MOOC Dropout Prediction: Lessons Learned from Making Pipelines Interpretable

Saurabh Nagrecha
iCeNSA,
Department of Computer Science and Engineering,
University of Notre Dame,
Notre Dame, Indiana 46556
snagrech@nd.edu

John Z. Dillon
iCeNSA,
Department of Computer Science and Engineering,
University of Notre Dame,
Notre Dame, Indiana 46556
jdillon5@nd.edu

Nitesh V. Chawla
iCeNSA,
Department of Computer Science and Engineering,
University of Notre Dame,
Notre Dame, Indiana 46556
nchawla@nd.edu

ABSTRACT

Dropout prediction in MOOCs is a well-researched problem where we classify which students are likely to persist or drop out of a course. Most research into creating models which can predict outcomes is based on student engagement data. *Why* these students might be dropping out has only been studied through retroactive exit surveys. This helps identify an important extension area to dropout prediction—how can we interpret dropout predictions at the student and model level? We demonstrate how existing MOOC dropout prediction pipelines can be made interpretable, all while having predictive performance close to existing techniques. We explore each stage of the pipeline as design components in the context of interpretability. Our end result is a layer which longitudinally interprets both predictions and entire classification models of MOOC dropout to provide researchers with in-depth insights of *why* a student is likely to dropout.

CCS Concepts

•**Information systems** → *Personalization;* •**Computing methodologies** → *Supervised learning by classification;* •**Applied computing** → *Learning management systems;*

Keywords

MOOC; Dropout; Machine Learning; Visualization

1. INTRODUCTION

Massive Online Open Courses (MOOCs) democratize access to education for students all around the world. However, the most prominent downside to MOOCs is their high attrition rates [23, 15, 20]. This has catalyzed considerable research towards identifying students likely to drop out of these courses [31, 16, 13]. Using past student data, these approaches train classifiers to predict whether a student will persist in the following week or drop out. Though significant work has been devoted to identifying *who* is likely to drop out, the *why* has largely been investigated through retroactive surveys [20, 11]. Through this paper, we expand on the work done on dropout prediction by taking a deeper look into the models learned and predictions made.

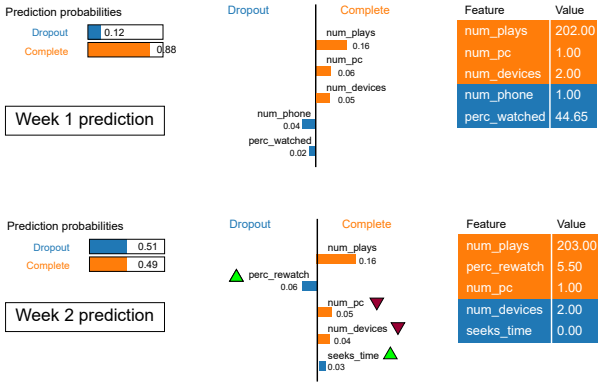


Figure 1: **An Interpreted Prediction for a sample student**. Dropout prediction's role ends at providing a set of labels at each week. Our model provides probability estimates similar to [13] and then links the predicted outcome to signals from the feature values.

Dropout predictions are meant to be used by MOOC creators and researchers for intervention efforts and to gain greater insight into refining future versions of the MOOC. Instead of just obtaining "anecdotal" predictions about each student we argue that it is more insightful to interpret the classifier at the model and the instance level. In Figure 1, the classifier trained on Week 1 data predicted that this student was likely to complete the course, whereas given the next week's data, the classifier predicted that the student was likely to drop out. Using our model, we can go beyond these binary predictions and suggest 1) probability estimates (similar to the approach followed in [13]) and 2) possible signals from feature values which contributed towards the classifier's predicted output. Our solution here uses Local Interpretable Model-Agnostic Explanations (or LIME for short) to reconcile the *who* with a possible signal for *why* from the data [21].

