Interpretable Deep Learning Techniques for University Dropout Prediction

Egyetemi lemorzsolódás előrejelzése értelmezhető mély tanulással

Máté Baranyi*, Tekla Kiss†, Marcell Nagy‡
*‡ Department of Stochastics, Budapest University of Technology and Economics, Hungary
† Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Hungary
*baranyim@math.bme.hu, †kiss.tekla@simonyi.bme.hu, ‡marcessz@math.bme.hu

Abstract—**The early identification of students at risk of dropout is of great interest and importance worldwide, since early leaving of higher education is associated with considerable economic and social costs. In Hungary, and especially regarding STEM undergraduate programs, the dropout rate is above the EU average. In this work, using advanced machine learning models, such as deep neural networks and gradient boosted trees, we aim to predict the final academic performance of students (graduated or dropped-out) based on data that are available at the time of enrollment, that were provided by the Central Academic Office of Budapest University of Technology and Economics. In order to ensure that our uploaded dataset is anonymous, we synthesize data with a Conditional Generative Adversarial Network. Besides making predictions, we also interpret our machine learning models with the help of state-of-the-art techniques such as SHAP values. The accuracy and AUC of the best-performing deep learning model are 74% and 0.72 respectively. That is quite remarkable compared to the ... accuracy and ... AUC scores of XGBoost.**

Kivonat—**Az egész világon megfigyelhető jelenség, hogy a felsőoktatásban tanuló diákok durván egy harmada lemorzsolódik. A lemorzsolódás komoly gazdasági és társadalmi költségekkel jár, így az utóbbi évtizedben egyre több a jelenség okait és folyamatát feltáró publikáció jelent meg. Magyarországon, különösen a műszaki felsőoktatásban, a lemorzsolódott hallgatók aránya az EU-s átlag felett van. A projektfeladatunkban gépi tanulási modelleket(pl. mély neurális hálók és gradient boosted tree modellek) felhasználva azt szeretnénk előrejelezni a felvételkor rendelkezésre álló adatok alapján, hogy ki fogja elvégezni az egyetemet. Az elemzésekhez a Budapesti Műszaki és Gazdaságtudományi Egyetem adatait használjuk, amiket a Központi Tanulmányi Hivatal szolgáltatott. Továbbá az adatok érzékenysége miatt egy tabuláris adatokra kifejlesztett Conditional Generative Adversarial Network modell segítségével szintetikus adatokat is generálunk. Amellett hogy egy becslést adunk a hallgatók végső egyetemi teljesítményéről, azt is megvizsgáljuk, hogy az egyetemi sikeresség vagy lemorzsolódás milyen tényezőktől függ. Ehhez "model interpretability" technikákat alkalmazunk, amelyek képesek az olyan black-box modellek döntéseit is értelmezni, mint a neurális hálók. A legjobb eredményt felmutató deep learning modellünk pontosága és AUC értéke rendre 72% illetve 0.74.**

Index Terms—**higher education, dropout prediction, deep learning, interpretable machine learning**

I. Introduction

Student drop-out and delayed completion are some of the most burning issues of higher education all over the world. Globally roughly every third student drops out from higher education [1]–[4], which is associated with considerable personal and social cost [5], [6]. In Hungary the graduation rate is particularly low, one of the lowest among OECD[1] [7] and EU countries [3], [8].

Understanding and identifying the factors affecting students' university performance, and detecting at-risk students are in the focus of educational research for a few decades [3], [9]–[14]. Moreover, due to the rise of data-driven approaches and the vast amount of data stored in educational administrative systems, a tremendous amount of predictive analytical educational research papers and artificial intelligence based decision support systems have been published in the last few years, for systematic reviews, we refer to [15]–[17].

The majority of the related works use traditional machine learning models (e.g. logistic regression, k-nearest neighbors, and decision tree based ensemble models) for dropout prediction [18]–[21]. Deep neural networks (DNN) are not frequently applied on tabular data, due to the fact that such datasets are usually not large enough for these models to achieve high performance, moreover DNNs require more fine-tuning, expertise, and these models are less interpretable than the traditional ones. However, there are some papers that successfully apply deep neural networks for student performance and dropout prediction [22]–[27].

While numerous predictive analytical educational studies have been published worldwide over the last decade, to the best of our knowledge in the Central European region little to no papers have been conducted regarding this research area. On the other hand, at the Budapest University of Technology and Economics (BME) a small research group was initiated a few years ago in the cooperation of the Institute of Mathematics and the Central Academic Office. This successful cooperation has resulted in a number of research papers on data-driven educational research, in particular on student dropout prediction [28]–[30]. In [30] the authors use traditional machine learning classifiers to predict students' final academic performance, and in a follow-up paper [29], they also introduce a web application as a decision-support tool that not only returns the probability of graduation but shows what are the key factors and to what extent they are affecting the individual predictions.

