A framework for predicting, interpreting, and improving Learning Outcomes

Chintan Donda **Sayan Dasgupta** **Soma S Dhavala** **Keyur Faldu** **Aditi Avasthi**

{chintan, sayan, soma.dhavala, k, aditi}@embibe.com
Indiavidual Learning Pvt Ltd
Diamond District, HAL Old Airport Rd, Domlur
Bengaluru, Karnataka 560008

October 7, 2020

Abstract

It has long been recognized that academic success is a result of both cognitive and non-cognitive dimensions acting together. Consequently, any intelligent learning platform designed to improve learning outcomes (LOs) must provide actionable inputs to the learner in these dimensions. However, operationalizing such inputs in a production setting that is scalable is not trivial. We develop a Embibe Score Quotient model (ESQ) to predict test scores based on observed academic, behavioral and test-taking features of a student. ESQ can be used to predict a student's future scoring potential as well as offer personalized learning nudges, both critical to improving LOs. Multiple machine learning models are evaluated for the prediction task. In order to provide meaningful feedback to the learner, individualized Shapley feature attributions for each feature are computed. Prediction intervals are obtained by applying non-parametric quantile regression, in an attempt to quantify the uncertainty in the predictions. We apply the above modelling strategy on a dataset consisting of more than a hundred million learner interactions on an online learning platform. We observe that the Median Absolute Error between the observed and predicted scores is 4.58% across several user segments, and the correlation between predicted and observed responses is 0.93. Game-like "what-if" scenarios are played out to see the changes in LOs, on counterfactual examples. We briefly discuss how a rational agent can then apply an optimal policy to affect the learning outcomes by treating the above model like an Oracle.

1 Introduction

Outcome-based Learning is gaining prominence in learner-centric educational services[28]. Rather than focusing on the the approach to learning, outcome-based learning sets the goals on what a learner can accomplish. It is no coincidence that learning outcomes (LOs) are typically defined using action oriented verbs, largely concerned with academic achievements[1]. This pivoting of measuring LOs allows for self-discovery and has the potential to move away from one-size-fits-all learning solutions. However, such a paradigm shift can put undue stress on already over stretched human resources in the education sector. But, thanks to deeper mobile penetration and faster technology adoption rates, online learning platforms can, not only supplement, but also amplify the reach and potential of every stakeholder in the system, especially teachers and students[29]. In fact, the opportunity is so huge that we can even revisit the scope of LOs. Studies have shown that besides skill and competency, psychological traits such as a grit and resilience also contribute to success in general, and in academics in particular [6, 4]. This leads to the following important question - how do we measure LOs. There are both direct and indirect ways to measure them[23]. Test scores such as GMAT, GRE, JEE, are the most widely used form of direct measurements. Psychometric tests are available to indirectly measure non-cognitive, psychological constructs[31, 22]. In this work, we consider Test Scores as a form of an LO measurement except that the LO is now composite in nature. Our thesis is that, an assortment of skills are required to *perform* well on a test and *succeed*. Online learning platforms have the benefit of capturing fine-grained learner interactions. Depending on the depth of instrumentation, they can track various pedagogic and behavioural aspects of the learning process. We set forth to a framework to leverage such fine-grained signals to positively drive LOs, and do this in a way that the student can be an active participant in the decision process.

2 Related Work

In [9], predictive models were used to classify learners based on their engagement, which instructors can act upon *in situ*. Behavioural patterns associated with performance were mined using Matrix Factorization techniques in [17]. Predicting student test scores based on activity was considered in [18] using a linear regression model with features such as *tasks completed* and *number of attempts*, among others. Using gradient-boosted trees, [13] found that factors such as *scheduling* and *content* are important predictors of students success in MOOCs. A critical review of the state of the art papers in predicting student scores (in MOOCs) was undertaken in [11]. One striking observation is that, in the context of MOOCs, there can be multiple definitions of success (or outcomes) measured in terms of *dropout rate*,*certification*, and *final exam grade*. The importance of feature engineering and the need for providing actionable predictions was emphasized. In the next Section, we discuss the details of our approach.

3 Modeling Framework

We consider an extensible, end-to-end modeling framework, where multiple models work in concert, feeding forward their results to downstream models and applications. We lay emphasis on feature engineering pipelines, incorporating uncertainty quantification via prediction intervals, and an optimization framework to produce actionable feedback based on the predictive model.