WWW'17 Companion, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4914-7/17/04.
http://dx.doi.org/10.1145/3041021.3054162

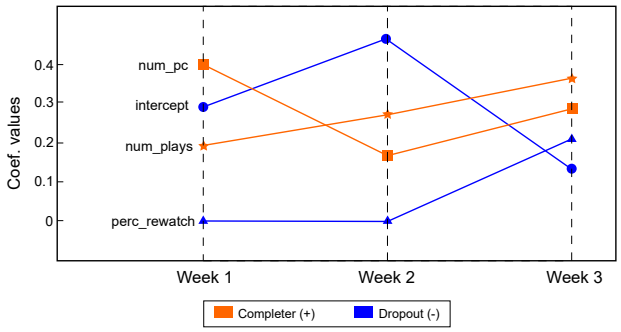


Figure 2: **Interpreting Linear Models**. Dropout prediction models trained at each week prioritize features differently. In this figure, we visualize the relative importance of various features by plotting linear model coefficients (and intercept) relative to each other. Longitudinal trends can be inferred tracing these coefficient values across time for retrained classifiers. We color code each feature according to the sign of its coefficient, which in turn is an indication of which class it nudges the classifier towards.

Echoing this argument for the models used for dropout prediction, we would like to know how the classifier discriminates signals for dropouts versus persisters at each week in the MOOC. Model inference on interpretable classifiers is the most intuitive way to achieve this [9]. An example of this is shown in Figure 2, where we interpret the coefficients and intercept of logistic regression classifier over a set of weeks. Here, we visualize the coefficients and intercept of the trained linear model on a real axis. Coefficients contributing to each class are color coded for ease of visualization. For classifiers other than the small subset we identify as interpretable, we suggest alternative techniques from literature to simulate this level of interpretability in Subsection 4.2.

While a MOOC is in session, these predictive models are re-trained each week and as a result, a given student might have multiple such predictions made during the course of their interaction with the MOOC. This imparts a longitudinal component to these models and predictions, shown in Figures 1 and 2. A predictive model evolves over time and the target student now has a traceable history of predicted outcomes. We see that the coefficients for a given feature can be traced across various weekly models. This re-prioritization is useful in characterizing the changing profile of dropouts at each week.

Our proposed approach to interpretability of dropout prediction can thus be broken down into the following components: 1) Model inference, 2) individual prediction inference, and 3) tracing models and predictions longitudinally.

We demonstrate how each of these components can be implemented in a real MOOC by providing a fully outlined case-study. Our data comes from an edX MOOC hosted at a Midwest university in 2015. Out of the 14,809 students that participated in the course 1,941 completed the course, resulting in completion rate of 13.1% which is typical of MOOCs. Following the clickstream-based approach used by [16], we implement an automated machine learning based dropout predictor. This framework observes past student data every week and predicts whether a student s active in the MOOC during week i is likely to drop out in the next week $(i+1)$. Our **definition of dropout** is the same as [28], which combines participation and learning objective components of the MOOC. If a student has not finished the final "certification" module of the course and has no interaction with the MOOC between time t and the course-end date, then we say that the student has dropped out.

Given this pipeline, our approach helps interpret individual predictions and classification models. It provides an interpretation at each week, and longitudinally across weeks, creating a contextualized history of models and their predictions. Establishing a link between signals recorded in clickstreams and likely outcomes is extremely valuable to MOOC researchers and content creators. The models themselves can be used to describe profiles of successful students and can directly be fed into learning management systems. Various types of signals which predict dropout can be used to create tailor-made intervention strategies.

Contributions This paper's main contributions are in the form of answers to the following questions:

- *How can MOOC dropout prediction pipelines be made interpretable?* What is an interpretable model? What parts of the pipeline can we interpret and what are some current limitations?
- *What machine learning techniques does one need to consider?* We explore feature engineering, data preprocessing, model selection and metrics as design parameters in the context of interpretability.
- *Can this be generalized to other MOOCs?* We show how our analyses could be extended to MOOCs which differ from ours in format and platform (Coursera, Udacity, Udemy, etc).

2. RELATED WORK

MOOCs serve as an excellent testing ground for multidisciplinary research and have attracted the attention of experts in computer science, applied statistics, education, psychology and economics. This mixed expertise has made it possible to attack problems like low completion rates in MOOCs from various fronts. Some of the relevant components of the dropout prediction problem are described below:

Dropout Definition.