[1] Organization for Economic Co-operation and Development

Moreover, Kiss et al. [28] investigate the incremental predictive validity of first-semester university performance indicators on graduation over pre-enrollment achievement measures.

This work extends the aforecited papers in several ways. First of all, the main focus of this work is to investigate, whether deep neural networks are able to identify students at risk of dropout. To this end, we compare the performance of DNNs to state-of-the-art tree-based machine learning models such as XGBoost. Furthermore, we aim to apply as many deep learning techniques and tools – that we learned in the course called *Deep Learning in Practice with Python and LUA* – as we can. In particular, we also put great emphasis on hyperparameter optimization, moreover we also generate synthetic data with a conditional generative adversarial network.

Compared to related predictive analytical studies, another key contribution of this work is that we do not only give a prediction of the final academic status, but also use machine learning interpretability and explainability tools such as SHAP values [31], partial dependence plots, and permutation importance. On the first hand, these tools help us understand and interpret the decisions of the models. On the other hand, they also provide global and local explanations of the predictions that can give insights for the decision-makers of higher education, moreover it can also help students find the skills they need to master in order to be successful in higher education, by highlighting the factors that decrease the estimated probability of graduation.

The main objectives of this work can be summarized as follows:

1) Using deep neural networks to predict the students' final academic performance (graduation or dropout) as accurately as possible, and compare their performance to other baseline machine learning models' accuracy such as XGboost and Random Forest.
2) Synthesize data using a conditional GAN, in order to fully anonymize our dataset.
3) Use model explainability and interpretability tools to understand the decisions of the models, moreover to find what are the key factors that affect university success.

The workflow of this project is shown in Figure 1.

II. Background on Hungarian Higher Education

To be able to follow the rest of this paper, in this section, relying on [30], we briefly describe the aspects of the transition to higher education in Hungary.

Secondary education takes place after 8 years of primary education and followed by higher education. At the end of secondary education, students take an exit exam, called *matura*, that is a matriculation exam as well, meaning that it is a university entrance exam at the same time, if the student applied to higher education. Matura consists of (at least) five separate sub-exams of core subjects: mathematics, history, Hungarian language and literature, a foreign language, and (at least) one chosen subject. Note that any sub-exam can be taken at a normal or advanced level.

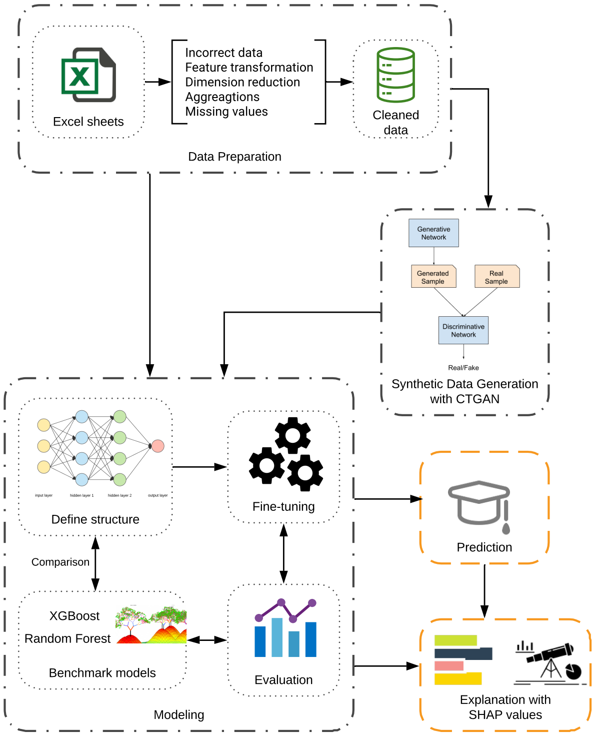

Figure 1. The workflow of this project

The following description of admission to higher education is from [30]. The admission to higher education in Hungary mostly relies on the secondary school performance of the students and in particular the results of their matura exam scores. Students applying to colleges in Hungary gain a University Entrance Score (UES) based on three factors: the sum of grades in secondary school and the average matura sub-exam scores (study points - SP), results of the matura sub-exams from two subjects required by the given university program (matura points - MP) and extra points (EP) for additional achievements (e.g. taking advanced-level matura exams, having certificate of a foreign language, earning a prestigious place in sport, art or academic competitions) and equal opportunity points (having disability, disadvantage, being on child care). Every bachelor's program requires matura exams of specified subjects, thus the aforementioned subject of students' choice may depend on the desired program (e.g. engineering bachelor programs usually require maths and a science subject).