3.1 Network of Models architecture

A functional view of the *Network of Models* architecture is shown in Fig. 1. It is inspired by the modality of construction of deep learning networks, where intermediate networks can be seen as feature engineering models. Unlike in traditional deep learning networks, the modules in this architecture can be black box or hand-crafted. For example, we used a Bayesian Knowledge Tracing (`BKT`)[5] model to summarize temporal learner interactions into interpretable features, which are consumed by a Concept Mastery Model. We used Deep Factorization model `deepFM` [12] to predict academic competency of student across 1,242 concepts. Note that `Embibe` platform has more than 11,000 concepts in its Knowledge Graph, and for this paper we merged granular concepts into broader concepts (eg: *Differential equation dilution problems* and *Differential Equation Growth and Decay Problems* are mapped to *Application of Differential Equations*) to arrive at the 1,242 concept set. `deepFM` has the ability to learn complex learner-concept interactions, and has a highly configurable model architecture. Along with the aforementioned academic features, a set of behavioural and test-taking features drive the Embibe Score Quotient model (`ESQ`) Model. `ESQ` attempts to model the scoring potential of a learner in a test yet-to-be-taken based on all observations available prior to taking the test[10]. The nudges model, using the learnt `ESQ` as an Oracle, predicts the changes in the learner state (described in terms of the observed features) to achieve the desired change in the LO. Subsequently we focus on ESQ, and briefly discuss the nudges model.

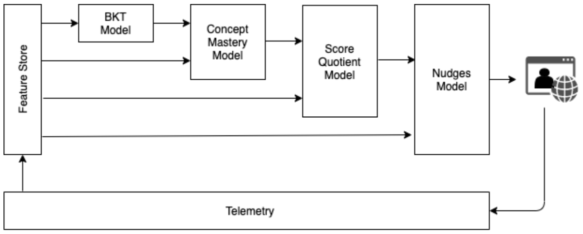


Figure 1: Network of Models

3.2 Data Preparation

`Embibe` is an online learning platform offering learning solutions for learners through K12 and beyond. It has a collection of immersive content as well as a repository of curated and AI generated assessments and tests. Tens of thousands of practice questions and hundreds of mock tests were made available for learners to prepare and practice [8]. For the purpose of developing `ESQ`, we consider data from 175,441 users on the `Embibe` Platform between January

2017 and December 2019. The raw interactions amount to more than 110 million data points. Data generated from bots and system testers are removed. We also consider only tests where at least 10% of total questions available should have been attempted, and 10% of total time available should have been spent.

3.3 Feature Engineering

An extensive list of more than 200 features was discussed in [30] useful in MOOC and online platform settings. Pyschological traits such as flow and resilience can also be measured based on survey type questions [31, 4]. On similar lines, we have considered academic, behavioral, effort and test-taking attributes (denoted as AQ, BQ, EQ, TQ respectively), as features to estimate the test score[10]. This notion is depicted in Fig .2.

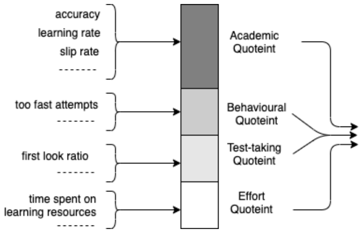


Figure 2: Hypothesized feature categories driving test score

AQ features: We use individualized BKT[32], to summarize the time series attempt level data. Whether a student is in a learned state in a concept/skill, whether the student has a tendency to guess a question against a concept are some of the features given by the BKT model. However, BKT is skill specific and it can not exploit student or concept similarities. In order to leverage this information, we fit deepFM[12] model to produce student competency across the 1,242 concept set. We reduce the dimension to 50 by random projections[2]. A subset of other features considered are *mean accuracy in last three tests*, *score on last test*.

TQ features: Fine-grained attempt-level events are meticulously captured while taking a test on Embibe platform. These events include looking at a question, choosing an answer option, changing an answer option, marking a question for review, attempting a question, swapping subjects, and many others. Using the timestamp for each event, we infer how a student has attempted the test paper at the event level. Each question is also tagged with its ideal time, which is defined as the expected time taken by an idealized student. Based on this we can infer what kinds of questions are attempted too quickly or too slowly. Each question is also tagged as whether *first-look* or not. *First-look* implies that an attempt decision on that question was taken by the student when the student viewed it for the first time, and has not altered that decision in any subsequent visit to that question. Features in TQ bucket summarize question answering behaviour while taking a test. All features are normalized to lie between 0 and 1.