The broadest definition of dropout is the lack of engagement by students. Student engagement can be notoriously difficult to quantify [10] as a result, various proxy measures are used in literature. Definitions of dropout depend on the MOOC's intended pedagogical and engagement goals, and target student generated signals which capture these forms of engagement. Two distinct forms of engagement which influence dropout definitions are participation and completion of learning objectives. Lack of participation in MOOCs can be defined as a lack of interaction with the MOOC [2, 16], submission of assignments and quizzes [25, 27], viewing video content [24], or participating in discussion forums [31]. Some researchers focus on whether the student has achieved the learning objectives of the MOOC, defining dropout as not earning a certificate in the course [13] or not being able to finish a certain set of modules [7]. Hybrid definitions aim to capture both these components of engagement and label the absence on both fronts as "dropout" [12, 28]. One such example used in [12] is as follows: if a student has been absent from the course for more than 1 month and/or has

viewed fewer than 50% of the videos in the course, then they are labeled as dropouts. In this paper, we use the definition used by [28]: if a student has no interaction between day t into the MOOC and the end of the MOOC and they have not completed the final "certification" module, then they are labeled as dropouts.

Feature Selection and Engineering.

The electronic nature of MOOC instruction makes capturing signals of student engagement extremely challenging, giving rise to proxy measures for various use-cases. Clickstreams [16, 24, 27, 13], assignment grades [12, 14], social networks [2, 31] and even demographic information has been used in literature as potential sources of feature data to predict dropout. In more recent approaches like [28], data from across various MOOCs was pooled together to predict dropout using transfer learning. In this paper, we focus on dropout centered around a single MOOC using clickstream and video data. The idea behind using clickstreams is to reconstruct a story from the trail of digital breadcrumbs left behind by users. Most MOOCs contain video content and student interactions with these videos has been shown to quantify engagement and ultimately predict dropout [16, 24]. We deliberately restrict ourselves to features which are available across MOOC platforms, namely— access patterns, user agent data and video features.

Interpretability.

Most existing works and this paper structure MOOC dropout prediction as a supervised classification problem. Making these models and predictions interpretable entails challenges at several levels of the pipeline. Preprocessing steps like principal component analysis used in [16, 27] render otherwise interpretable input features uninterpretable. Prevailing solutions to making interpretable classification models is to use "interpretable" classification algorithms [1], or to be model-agnostic and analyze the predictions post-hoc. Interpretable classification techniques identified in [22] include linear models [13], decision trees [26], falling rule lists [30]. Examples of non-interpretable classifiers include recurrent neural networks [8] and stacking multiple classifiers [29]. On the other hand, model-agnostic interpretability is easier to integrate into existing pipelines using wrappers like Parzen [1] and Local Interpretable Model-Agnostic Explanations (LIME) [21]. These explain individual predictions by training locally interpretable models. The model agnosticity of these solutions helps them bypass the fidelity-interpretability trade-off. Student-centric stories like those shown in Figure 1 from [13] help illustrate their likelihood of dropout over time. As Figure 2 shows, we extend this metaphor by showing the feature values of a student in a given week which contribute towards their probability score.

3. DATA DESCRIPTION

"I Heart Stats" was an introductory statistics course offered by the University of Notre Dame. A major objective of this course was to alleviate student anxiety towards statistics [7]. This course was made available through edX[1] from April 15 to June 17, 2015 spanning a total of 64 days. We describe the format of how this course was offered and how this shaped our dropout definition. Throughout this process, we ensure reproducibility of our methods on other such MOOCs and generalizability beyond our chosen learning platform (edX).

[1]https://www.edx.org/ is a popular MOOC platform.