There are two ways to calculate the university entrance score, and the system automatically takes into account the one that is more advantageous for the student. The first way is the *summation* method, that is $UES = SP + MP + EP$, and the other is the *doubling method*: $UES = 2 \cdot MP + EP$. The maximum acquirable admission point score is 500 points.

Each university defines a minimal university entrance score (MUES) to its programs and accepts those students whose UES are greater than the MUES of the desired program.

III. Data Preparation

We have received the data from the Central Academic Office and it consists of 39,048 Hungarian students who enrolled between 2010 and 2018 to an undergraduate program of the Budapest University of Technology and Economics. Our data preparation started with filtering: we excluded the currently *active* students, i.e. we only investigate dropouts and graduates, which means 24,682 rows in the dataset. We also consider students as being graduated if they have completed their required studies, even if they have not obtained a diploma yet (e.g. due to the lack of foreign language certificate). We also excluded 615 students, who changed their major, otherwise they would have been considered as dropouts at their previous undergraduate program even if they are successful students. Note that some undergraduate programs have been launched or terminated during the examined period, hence we also omit these programs and their students, since we could not extract useful information from the low amount of data regarding these programs. After the filtering 23,136 rows remained in the dataset, however there was an upgrade of the administrative system in 2012 and some data were not restored yet, hence the high school grades are missing in more than 12 thousand rows, what is more, the matura exam results are absent in 15,485 rows. Hence, we trained and tested the models on different datasets, namely we created two new datasets: in the first set, the missing fields are imputed using Multiple Imputation by Chained Equations (MICE) with Bayesian Ridge Regression, and the other is obtained by the exclusion of incomplete rows. Note that XGBoost can handle missing values, and we have trained the XGBoost model on both the original dataset, and on a dataset with complete rows, and we found that the model performed worse on the imputed dataset. Due to the sensitive nature of our data, we generated an alternative synthetic dataset with a recently proposed conditional generative adversarial network that was designed for modeling tabular data [32].

Most of our analyses are performed on a dataset, that was obtained by removing the incomplete rows. Unfortunately, this dataset contains only 5,239 instances, but it is still larger than the dataset of most of the related works [12], [24], [26]. The features of the dataset are listed in Table I.

Table I
Summary of features

Feature Class	Feature name	Type
University program related	Student ID	Nominal
	Program ID	Nominal
	Faculty of program	Categorical
	Financial Status	Binary
	Re-enrolled	Binary
Target variable	Final status	Binary
High school performance related	University entrance score	Real
	Surplus score	Real
	Extra points	Real
	Competition achievement	Binary
	UES calculating method	Binary
Matura exam results	Hungarian Language and Literature	Real
	Mathematics	Real
	History	Real
	Foreign Language	Real
Average of grades	Mathematics	Real
	Hungarian Language	Real
	Hungarian Literature	Real
	History	Real
	Foreign Language	Real
	Chosen science subject	Real
Certificate of foreign language	Score (based on the number and level of certificates)	Real
Personal details	Gender	Binary
	Age	Real
	Location type of high school	Nominal
	Years elapsed between the time of matura and enrollment	Real

During the data preprocessing, we have also transformed and created new features, e.g. we have created a new ID variable because the original student ID does not identify the records uniquely, since a student can enroll in different undergraduate programs, and that means multiple rows with the same student ID. The question naturally arises: if a student drops out from a program but then enrolls in another one and then graduates, how should we define the academic performance of the student? Is (s)he a dropped out student or a graduated one? Our solution is the following: we assign different IDs to these two scenarios, hence according to our ID, in this case, there are two different students, one of them dropped out but the other one graduated.

We have also created an attribute, called *re-enrolled*, which indicates whether the student has already been enrolled in the university before, dropped out, and then enrolled again. Unfortunately, due to some certain policy changes, the number of re-enrolling students increases from year to year. Moreover, the generated *surplus score* attribute measures how much the given student outperformed the minimum university entrance score with their entrance score, and it is weighted with the MUES of the undergraduate program. The so-called *years elapsed* variable is defined as the years elapsed between the time of the matura examination and the enrollment. We also introduced a new variable that encodes the location type of the high school of the students, and finally, we defined a score for the foreign language exams that takes into account the number of different languages and the level of the exams.

Without attempting to be comprehensive, further data preprocessing steps involve: rescaling the university entrance score[2], multiplying the advanced level matura exam scores by 1.5[3], grouping together the elective subjects, pivoting and merging several tables, encoding the categorical features, and rescaling the numerical attributes to the 0-1 interval.

