Forecasting Student Success through Artificial Intelligence: A Case Study Using Neural Networks

Dongli Liu[1], Wendy Paraizo[1], Daniel Ifejika[1]

[1] Progress Campus of Centennial College. 941 Progress Ave, Scarborough, ON M1G 3T8 CA

Abstract

Accurately forecasting student success is crucial for improving retention and outcomes. This study predicts first-year persistence using Artificial Intelligence (AI) and Neural Networks (NN). A real-world dataset is processed to address challenges like missing values and class imbalance. The NN model, optimized with advanced techniques, outperforms traditional methods in predicting persistence. While the results show promise, the study also highlights challenges in model interpretability and generalization, suggesting areas for future exploration in AI-driven educational interventions.

Key words: Artificial Intelligence; Neural Network; Student Success; Predictive Analytics.

Introduction

Student success is a critical concern for educational institutions, influencing graduation rates, career outcomes, and societal contributions. However, predicting student success is complex, as factors such as socio-economic background, prior academic performance, and personal circumstances can significantly affect outcomes. Institutions face challenges in identifying students at risk of underperforming or dropping out, making it essential to develop methods for early intervention.

Programs like the Helping Youth Pursue Education (HYPE)[1] initiative at Centennial College have demonstrated the importance of targeted support for students from underserved communities [2]. HYPE's focus on addressing barriers to post-secondary education for youth from disadvantaged backgrounds highlights the need for data-driven strategies to improve retention and success rates in higher education [1].

Artificial Intelligence (AI), particularly Neural Networks (NN), offers promising solutions in education. Neural networks can analyze large datasets, uncovering complex patterns that traditional methods may miss. This study uses AI to predict first-year persistence, a key indicator of student success, using a real-world dataset. The goal is to demonstrate how neural networks can improve prediction accuracy and provide insights for better supporting students, while addressing challenges such as missing data, class imbalance, and insufficient amount of data.

Methodology

Dataset Preprocessing

The dataset used in this study originates from Centennial College, specifically from engineering school, and contains data on first-year students, including socio-economic status, high school GPA, attendance records, participation in extracurricular activities, and other features hypothesized to influence student success, as shown in Table 1.

Column	Non Null	Dtype
First Term Gpa	1420	float64
Second Term Gpa	1277	float64
First Language	1326	float64
Funding	1437	int64
School	1437	int64
Fast Track	1437	int64
Coop	1437	int64
Residency	1437	int64
Gender	1437	int64
Previous Education	1433	float64
Age Group	1433	float64
High School Avg Mark	694	float64
Math Score	975	float64
English Grade	1392	float64
FirstYearPersistence	1437	int64

Table 1: Information of the Raw Data

Missing values were handled by applying `dropna` columns with a small portion of null values. For the remaining missing data, `IterativeImputer` with `RandomForestRegressor()` was used to infer values, preserving feature relationships. This method was particularly important for columns like *High School Average Marks*, which had over 50% missing values, as using mean imputation could have introduced significant bias [3].

```
imputer = IterativeImputer(
    estimator=RandomForestRegressor(),
    max_iter=10,
    n_nearest_features=6,
    imputation_order="ascending",
    tol=1,
    random_state=seed)
imputed = imputer.fit_transform(dropped)
```

Listing 1: `sklearn.impute.IterateImputer` is a experimental class that infers values by iteratively modelling a function with other features. It is very useful for *multivariate* problems. [3]

The data is significantly imbalanced, with 1138 positive while only 299 negative instances.To address **class imbalance**, *upsampling* was used instead of *downsampling* to retain valuable information [3]. This approach was also chosen because the dataset is relatively small, and maintaining as much data as possible is crucial for training an effective model.

Input Pipeline

The `tf.data.Dataset` API was utilized to create a descriptive and efficient input pipeline [4]. While the dataset used in this project is not large, the choice to use this API aligns with practical considerations, such as the potential deployment of the model for both *prediction* and *online machine training*[5][1].

By leveraging the capabilities of the API `tf.data.Dataset`, an efficient data processing chain was constructed following the approach detailed by A. Géron. The dataset was then split into `train_set`, `val_set`, `test_set` using the `take()` and `skip()` functions provided by the API.

Search Model Structure

Neural Network Architecture The neural network model used in this study is designed to predict first-year persistence. The architecture consists of an input layer, two hidden layers, and an output layer. The input layer is designed

1 a method of machine learning in which data becomes available in a sequential order and is used to update the best predictor for future data at each step.

to accommodate the 30 features in the dataset. The hidden layers consist of 128 neurons in the first layer and 64 neurons in the second, chosen based on initial experiments to balance model complexity and performance. Each layer uses the Rectified Linear Unit (ReLU) activation function, which is known to handle the vanishing gradient problem better than traditional sigmoid functions.

The output layer consists of a single neuron, using the sigmoid activation function to predict the binary outcome of student persistence: 1 for continued enrollment and 0 for non-persistence. Dropout regularization techniques were applied in the hidden layers to prevent overfitting, with a dropout rate of 0.3. The model is trained using the Adam optimizer, which combines the advantages of both AdaGrad and RMSProp for faster convergence and improved performance in sparse gradients.

Experimental Setup The dataset was split into three subsets: training, validation, and testing. The training set comprises 70% of the data, while the validation and testing sets each consist of 15%. The model was trained for 50 epochs, with early stopping implemented to prevent overfitting if the validation loss does not improve after 10 epochs.

During training, the model's performance was evaluated on the validation set to tune hyperparameters such as the learning rate and number of hidden neurons. After training, the model was tested on the unseen test set to evaluate its generalization performance.

Evaluation Metrics The performance of the neural network model was evaluated using a range of metrics that assess both classification accuracy and the ability to correctly identify at-risk students. The following evaluation metrics were used:

Accuracy: The percentage of correctly classified instances in the test set. Precision: The proportion of true positive predictions out of all positive predictions, indicating how well the model identifies students who are at risk. Recall: The proportion of true positive predictions out of all actual positive instances, showing how well the model identifies all students at risk. F1-Score: The harmonic mean of precision and recall, providing a balance between the two. AUC-ROC (Area Under the Receiver Operating Characteristic Curve): A performance measurement for classification problems at various threshold settings, which helps evaluate the model's ability to discriminate between students who persist and those who do not. These metrics were chosen to ensure a comprehensive evaluation of the model's performance, focusing not just on overall accuracy but also on its ability to effectively identify students in need of support.

References

[1] P. Armstrong, H. Jafar, D. Aromiwura, J. Maher, A. Bertin, and H. Zhao, *Helping Youth Pursue Education (HYPE): Exploring the Keys to Transformation in Postsecondary Access and Retention for Youth from Underserved Neighbourhoods*. Toronto: Higher Education Quality Council of Ontario, 2017.

[2] J. Maher and A. Bertin, "Sustaining the Transformation: Improving College Retention and Success Rates for Youth from Underserved Neighbourhoods," *Journal of Global Citizenship & Equity Education*, vol. 3, no. 1, pp. 1–20, 2013, [Online]. Available: https://journals.sfu.ca/jgcee

[3] Scikit-Learn, "Scikit-Learn Documentation." [Online]. Available: https://scikit-learn.org/stable/

[4] Google, “TensorFlow Tutorials and Related Google Documentation.” [Online]. Available: https://www.tensorflow.org/tutorials

[5] Wikipedia, “Online machine learning.” [Online]. Available: https://en.wikipedia.org/wiki/Online_machine_learning

[6] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd ed. O'Reilly Media, Inc., 2022.