Table 1: Glossary of symbols

Symbol	Description
S_i	Set of students by week i
$S_i^{(i-1,i)}$	Set of students in week i also existing in week $i-1$
s_{ij}	The j^{th} student by week i
X_i	Feature data on S_i
$X_i^{(i-1,i)}$	Feature data on $S_i^{(i-1,i)}$
x_{ij}	Feature data for student j by week i
Y_i	Set of labels for students by week i
y_{ij}	Label of the j^{th} student by week i

3.1 Course Format

The course contained eight modules covering various introductory statistical topics from levels of measurement to ANOVA, designed to be completed in sequential order. However, since all the modules were released to the students from the first day of the MOOC, students had the choice to complete them at their own pace and in the order they chose. The key content in this course was delivered via 96 videos spread throughout the course. Under the Clark Taxonomy of MOOCs [5], this MOOC has traits largely similar to `asynchMOOCs`, in that students have the freedom to — 1) join and leave any time during the MOOC, 2) consume contents in the order they choose. One notable trait here is that the students can only access the MOOC during the window of a set start and end date, which is a trait of `synchMOOCs`.

How can we generalize this analysis to other MOOC formats? Even though this paper uses a largely `asynchMOOC` for its analyses, the steps described here can be easily generalized to MOOCs which follow a different format. We consider two alternative MOOC formats (not mutually exclusive) that represent a special case of the one followed in "I Heart Stats":

1. *Courses which require registration from the start of the course*— If we *only* consider the subset of students enrolled from day one in our MOOC, then it becomes analogous to such a format.

2. *Courses which do not disseminate all of their content from day one* and impose a set assessment schedule— If such a course follows the same definition of dropout as used in [28] and this paper then this paper's analyses can be used without modification. Typically, such courses use a dropout definition based on attrition instead [11]. We discuss how this can be treated as a modified version of our MOOC in Subsection 3.3.

3.2 edX Clickstream Data

Platforms like edX make course their data available to MOOC providers in the form of raw, queriable logs. Typically, these capture information on course content, learner progress, discussion forum interactions and (clickstream) tracking logs. In this paper, we use the "student_events" from the "tracking_log" data for feature extraction and "courseware_studentmodule" from the "learner progress" to label module (and therefore course) completion.

It should be noted that the clickstream data used for prediction in this paper does not include fields like IP address, location, ISP, referrers, landing search queries. These are typically available via proprietary tools like Adobe Analytics [17]. In this paper, we restrict our analysis to features extracted from video viewing patterns and device information. In line with most MOOCs, this course's offerings are in the form of videos. This is a basic proxy for student access patterns, how they interact with the course content and what kind of device experience they have when accessing the course.

How can we generalize this analysis to other MOOC platforms? Echoing the argument made by [16], we perform our analyses using a relatively small subset of available feature data to demonstrate the predictive value of clickstream data that is most commonly available across MOOC platforms. We deliberately select two of the most commonly available feature-sets that are available across most popular MOOC platforms. These features have been found in dropout prediction implementations in literature across platforms like edX [2], MITx [27], Harvardx [28] (both hosted on edX), and Coursera [16, 13]. We believe that these features can be similarly found in clickstream logs for other MOOC platforms not listed here.

3.3 Defining Dropout Labels

Our definition of dropout is a slightly modified version of [28]. Instead of a certificate, we use the completion of the final module as part of the course design to denote completion of learning objectives. As a result, our definition of dropout is as below—

> A student s has stopped out by time t if and only if: s does not complete the final module and s takes no further action between time t and the course-end date when certificates are issued

The learning objective component of this definition ensures that we correctly label students who complete the course's objectives as "Completers". This is a direct result of the course format offering students all of the content to study at their own pace. The participation component ensures that we correctly label students who do not interact with the course as "Dropouts". In alternative course formats where content is *not* distributed all at once, the special case of this definition becomes equivalent to [2], which is purely participation dependent.

4. MAKING THE DROPOUT PREDICTION PIPELINE INTERPRETABLE

We describe each component of the dropout prediction pipeline and discuss its role in our interpretablity goals. Using this pipeline, we train predictors each week to get models and predictions. We use this to create a longitudinal interpretation of how the models and predictions evolve in the MOOC. We take a deeper look into the interplay between these two phenomena to trace individual predictions back to student feature data.

4.1 Overview of the pipeline

Dropout prediction follows the simple idea of *learn on past student behavior to predict future outcomes*. This dictates the way authors have structured pipelines to approach this problem. Dropout prediction is an imbalanced class problem at heart and in keeping with machine learning convention, we label the minority class ("Completers") as 1 and the majority class ("Dropouts") as 0 throughout this paper.