IV. Methods

A. Data generation

Due to the sensitive nature of our data, one of our objectives was to generate an anonymous synthetic dataset that is somewhat similar to the original. Our goal as to use a sophisticated method and Bálint Gyires-Tóth suggested tying GANs. Fortunately, there are a few recently proposed GANs that are designed for anonymizing and modeling tabular data, e.g. table-GAN [33], TGAN [34], and CTGAN [32]. In the

[2] In 2013 the maximum obtainable UES changed from 480 to 500

[3] In order to decrease the dimensionality of the data

final version of this project, we used the CTGAN model. As an alternative option, we could use a privacy-preserving machine learning framework, called CrypTen[4], but the reason why we opted for GANs is that our original dataset is relatively for deep neural networks, but with the help of GANs we can create a larger dataset.

B. Modeling

In this work we apply several machine learning models: for baseline models we use random forest and XGBoost, and we train two different neural networks. The difference between the two DNNs relies in a few hyper-parameter setting. Namely they differ in depth, activation function, kernel initializer, and optimizer. The hyper-parameters are shown in Table II, where the chosen parameters of the first network are highlighted in **bold**, and the parameters of the second network are $\underline{\text{underlined}}$.

1) The problem: After the data preparation steps described in Sec. III, we put together a dataset containing information about roughly 5000 students. The problem at hand is a binary classification problem:

- Inputs: attributes of students entering into our university
  - these are known at the time of enrollment
- Output: final academic performance
  - binary variable (1 if graduated, 0 if dropped out)
  - its distribution is around 50-50% in the dataset

Note that the reason why we have a balanced dataset, is not because only half of the students graduate, but because we have data of students who enrolled between 2010 and 2018, which means that a students who enrolled after 2016 could not graduate yet, but they may dropped out already. This can be considered as a natural oversampling technique.

2) Network structure: A recent paper [35] proposes an interpretable deep tabular data learning network, called TabNet, that can even outperform decision tree variants on a wide range of tabular data. Unfortunately, in this project we could not utilize the current implementations[5-6] of TabNet. Besides TabNet, articles and blog posts usually state that a simple fully connected deep neural network is sufficient for almost all tabular data analysis tasks, thus we started with these.

C. Evaluation and optimization

We conducted a hyper-parameter optimization with the *hyperas* package [36], that was also demonstrated in the labs of the aforementioned course. The tested parameters are described in Table II. Note that this test was conducted on slightly different subset of the features.

The hyperas package uses the Tree-structured Parzen Estimator (TPE) hyper-parameter optimization algorithm. The parameter settings of the algorithm are as follows: the number of iterations was set to 300, each with 30 epochs, the optimized metric was chosen to be the accuracy on validation dataset. Validation and training accuracy of the models against the tuned hyper-parameters can be seen in Figure 2 and in Figure 3, respectively. From these figures it is clear, that we are trying to solve a non-trivial problem since it not so easy to train a NN on this dataset, and a lot of training trials end up with the accuracy of a coin-toss. However we could achieve similar accuracy to that of related works

hyper-parameter	**values**
depth[7] of the NN:	$\{1, 2, 3, 4, \underline{5}, \mathbf{6}, \ldots, 10\}$
size of the FC layers:	$\{\underline{\mathbf{16}}, 24, 32, 40\}$
activation functions:	{$\underline{\text{ReLU}}$, Sigmoid, **SeLU**}
additional layer between FC layers:	$\underline{\textbf{Dropout}}$, BatchNormalization, AlphaDropout
kernel initializer:	$\underline{\text{(glorot)normal}}$[8], **(glorot)uniform**[9], uniform(-1,+1), constant ones}
optimizer:	{SGD with different learning rates (.01, .1, 1), **Adam**, $\underline{\text{RMSprop}}$}
batch size:	{$\underline{\textbf{16}}$, 32, 64, 128 }

Table II
THE TESTED HYPER-PARAMETER SPACE

TPE iteratively modifies the sampling distribution of the hyper-parameter space, preferring those set-ups that worked better before, thus it makes sense to look at the last couple of set-ups as well, see the most frequently used parameters in the last N steps in Table III.

During the hyper-parameter optimization we have also tested a few new techniques, e.g. the AlphaDropout layer and SeLu (scaled exponential linear units) activation function. These techniques are the building blocks of self-normalizing neural networks, which was proposed by Klambauer et al. [37]. The authors of [37] argue that the usage of these methods can enhance the performance of feed forward neural networks.

The chosen parameters after studying the logs of the optimization are highlighted with boldface in Table II, referred as Model 1 from now on. An alternative setup, returned by the optimization as the ”best model” was also tested, underlined in the same table, referred as Model 2 from now on. The maximal accuracy we were able to reach is around 70-75% on the test dataset. This is around the same for XGBoost and RandomForest. These models were able to reach much better accuracy on the training set but their ability to generalize is not better, see Table...

