DRAFT: A Review of the Retention Analytics Dashboard model

Summary

This paper describes technical details of the predictive model Academic Experience Design & Delivery (AXDD), a unit of UW-IT, developed for the Retention Analytics Dashboard (RAD). The overarching goal is to assist groups on campus who work directly with students to utilize available student experience and academic data to facilitate student success. The modelling described here uses existing UW data to identify students at risk of not meeting a GPA minimum or withdrawing early.

The model is the product of iterative development in conjunction with the Office of Minority Affairs & Diversity (OMAD). It is intended to help inform advisers' interactions with students, not provide a definitive expectation of a student's outcome. Early versions of the model and visualizations were prototyped during 2019 and the current version was revised and deployed in 2020 shortly before the beginning of all-online instruction in spring quarter. We use a simple implementation of a neural network to generate predictions at the beginning of each quarter.

Modeling Approach

After comparison of multiple models we settled on a simple feed-forward neural network implemented in Python via scikit-learn. After tuning, the neural network has advantages for our use case: it is computationally quick to train and subsequently update, it performs well for classification tasks where the data is not linearly separable, and is not sensitive to multi-collinearity. The main trade-off is interpretability. Post-estimation visual diagnostic tools can help with explanation. In our experiments tree models (e.g. CatBoost, xgboost) yielded similar results with more easily interpretable models but at a steep computational cost.

Data Collection

Our model uses data primarily from UW's student database. Two data points are drawn from OMAD's own database, visits to advisers and the Instructional Center (IC). IC visits, which rely on an in-person check-in system, dropped off after the switch to online learning in 2020. Because the model was developed with OMAD the sample is limited to OMAD students.[1] On average this is 4,895 students per quarter since 2006 or 5,834 excluding summer quarters.

- tenth day credits
- attempted credits
- test scores
- number of courses
- non-graded courses
- age
- total points
- GPA
- transfer GPA

[1] Expanded to include Integrated Social Science in Spring 2020.

- high school GPA
- STEM courses
- number of required credits (VLPA, I&S, writing, etc.)
- WA resident
- low family income
- last school type
- size of current courses
- gender
- current major
- STEM credits
- running start
- conditional admit
- high school foreign language
- parent/guardian education
- course fees this term
- class standing
- full-time status
- child of alum
- high-school grad w/ distinction
- high school math level
- number of unmet course requests
- registration date/period

The target variable, “adverse outcome”, is calculated from transcript data. A student is considered to have an adverse outcome for the quarter if they receive a grade <= 2.5 and/or drop or withdraw from any course(s).

Data Cleaning and Transformation

Variables that cannot be observed before the end of a quarter such as GPA are lagged by one term. Additional features were derived from the above data, e.g. cumulative grade points and credits over time, quarter-to-quarter change in GPA, change in major this quarter, multiple majors, credits above full-time load, late/early registration. This results in a set of 71 features before pre-processing. The data are then split 75/25 into a train/test set.

We use a transformation pipeline object to handle imputation and re-encoding steps. This facilitates a clean separation between the train/test data and transformation steps applied to the training set. Experiments with imputation techniques found no benefit to more complex techniques such as ROSE and SMOTE over simple modal and mean imputation for categorical and continuous variables respectively. Because neural networks are sensitive to scaling the numeric features are centered and scaled; one hot encoding is used for categorical features such as class standing.

Model - Multi-layer Perceptron

The model used for generation predictions is a simple neural network using a multi-layer perceptron (MLP) classifier. The initial setup of a neural network is similar to other regression/classification tasks. The model requires a set of input variables and the corresponding output or target variable(s). E.g. the output/target might be the result of a medical test labeled positive and negative while the input data are entries from patients’ medical history.