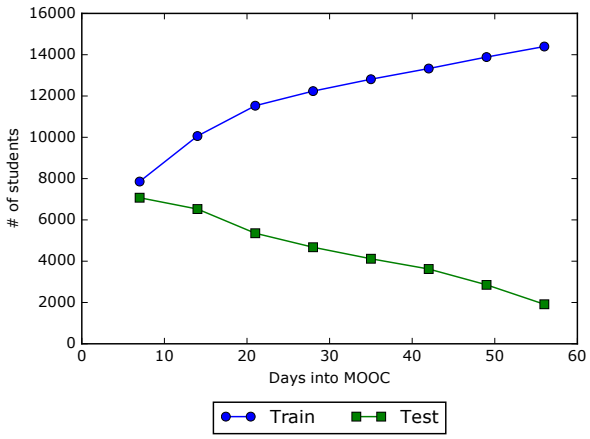


Figure 3: **Size of training and testing data at each stage**: We train on the cumulative number of students we have observed by each week, and so the number of students in the training data keeps increasing monotonically.

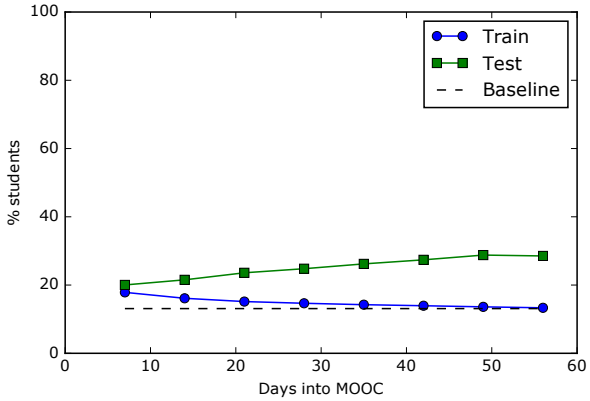


Figure 4: **Percent minority class (completers)** over time. Students who are active into the latter days of the MOOC are more likely to complete the course. This ratio for the training set approaches the overall completion ratio (black dashed line) towards the end of the MOOC.

Feature Engineering.

Even though we limit our treatment to video interaction and user-agent data, the feature engineering lessons here are generalizable to any other dataset which also uses these or similar variables. This data is recorded in the form of JSON-type events, which we then structure into columnar feature form as described below.

Video interaction data in edX is recorded for each user-generated event separately. Each row in the raw clickstream logs records actions like load, start, play, pause, stop, seek (from timestamp T_1 to T_2 in the video) and change of playback speed. We aggregate these into simple, interpretable features for each video_id-student_id pair: percent watched,

percent re-watched, total time of seeks, number of forwards, plays, seeks and speed changes. Each of these captures various latent factors in a student's consumption of the video content. This is still too granular for our study since we need features at the student level, so we aggregate these video_id-student_id pairs for each day of the MOOC. We average out the fractional quantities (percentage of video watched and rewatched) and add the rest of the variables which represent raw counts.

Clickstreams record user agent data as raw strings which need further processing to be made into features. Parsing this string returns categorical features which describe the device, browser and operating system in intricate detail. In addition to this parsed information, we would like to know if the device is a pc, tablet or phone and whether it is touch capable or not. These signals influence the degree to which the user is able to interact with the MOOC at large. These categorical variables are then encoded into one-hot encoded versions of their respective value. Similar to our treatment of video interaction features, we aggregate user_agent-student_id pairs by adding their absolute counts. So, if a student uses Chrome on a Windows PC and Chrome on an Android phone, their feature-space would record 2 Chrome browsers, 1 Windows OS and 1 Android OS.

Preprocessing.

Like most MOOCs, our dataset also suffers from class imbalance, where the number of dropouts far outweighs the number of completers. Class imbalance is a well documented problem in machine learning [4], where the presence of imbalance can bias a predictor to output the majority class. In dropout prediction, this can cause a classifier to aggressively predict that students are likely to dropout. This is especially dangerous for those students who in reality are completers, but are mislabeled by such a biased model as dropouts. The metrics and evaluation angle to this problem is discussed in greater detail in **Evaluation**. Though the severity of class imbalance has been recognized by MOOC researchers, measures to mitigate its effects in biasing predictors are surprisingly absent.