Note that for both selected neural networks 10-fold stratified cross-validation was applied. Figure 4 and Figure 5 show the progress of the binary cross-entropy loss and the accuracy through the epochs for each fold. The training of the folds lasted for 150 epochs with a batch size of 16.

Based on these we decided to go forward with the structure

[4] https://crypten.ai/

[5] https://github.com/dreamquark-ai/tabnet

[6] https://github.com/titu1994/tf-TabNet

[7] number of FC layers before the output layer

[8] zero centered normal dist. with $std = \sqrt{2/(\#nodes_{in} + \#nodes_{out})}$

[9] unif. dist. with limits $\pm\sqrt{6/(\#nodes_{in} + \#nodes_{out})}$

	300	250	200	150	100	50
depth	6	6	6	6	6	6
activ	selu	selu	selu	selu	selu	selu
dens_size	40	24	24	24	24	24
bn_or_drop	drop	drop	drop	drop	drop	drop
drop_rate_d	(0.3, 0.4]	(0.2, 0.3]	(0.3, 0.4]	(0.3, 0.4]	(0.4, 0.5]	(0.4, 0.5]
kernin	glorot_normal	glorot_uniform	glorot_uniform	glorot_uniform	glorot_uniform	glorot_uniform
optim	rmsprop	rmsprop	adam	adam	adam	adam
n_batch	16	16	16	16	16	16

Table III
MODES OF THE HYPER-PARAMETERS IN THE LAST $N \in \{300, \ldots, 50\}$ STEPS OF THE OPTIMIZAION

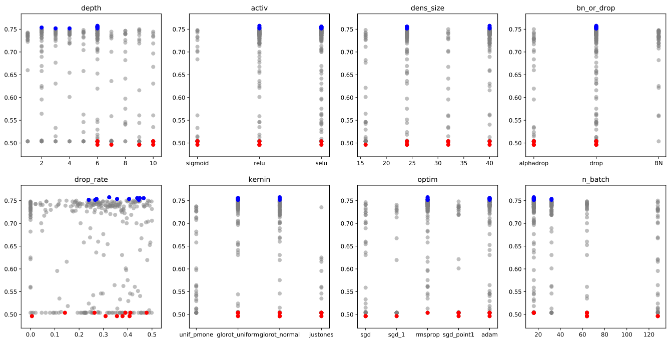

Figure 2. Accuracy on the *validation* set against different hyper-parameters. Blue points indicate the best ten, read point indicate the worst ten models.

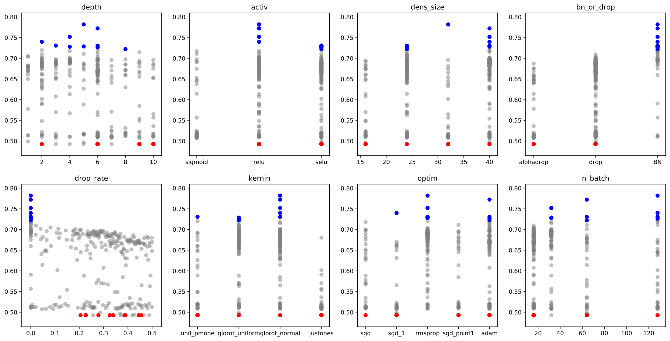

Figure 3. Accuracy on the *training* set against different hyper-parameters. Blue points indicate the best ten, read point indicate the worst ten models.

of Model 1. Its final training was conducted on the whole training set (from which the previous folds were generated), and it lasted for 300 epochs with a batch size of 16. The progress of the loss and the accuracy on the training and the test sample is depicted in Figure 6. The stabilization of the values is probably due to the applied learning rate reduction, which kicked in after the training loss plateaued for 30 consecutive epochs.

We examined some a few model interpretability tools to understand the trained model. In the boxplots of Figure 7, we can see how the predicted probability of the graduation changes along the UES in the training set. This is in line with the expected behavior of the model.

We also checked the permutation importance of the features.

