 - 1328. 10.1109/IJCNN.2008.4633969.