An effective technique to overcome this bias is to resample the data to alter the distribution of labels in the training set. Approaches range in complexity from randomly undersampling the majority class to creating synthetic instances of the minority class [3]. In educational data mining, these techniques have been proven useful when predicting dropout in Mexican high schools [19]. In this paper, we predict dropout both with and without random undersampling of the majority class to demonstrate the value of overcoming this bias.

Classification.

For each week (i), we train a classifier on X_{i-1} student data observed from the start of the MOOC till the end of week $i-1$ and observed labels Y_{i-1}. Cross validation of our trained model ensures that we do not overfit on our training set and can generalize to test data. We then predict whether students active at the beginning of week i ($S_i^{(i-1,i)}$) are likely to drop out. The outcomes of these students are used as true labels for evaluation at each stage and repeat the process for each consecutive week. In our pipeline, we compare examples of both interpretable and non-interpretable classification techniques. We select the two most directly interpretable classification techniques— Decision Trees and

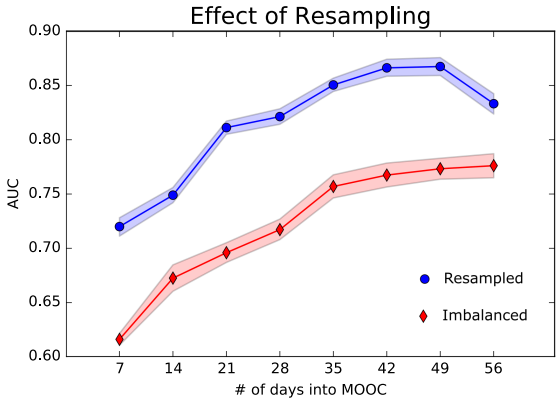


Figure 5: **Resampling data helps achieve better AUC**. It should be noted that the Y-axis is magnified for greater clarity.

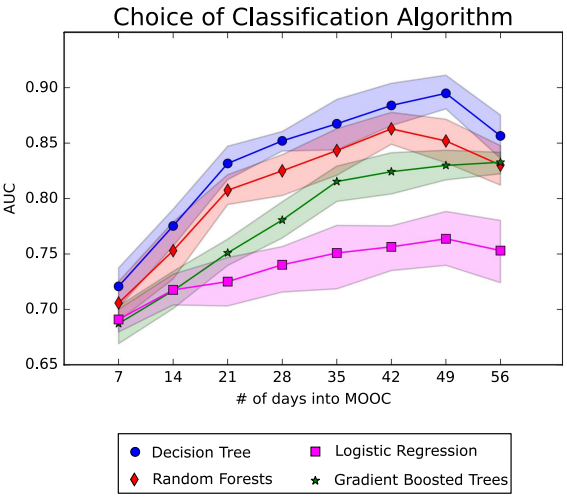


Figure 6: **Comparison across classification techniques on the same test data**. Significance testing done using confidence intervals of 99%.

Logistic Regression and pit them against non-interpretable techniques like Random Forest and Gradient Boosted Trees. For this example, our goal is to select the best model out of a set of classifiers, which is why we restrict this experiment to these algorithms. A detailed analysis of suitable alternative approaches is presented in Subsection 4.2.

Evaluation.

Perhaps the most notable aspect of imbalanced class prediction is how it is evaluated. Given an imbalanced class problem with 99:1 imbalance, a naive predictor which always predicts the majority class will effortlessly achieve 99% accuracy. In our MOOC, a simple classifier which predicts that everyone will drop out scores an accuracy of 86.9% (= 100% - minority class abundance). In light of the misleading nature of accuracy, MOOC researchers have borrowed several metrics used in imbalanced class learning and information retrieval. Several alternatives include Specificity [12], Recall [12], Kappa Statistic [24] and Area Under the ROC