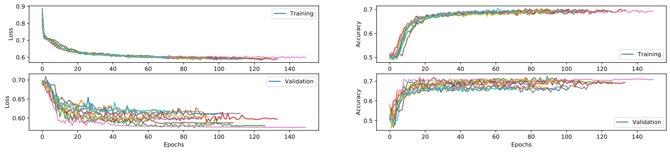

Figure 4. Loss and accuracy of Model 1 through the cross-validation process

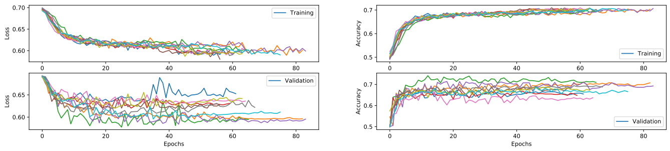

Figure 5. Loss and accuracy of Model 2 through the cross-validation process

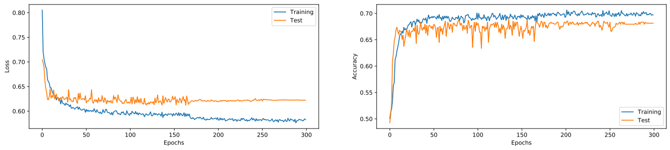

Figure 6. Loss and accuracy of the final model training

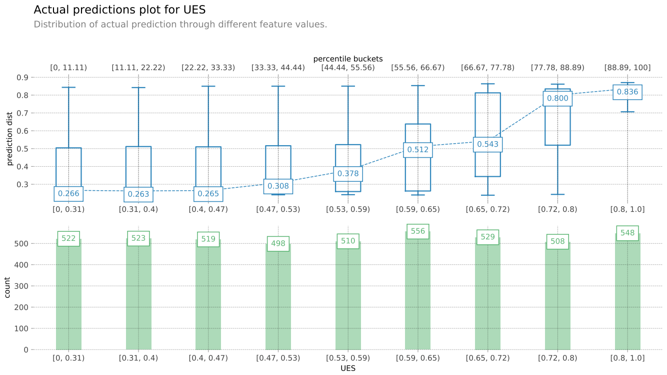

Figure 7. Actual predictions along the UES. The UES score is mini-max scaled to the $[0, 1]$ interval, and the size of the bins is balanced.

	name	importance
44	years_between	0.0499
0	program_code_prog_1	0.0250
18	program_code_prog_9	0.0199
45	surplus_score	0.0197
19	financing_source_self_funded	0.0154
34	math	0.0135
15	program_code_prog_6	0.0122
30	ues	0.0116
27	ues_calc_method_summation	0.0110
32	foreign_lang	0.0109

Table IV
THE 10 MOST IMPORTANT FEATURES BASED ON PERMUTAION IMPORTANCE IN THE TRAINING SET

	name	importance
44	years_between	0.0332
0	program_code_prog_1	0.0179
18	program_code_prog_9	0.0145
40	math_hs	0.0122
30	ues	0.0092
45	surplus_score	0.0092
27	ues_calc_method_summation	0.0088
19	financing_source_self_funded	0.0080
22	gender_Male	0.0076
23	place_of_hs_capital_city	0.0053

Table V
THE 10 MOST IMPORTANT FEATURES BASED ON PERMUTAION IMPORTANCE IN THE TEST SET

The idea is that feature importance can be measured by looking at how much the performance of a trained model decreases, when the values of a feature are permuted (shuffled), ceteris paribus. Namely, first we train a model and calculate its performance, then we shuffle the values of one feature, and without re-training the model, we again calculate its performance on the shiffled data and investigate how much the performance worsened. Clearly if a variable is important, then the accuracy of the model will drastically decrease after the permutation. On the other hand if we shuffle the values of a redundant variable, then the performance of the model will remain the same, or in some cases it can happen that it even increases due to the randomness. The 10 most important features according to permutation importance is listed in Table IV and V.

D. SHAP value

In this project, we would also like to understand why the machine learning models predict dropout or graduation for a given student. To this end, similarly to [29], we use the SHapley Additive exPlanations (SHAP) approach [31]. The SHAP value is a state-of-the art machine learning model explainability tool, that is based on the Shapley value, which was introduced by Lloyd Shapley in [38]. Shapley value is a cooperative game concept that quantifies the contribution of each player in a coalition.

Note that some machine learning models, especially deep neural networks, are extremely difficult or might even impossible to interpret, that is why some segments of *industry* are afraid of deploying neural networks based systems, for example economists prefer linear regression models with a reason. On the other hand explainable machine learning is a new research area of this field, and in this work we aim to show how can we explain the decisions of sophisticated models with these techniques.