BQ features: In addition to what a learner "knows", scoring on a test also depends on "how" the learner attempts any given test. Traits such as carelessness and over-confidence can affect the final test score. BQ features stand as a proxy for these traits. Features such as *ratio of careless mistakes to all attempts*, *ratio of time spent on non attempts to total time spent*, fall under this bucket.

EQ features: These features measure how much effort was spent by the student between two tests. On the Embibe platform, students can search, browse, watch content, ask questions and perform many other actions. We summarize how students engage based on time spent, and how the student is navigating from one environment to another. Quantification of effort based on what transpired between two events is a useful feature to include[15].

3.4 Model Building

Let $y_t \in \mathbb{R}$ and $x_t \in \mathbb{R}^p$ be the response variable (score), and the $p-$dimensional feature vector at time t, respectively. Then, we are interested in finding the mapping:

$$f : x_{0:t}, y_{0:t} \rightarrow y_{t+1}$$

Data flow is depicted in Fig. 3. We consider Random Forests (RF)as a baseline model, with 500 trees with depth 5. We only consider (x_t, y_t) to predict y_{t+1} as RF does not naturally accommodate time-series data. We also consider a Recurrent Neural Net (RNN) with stacked LSTM layers. We set the size of the LSTM's hidden state at 20. No hyper parameter tuning was done.

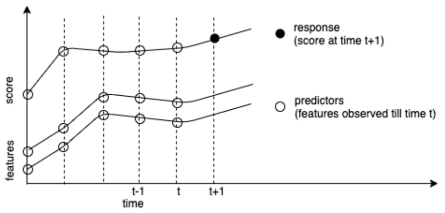


Figure 3: Data flow for Model Setup

3.5 Interpretability

A model is an approximation of reality. An interpretation can be considered as a way to assert that reality either in quantitative terms or in natural language. It is urged to use interpretable models, where possible[25]. There is a surge in providing interpretations to even seemingly opaque deep learning models. A recently introduced *Shapley values*[16] based model explanation technique unifies many existing feature attribution methods such as a `LIME`, `DeepLift`, and `TreeInterpreter`. Shapley possesses some nice theoretical properties such as 1) local accuracy 2) missingness and 3) consistency. In addition, thanks to the additivity property, feature attributions of any individual instance adds up to the predicted value. In our context, we can break up the test score into AQ, BQ, TQ, EQ components, and provide it as part of engaging feedback. We use `TreeExplainer` optimized for tree models such as a RFs, and `DeepExplainer` for RNNs, available in [16].

3.6 Uncertainty Quantification

Generally speaking, Machine Learning (ML) algorithms are concerned with predictions. Quantifying uncertainty, either in terms of confidence intervals (CIs) and/or prediction intervals (PIs) is usually an afterthought. The problem is exasperated in the Deep Learning space[20]. Such information is particularly useful when deploying ML models in production settings. For example, the product owner may decide not to use the prediction when it is very vague. Generating CIs for models that use third-party tools are very hard. On the other hand, generating PIs is somewhat straightforward, even though not widely known. It is only recently that Quantile Regression (QR)[14] has gained popularity among ML practitioners[27, 24]. We use `sk-garden` to fit Quantile Random Forest, and implement Check Loss, defined below, to fit quantiles with RNNs. The Check Loss is given as:

$$\rho_\tau(e) = (\tau - I(e < 0))e$$

The non-parametric PIs can be obtained by reporting $(\tau, 1-\tau)^{th}$ quantiles at every observed set of predictors. Not only are PIs easier to implement with third party tools, they are also relatively easier to explain than CIs.