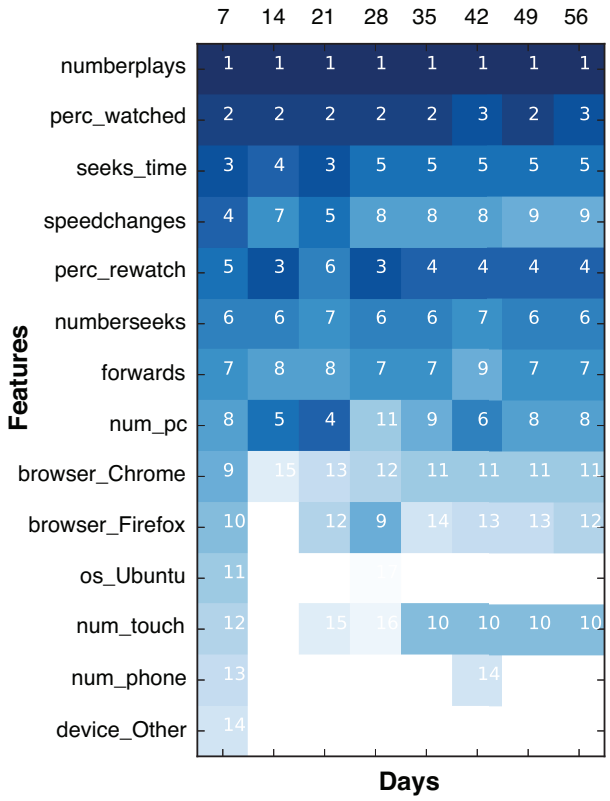


Figure 7: **Model Interpretability— Rank ordering Feature Importances**. Decision trees and ensembles using decision trees as base estimators can be inspected using their feature importance scores. This colormap charts their relative ranks across weeks for Random Forests.

Curve (often abbreviated as "AUC") [28, 13]. In the interest of comparison with existing literature, we use AUC as our metric of choice in this paper. A perfect predictor would score 1.0, whereas a random predictor would score 0.5 on AUC. The AUC values are computed for each week, on all classifiers and summarized in Figure 6. Now that we have a set of classifiers and their respective performance scores, we find out *which* model is the most suitable in a given dropout prediction pipeline.

Model Selection.

We identify two key traits which guide our model selection process—1) consistently high predictive performance and 2) (in case of a tie in 1),...) greater interpretability. This can be answered using the AUC scores and the significance analysis illustrated in Figure 6. Decision Trees perform better than, or equivalent to competing techniques consistently and emerge as the classifier of choice on both of our desired criteria. We note that a singular algorithm may not always outperform competing techniques at all points in the MOOC. These predictive models are based on a small subset of features available to competing techniques, and yet they come within striking distance of AUC scores like 0.88 [27], 0.87 [13] and 0.850 (AT1x) to 0.907 (SW12.2x) [28].

4.2 Model Level Interpretability

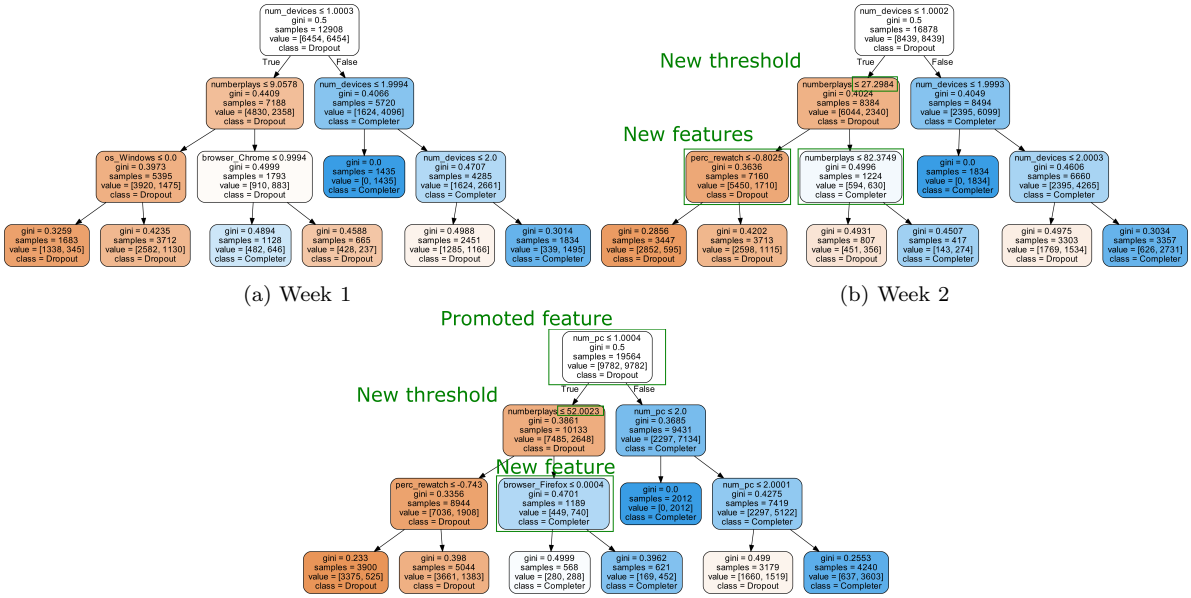
Interpretability in models can be approached in three ways using— 1) intrinsically intepretable models, 2) partially interpretable models and 3) after-the-fact approaches which treat the classifier as a black-box. Since these operate at very different stages in the prediction process, these are not necessarily mutually exclusive.