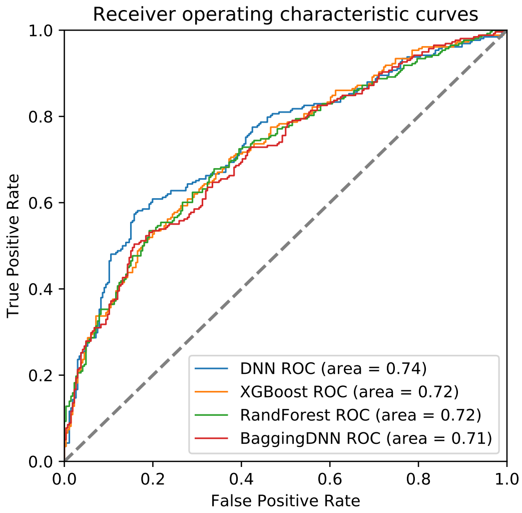

Figure 8. ROC curves of the trained final model, the bagged estimator version, a random forest, and an XGBoost model

In machine learning settings, the players are the explanatory variables and the v value function of a coalition corresponding to the given subset of variables is defined as follows. Following the notations of [39], let f denote the machine learning model that predicts the target variable (in our case the final academic performance) for an instance x (in our case an instance is a d-dimensional vector containing the attributes of a student). Let D be the index set of variables, i.e. $D = \{1, 2, \ldots, d\}$ and S be a subset of D. Then we define $f_S(x)$ as the conditional expectation of $f(x)$ given the values of the X_i attributes belonging to the set S, formally:

$$f_S(x) = \mathbb{E}(f(x) \,|\, X_i = x_i, \, \forall i \in S)$$

Note that if S is an empty set, then $f_S(x)$ is the expectation of $f(x)$, i.e. $f_{\{\}}(x) = \mathbb{E}(f(x))$. Using these notations, the value of a contribution of a subset of features is defined as:

$$v(S) = f_S(x) - f_{\{\}}(x),$$

which is the change in prediction caused by observing the values of the S subset of attributes for a given instance x. The contribution of the ith feature is defined as its Shapley value with respect to the $v_S(x)$ value function, i.e.:

$$\begin{aligned}\varphi_i(x) &= \frac{1}{|D|} \sum_{S \subseteq D \setminus \{i\}} \frac{v(S \cup \{i\}) - v(S)}{\binom{|D|-1}{|S|}} \\ &= \sum_{S \subseteq D \setminus \{i\}} \frac{|S|!(d - |S| - 1)!}{d!} \left(v(S \cup \{i\}) - v(S)\right)\end{aligned}$$

V. Conclusion

'I always thought something was fundamentally wrong with the universe'

Acknowledgement

We thank the instructors of the course called *Deep Learning in Practice with Python and LUA* for designing and delivering this awesome course, where we could learn state-of-the-art deep learning technologies. We would especially like to express our gratitude to Professor Bálint Gyires-Tóth, for his useful suggestions on our project plan, his ideas and expertise improved this project a lot. We would also like to thank Roland Molontay, for his assistance and comments that greatly improved the manuscript. We are also grateful to the colleagues at the Central Academic Office, namely Mihály Szabó, Bálint Csabay, and István Bognár, for providing the data for this research, and also for their assistance in data understanding.

References

[1] L. Aulck, N. Velagapudi, J. Blumenstock, and J. West, "Predicting student dropout in higher education," *arXiv preprint arXiv:1606.06364*, 2016.

[2] P. von Hippel and A. Quezada-Hofflinger, "The data revolution comes to higher education: Identifying students at risk of dropout in chile," *Social Science Research Network*, 2017.

[3] J. J. Vossensteyn, A. Kottmann, B. W. Jongbloed, F. Kaiser, L. Cremonini, B. Stensaker, E. Hovdhaugen, and S. Wollscheid, "Dropout and completion in higher education in Europe: Main report," 2015.

[4] U. Heublein, "Student drop-out from German higher education institutions," *European Journal of Education*, vol. 49, no. 4, pp. 497–513, 2014.

[5] N. Raisman, "The cost of college attrition at four-year colleges & universities. policy perspectives." *Educational policy institute*, 2013.

[6] A. Latif, A. Choudhary, and A. Hammayun, "Economic effects of student dropouts: A comparative study," *Journal of Global Economics*, 2015.

[7] OECD, *Education at a Glance 2013*, 2013.

[8] O. for Economic Co-operation and D. Staff, *Education at a glance: OECD indicators 2013*. OECD, 2013.

[9] G. S. Abu-Oda and A. M. El-Halees, "Data mining in higher education: university student dropout case study," *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 1, p. 15, 2015.

[10] D. S. Fike and R. Fike, "Predictors of first-year student retention in the community college," *Community college review*, vol. 36, no. 2, pp. 68–88, 2008.

[11] M. Yorke, *Leaving early: Undergraduate non-completion in higher education*. Routledge, 2004.

[12] J. Lin, P. Imbrie, and K. J. Reid, "Student retention modelling: An evaluation of different methods and their impact on prediction results," *Research in Engineering Education Sysmposium*, pp. 1–6, 2009.