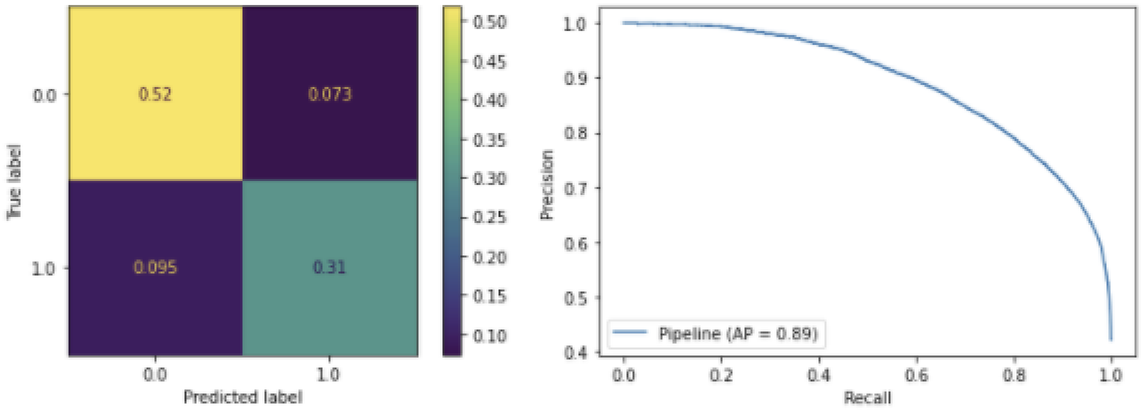
Generally speaking a neural network attempts to iteratively map those inputs to the outputs through “hidden layers” using different “activation functions” - where the number of hidden layers, their size, and the activation function are chosen by the user. Therefore, tuning and optimizing an MLP model typically involves experimentation with different settings for the layers and functions. Note that the hidden layers' weight values can be saved but unlike a regression model they are not usually of interest and are not parameter estimates for the variables.[2] Although there is no hard limit to the number of layers, in practice a small number, two to three, is preferable to avoid overfitting.

The basic unit of the MLP binary classifier is the perceptron which maps input vectors to a single output via a user-defined threshold function. In short, a perceptron attempts to replicate the architecture of a biological neuron by combining inputs, weighting them separately, adding some small bias/error, then passing the summed weights through a nonlinear function to produce an output value. The perceptron function, *f(x)*, outputs a positive prediction where $w \cdot x + b > 0$ and a negative otherwise, where:

- *w* represents the weight vector
- *b* is the bias term (a small adjustment that shifts the curve away from the origin)[3]
- *x* is a vector of input values

This basic setup can be easily generalized to the sum of $w_i x_i$ over multiple inputs and layers. In our case, the network is fully connected - every neuron in a layer contains a forward link to every neuron in the next layer. There are several common activation functions; our model uses the logistic activation which maps the values to a smooth sigmoid curve. The MLP uses backpropagation[4] to optimize the weights over successive iterations.

After testing different configurations of hidden layers and activation functions we selected a logistic activation with three hidden layers of 200 nodes. We use an alpha of .0001 with early stopping enabled to limit overfitting. These settings result in a good balance between precision and recall.



Training set diagnostics for multilayer perceptron model

[2] For those familiar with logistic regression a neural network with no hidden layers using a logistic activation reduces to the linear logistic regression model.
[3] Weights affect the shape of the sigmoid curve, bias offsets the x-intercept.
[4] Backward error propagation involves calculating the gradient of the error with respect to each input-output set. The weights are then updated in the inverse direction of the total gradient according to a user-defined learning rate.

Deployment

Our model is run and updated quarterly. Shortly before each quarter we refresh the dataset and create new, unseen data for the upcoming term's predictions. After re-estimating the model with updated historical data we use the stored model object to generate predicted values. Rather than determine a threshold for true/false predictions our display uses the model's estimated probabilities, rescaled from -5 to +5. Rescaling provides consistency with other data displayed in the dashboard.[5]

Our most recent evaluation of the dashboard and model found that a prediction score below zero identified ⅔ of students who had at least D/F/W. A score below zero also captured 75% of students who had received a grade of 2.5 or lower in three or more courses.

Last updated June, 2021

[5] The predictions are intended for use in conjunction with other data in the dashboard which includes participation, grades, and assignment status from the Canvas LMS.