3.7 From predictions to actions

A nudge can be defined as an action taken by an intelligent agent to positively impact LOs. Damgaard et al. [7] studied the effectiveness of nudges in the education domain, taking examples of nudges tried in different countries to help students study better. Some of their suggested nudges include displaying peer performance, use of relative grading rather than absolute, re-framing incentives as loss, and showing learning videos on the importance of grit and hard work. A schematic of the `nudges` model is shown in Fig. 4. Providing feedback, a kind of nudge, is also found to be useful [21]. Bramucci et al [3] suggested the use of a recommendation engine to help students pick courses that would maximize their success. We can see nudges in the much broader context of Reinforcement Learning (RL). Shayan et al. [26] reviewed RL as a generic paradigm for planting nudges into the overall learning framework. In this work, we

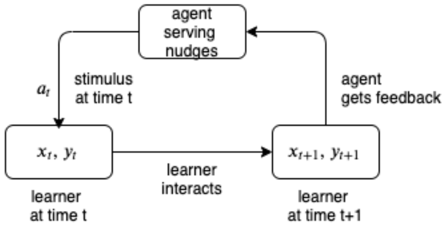


Figure 4: A nudge model

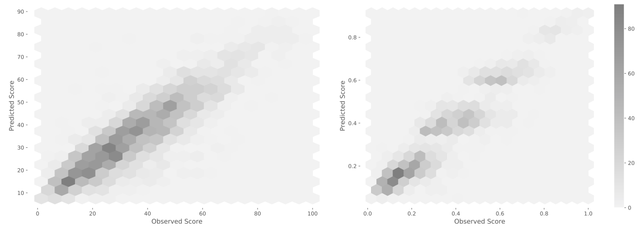
develop ESQ that can act as an Oracle, which for a given student's state, returns the reward function. This allows us to simulate how students learn, and design optimal policies to provide nudges. That is, we study the inverse problem: *What state should a learner move to, from a given state in order to obtain a particular reward?*. It is akin to back propagating the output gradient (change in reward) to find the input gradient (change in input state). If indeed the causal factors are also encoded in the state, then by solving this inverse problem, we can also suggest which nudge must be given to the student, for the desirable change in the reward (improvement in the test score). We pose this as an optimization problem. Recall that, SBT learns the function $f(.)$, parameterized by θ: $y \leftarrow f(x;\theta)$. We treat θ as the unknown while learning the function(training phase), and while solving for nudges, we treat δx as the unknown in the following minimization problem:

$$\min_{\delta x} \ell(y + \delta y, f(x + \delta x; \theta))$$

Here, δy is the desired improvement in the score from y, δx is the anticipated change needed in the student's state from x, $\ell(.)$ is a suitable loss function that handles the domain constraints. It is out of scope of this paper to discuss the details, but we demonstrate its utility in the next section.

4 Results and Discussion

We fit a RandomForest model on the entire dataset. However, we observed that variance was changing with observed score in the residuals, particularly in the low scoring buckets (students score less than 25%), and the Median Absolute Error (MedAE) was above 10% which was not acceptable for real world deployment. To address this issue, We segment the users into four buckets based on test scores seen on their previous three test attempts, and fit one RandomForest for each score bucket. We did not use any bucketing strategy to fit RNNs. A two dimensional histogram of observed vs. predicted responses is shown in Fig. 5. The residuals of the RNN models have some segmentation, which we believe is due to user encoding we did to reflect the score buckets. Between the two models, the RNN had the smaller Median Absolute Error (MedAE) at 4.53%, and the Pearson correlation ρ between the predicted and observed density was 0.9331 (see Table 1). With bucketwise RFs, the MedAE is 6.15% and ρ is 0.8235. To see the effect of adding Concept Mastery vectors, we ran an experiment on one third of the dataset. We obtain 90% PIs by fitting a Quantile RF model at $\tau = 0.05$ and 0.95. We train the same RNN model with Check Loss to fit quantiles. For a particular user, PIs over a period of time are shown in Fig. 6.