[13] Z. Kovacic, "Early prediction of student success: Mining students' enrolment data." *Proceedings of Informing Science & IT Education Conference*, 2010.

[14] R. Asif, A. Merceron, S. A. Ali, and N. G. Haider, "Analyzing undergraduate students' performance using educational data mining," *Computers & Education*, vol. 113, pp. 177–194, 2017.

[15] A. Hernández-Blanco, B. Herrera-Flores, D. Tomás, and B. Navarro-Colorado, "A systematic review of deep learning approaches to educational data mining," *Complexity*, vol. 2019, 2019.

[16] A. Dutt, M. A. Ismail, and T. Herawan, "A systematic review on educational data mining," *IEEE Access*, vol. 5, pp. 15 991–16 005, 2017.

[17] M. Kumar, A. Singh, and D. Handa, "Literature survey on educational dropout prediction," *IJ Education and Management Engineering*, vol. 2, pp. 8–19, 2017.

[18] L. Aulck, N. Velagapudi, J. Blumenstock, and J. West, "Predicting student dropout in higher education," *arXiv preprint arXiv:1606.06364*, 2016.

[19] C. Beaulac and J. S. Rosenthal, "Predicting university students' academic success and major using random forests," *Research in Higher Education*, pp. 1–17, 2019.

[20] G. W. Dekker, M. Pechenizkiy, and J. M. Vleeshouwers, "Predicting students drop out: A case study." *International Working Group on Educational Data Mining*, 2009.

[21] K. Shaleena and S. Paul, "Data mining techniques for predicting student performance," in *Engineering and Technology (ICETECH), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1–3.

[22] W. Xing and D. Du, "Dropout prediction in moocs: Using deep learning for personalized intervention," *Journal of Educational Computing Research*, vol. 57, no. 3, pp. 547–570, 2019.

[23] B.-H. Kim, E. Vizitei, and V. Ganapathi, "Gritnet: Student performance prediction with deep learning," *arXiv preprint arXiv:1804.07405*, 2018.

[24] C. Mason, J. Twomey, D. Wright, and L. Whitman, "Predicting engineering student attrition risk using a probabilistic neural network and comparing results with a backpropagation neural network and logistic regression," *Research in Higher Education*, vol. 59, no. 3, pp. 382–400, 2018.

[25] J. Y. Kuo, C. W. Pan, and B. Lei, "Using stacked denoising autoencoder for the student dropout prediction," in *2017 IEEE International Symposium on Multimedia (ISM)*. IEEE, 2017, pp. 483–488.

[26] R. Alkhasawneh and R. Hobson, "Modeling student retention in science and engineering disciplines using neural networks," in *2011 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2011, pp. 660–663.

[27] M. Plagge, "Using artificial neural networks to predict first-year traditional students second year retention rates," in *Proceedings of the 51st ACM Southeast Conference*. ACM, 2013, p. 17.

[28] B. Kiss, M. Nagy, R. Molontay, and C. Bálint, "Predicting dropout using high school and first-semester academic achievement measures," in *2019 17th International Conference on Emerging eLearning Technologies and Applications (ICETA)*. IEEE, 2019.

[29] M. Nagy, R. Molontay, and M. Szabó, "A web application for predicting academic performance and identifying the contributing factors," in *47th Annual Conference of SEFI*, 2019.

[30] M. Nagy and R. Molontay, "Predicting dropout in higher education based on secondary school performance," in *2018 IEEE 22nd International Conference on Intelligent Engineering Systems (INES)*. IEEE, 2018, pp. 389–394.

[31] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," 2017.

[32] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan," in *Advances in Neural Information Processing Systems*, 2019.

[33] N. Park, M. Mohammadi, K. Gorde, S. Jajodia, H. Park, and Y. Kim, "Data synthesis based on generative adversarial networks," *Proceedings of the VLDB Endowment*, vol. 11, no. 10, pp. 1071–1083, 2018.

[34] L. Xu and K. Veeramachaneni, "Synthesizing tabular data using generative adversarial networks," *arXiv preprint arXiv:1811.11264*, 2018.

[35] S. O. Arik and T. Pfister, "Tabnet: Attentive interpretable tabular learning," *arXiv preprint arXiv:1908.07442*, 2019.

[36] P. Max, "Hyperas," https://github.com/maxpumperla/hyperas, 2019.

[37] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," 2017.

[38] L. S. Shapley, "A value for n-person games," *Contributions to the Theory of Games*, vol. 2, no. 28, pp. 307–317, 1953.

[39] E. Štrumbelj and I. Kononenko, "Explaining prediction models and individual predictions with feature contributions," *Knowledge and information systems*, vol. 41, no. 3, pp. 647–665, 2014.