(a) RF

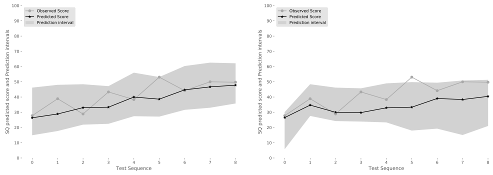
(b) RNN

Figure 5: Observed vs Predicted Density

Table 1: Performance summary of RF and RNN models. n: number of samples; p: number of features; r: train/test split ratio. ()* represents Model with Concept Mastery vectors

Model	p	n	r	Train				Test			
				RMSE	MAE	MedAE	ρ	RMSE	MAE	MedAE	ρ
RF	67489	54	0.8	11.21	8.13	6.16	82.35	12.50	9.12	6.73	77.15
RF*	20358	104	0.95	11.50	8.13	5.83	80.40	12.08	8.84	6.67	77.07
RNN	67489	54	0.8	7.1	5.51	4.53	91.31	7.43	5.69	4.57	92.05
RNN*	20358	104	0.95	8.84	6.72	5.37	88.60	10.42	7.54	5.91	88.92

We can use the width of PI as a measure to assert confidence in the predictions – predictions with wide PIs can be not shown. The Shapley model summary is shown in Fig. 7. It turns out that, not surprisingly, *mean accuracy* (coded as aq_16) is the most discriminating feature. It is interesting to note that, what kind of test a student is taking also influences the test score(bq_17). A student with a particular ability can perform differently based on the difficulty of



(a) RF (b) RNN

Figure 6: 90% Prediction Intervals for a user

the test - a viewpoint fundamental in Item Response Theory modeling. Overall, AQ features' contribution is 60.98% to the mean predictions. Feature attributions at an individual level can also be computed based on Shapley values. For a particular user, we can show the contribution of each of the features to the score. A force-plot of individualized `Shapley values` for a user is shown in Fig. 8. For example, we can see that, `aq_16` contributed 1.57 points to the predicted score, whereas `bq_20`(*number of test sessions*) reduced predicted score by 0.73 points, possibly because, the student should have practiced this test on par with others. These individualized feature attributions can be used to break up the score into explainable and actionable components. The nudges model takes this idea further and systematically perturbs the inputs which can potentially induce a desired a change in the score (score is a proxy for LO in this case). Using the `nudges` model introduced earlier, we vary one feature at time - much like how coordinate descent works. In fact, the analogy is precisely that. We are optimizing the learning process to improve the LOs. In Fig. 9, we show affected features which improve the current score by more than 10 points.

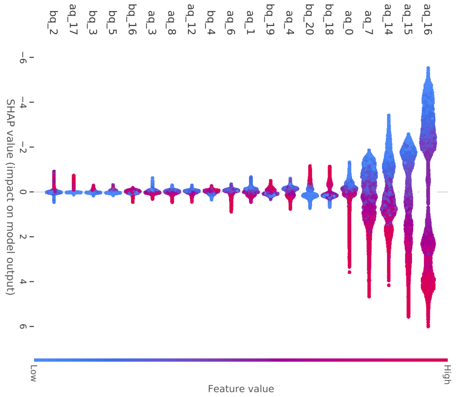


Figure 7: Shapley Model Summary

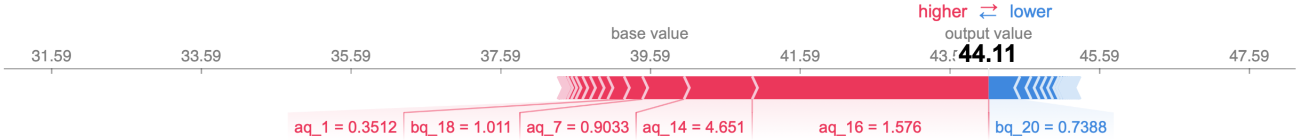


Figure 8: Individualized Shapley values for a user

Our approach draws parallels to recent work on providing explanations to black-box deep learning models via counterfactual examples[19]. While the `nudges` model can figure out what the state should be, it does not influence the state directly. On the `Embibe` platform this happens via a messaging capability - a user can be shown actionable feedback like *You seem to be making careless mistakes. Revise your calculations before submissions*. Once a causal link is established between an action and the affecting feature, `nudges` model can either influence the learner actions directly or it can be used in a gaming mode, where the learner can play with different simulated actions and pick a suitable action. To see the effect of the feedback, we mined historical data and sampled 1,116 users who have taken at least 10 valid tests (defined earlier) on our platform. In Fig. 10, we show how undesirable behavior parameters, such as *wasted attempts, unused time, overtime incorrects*, trend down as marks scored increases test-on-test. The direction of the results is encouraging but we exercise caution in interpreting them. We have not controlled for user attributes,

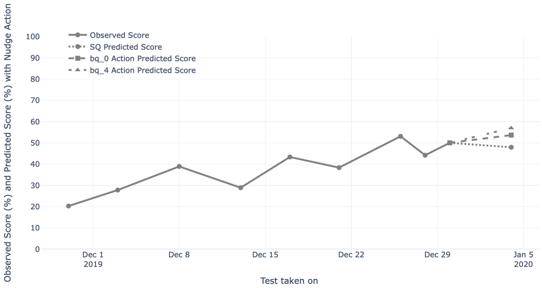


Figure 9: Nudges in Action

and can not attribute the behavioral change only to the feedback mechanism. However, the analysis suggests that we can persist with the hypothesis that providing useful feedback helps in impacting the intended behavioural aspects. A controlled A/B experiment will test that hypothesis.

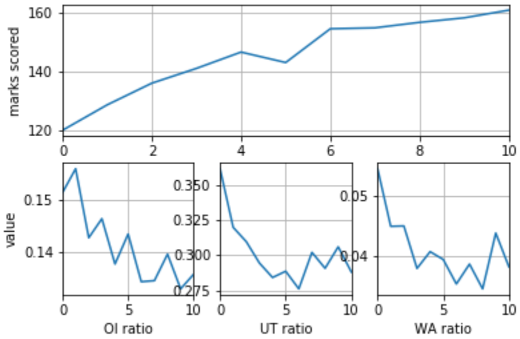


Figure 10: Test-on-test progress of marks scored and change in behavior parameters. Reading clockwise from top are plots for *marks scored, wasted attempt ratio, unused time ratio and overtime incorrects ratio*

5 Conclusions and Future Work

In this work, we proposed a framework for predicting Learning Outcomes, defined in terms of test scores. Various features were considered that summarize learner interactions on the platform. We used individualized BKT parameters to summarize temporal attempt level data. Deep Factorization model, with dimensionality reduction, was used to model a user's mastery in 1242 concepts. We are testing our approach on larger datasets. We considered RandomForest model as a baseline to predict test scores. We also considered RNNs as they can naturally handle the sequential nature of the attempt data. While the RNN model gave best results, the residuals did not look smooth. Using `TreeExplainer`, we obtained individualized Shapley values. Based on this data, we observed that, AQ features contribute about 60.98% of the scores. In order to fit Shapley values to an RNN, based on `DeepExplainer`, we had to drop the time-distributed layer, which reduced the RNN performance. We are working on architecture search and feature encodings to improve the performance and model diagnostics, while retaining interpretability. Individualized feature attributions can be presented to provide a breakup of the test score in terms of the features. If the features themselves are actionable, it is possible to directly avail this information to drive actions. We modelled this idea formally by posing nudge recommendation as an optimization problem. Since we are treating the ESQ as an Oracle, many simulations can be performed to identify the optimal policy. Our formalism can be easily incorporated into a Reinforcement Learning paradigm. We are working on the framework to provide optimal nudges and validate this hypothesis.

Acknowledgements

The authors would like to thank Achint Thomas for very helpful discussions and Anwar Sheikh for pedagoical suggestions.

References

[1] M. Battersby. So, what's a learning outcome anyway? *Vancouver: Centre for Curriculum, Transfer and Technology, British Columbia Ministry of Advanced Education*, 1999.

[2] E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the Seventh ACM Sigkdd International Conference on Knowledge Discovery and Data Mining*, page 245–250, New York, NY, USA, 2002. Association for Computing Machinery.

[3] R. Bramucci and J. Gaston. Sherpa: increasing student success with a recommendation engine. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, page 82–83, 2012.

[4] M. Christopoulou, A. Lakioti, C. Pezirkianidis, E. Karakasidou, and A. Stalikas. The role of grit in education: A systematic review. *Psychology*, 9:2951–2971, 2018.

[5] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4:253–278, 1994.

[6] M. Csíkszentmihályi. *Creativity: Flow and the Psychology of Discovery and Invention*. Harper Perennial, New York, 1996.

[7] M. T. Damgaard and H. S. Nielsen. The use of nudges and other behavioural approaches in education. *EENEE Analytical Report*, 2017.

[8] S. Dhavala, C. Bhatia, J. Bose, K. Faldu, and A. Avasthi. Auto generation of diagnostic assessments and their quality evaluation. *Proceedings of The 13th International Conference on Educational Data Mining*, pages 730–735, 2020.

[9] E. Erkan, G.-S. Eduardo, B.-L. Miguel, D. Yannis, and A.-P. Juan. Generating actionable predictions regarding mooc learners' engagement in peer reviews. *Behaviour & Information Technology*, pages 1–18, 09 2019.

[10] K. Faldu, A. Avasthi, and A. Thomas. Method and system for measurement of road profile, 3 2018. Pub. No. : US 2018/0090023 A1.

[11] J. Gardner and C. Brooks. Student success prediction in moocs. *User Modeling and User-Adapted Interaction*, 28:127–203, 2018.

[12] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He. Deepfm: A factorization-machine based neural network for ctr prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 1725–1731. AAAI Press, 2017.

[13] F. H. Khe, H. C. Xiang, and Y. T. Qiao. What predicts student satisfaction with moocs: A gradient boosting trees supervised machine learning and sentiment analysis approach. *Computers & Education*, 145, 2020.

[14] R. Koenker. *Quantile Regression*. Econometric Society Monographs. Cambridge University Press, 2005.

[15] A. Lalwani and S. Agrawal. What does time tell? tracing the forgetting curve using deep knowledge tracing. *Artificial Intelligence in Education*, pages 158–162, 06 2019.

[16] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee. Explainable ai for trees: From local explanations to global understanding. *arXiv preprint arXiv:1905.04610*, 2019.

[17] M. Mirzaei and S. Sahebi. Modeling students' behavior using sequential patterns to predict their performance. In S. Isotani, E. Millán, A. Ogan, P. Hastings, B. McLaren, and R. Luckin, editors, *Artificial Intelligence in Education*, pages 350–353. Springer International Publishing, 2019.

[18] M. Mogessie, G. Riccardi, and M. Ronchetti. Predicting students' final exam scores from their course activities. *IEEE Frontiers in Education Conference*, pages 1–9, 10 2015.

[19] R. K. Mothilal, A. Sharma, and C. Tan. Explaining machine learning classifiers through diverse counterfactual explanations. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617, 01 2020.

[20] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, 2015.

[21] D. J. Nicol and D. Macfarlane-Dick. Formative assessment and self-regulated learning: a model and seven principles of good feedback practice. *Studies in Higher Education*, 31(2):199–218, 2006.

[22] D. Patry and R. Ford. Measuring resilience as an education outcome, 06 2016.

[23] B. A. Price and C. H. Randall. Assessing learning outcomes in quantitative courses: Using embedded questions for direct assessment. *Journal of Education for Business*, 83(5):288–294, 2008.

[24] F. Rodrigues and F. Pereira. Beyond expectation: Deep joint mean and quantile regression for spatiotemporal problems. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–13, 02 2020.

[25] C. Rudin. Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. *arXiv e-prints*, 11 2018.

[26] D. Shayan, A. Vincent, and B. Emma. Where's the reward? a review of reinforcement learning for instructional sequencing. *International Journal of Artificial Intelligence in Education*, 29:568—-620, 2019.

[27] I. Takeuchi, Q. V. Le, T. D. Sears, and A. J. Smola. Nonparametric quantile estimation. *Journal of machine learning research*, 7(Jul):1231–1264, 2006.

[28] K. Tshai, J.-H. Ho, E. Yap, and H. Ng. Outcome-based education – the assessment of programme educational objectives for an engineering undergraduate degree. *Engineering Education*, 9(1):74–85, 2014.

[29] D. Vasant, N. Nandan, M. Shankar, and P. Nagaraju. Big data as an enabler of primary education. *Journal of Big Data*, 4(3):137–140, 2016.

[30] K. Veeramachaneni, U. O'Reilly, and C. Taylor. Towards feature engineering at scale for data from massive open online courses. *CoRR*, abs/1407.5238, 2014.

[31] B. Wolfigiel and A. Czerw. A new method to measure flow in professional tasks - a flow-w questionnaire (flow at work). *Polish Psychological Bulletin*, 48(2):220–228, 2017.

[32] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In H. C. Lane, K. Yacef, J. Mostow, and P. Pavlik, editors, *Artificial Intelligence in Education*, pages 171–180, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.