Intrinsically Interpretable.

We make a case for readily interpretable classification algorithms Decision Trees and Linear Models and describe the steps to draw inferences from them [9]. Decision trees encode multiple sets of rules in the form of hierarchical nodes which subset the instance space till a stopping criteria is achieved. Each node is a logical test on a feature value and each path-to-leaf represents a distinct "rule" followed exclusively by the instances in that leaf. Decision Trees for dropout prediction resemble the one shown in Figure 8, where values of engagement define dropout-intensifying or completion-intensifying signals. In the interest of being interpretable, these trees are often pre-pruned to be limited in their depth. The decision trees used in this paper are similarly pre-pruned to a depth of 5. However, for brevity of illustration, we demonstrate 3 consecutive trees limited at depth 3 in Figure 8.

Immediate observations can be drawn from this figure at each week and across weeks. Three distinct profiles of Completers exist at Week 3, these are inferred from paths to leaves indicating label = Completer. These profiles are 1) "Students who log in through one or no PCs, hit play more than 52 times on a video and do not use Firefox", 2) "students who use 1 or 2 PCs" and 3) "students who use more than 2 PCs". The last two demographics can be merged together into one segment: "students who use 1 or more PCs". Across the weeks, the changes in the topology and node values of the tree indicate a change in the behavior of students who drop out. By Week 2, it does not matter if the student uses Windows OS or Chrome as their browser, instead these are replaced by how much percent on average they rewatch video content. In Week 3, we see that Firefox usage indicates Completion in students with 1 or no PC with more than 52 video play interactions and the total number of devices does not matter.

Similar inferences can be drawn for Logistic Regression models shown in Figure 2. At the model level, instead of features values, the respective feature coefficients and intercept term are of interest. The intercept term is the baseline tendency of the linear model to predict a given class in absence of any signal, which here for the first 3 weeks points towards Dropout consistently. On the other hand, the magnitude of coefficient values indicates their relative importance and their sign (positive or negative) corresponds directly with the predicted classes (Completers or Dropouts, respectively). So, in the first week, Completers are likely to be associated with a high number of PCs logged in through and high number of video play interactions. Over time, the signal from the number of times a student hits play in videos intensifies in its effect on Completion tendencies, and the number of PCs becomes less important. During Week 3, a new signal becomes important in reinforcing Dropout behavior, i.e. the percentage of video content re-watched. This signal at Week 3 is useful to MOOC researchers by being a direct link to content consumption.



(a) Week 1 (b) Week 2

(c) Week 3

Figure 8: **Decision Trees over the first 3 weeks in the MOOC**. Changes in the structure are a direct reflection of dropouts and completions in the MOOC.

These extremely detailed insights from model inference exhibit the power of using Decision Trees and Sparse Linear Models. Their predictive performance from Figure 6 further supports their predictive power. Other approaches like Interpretable Decision Sets are designed with the primary purpose of being interpretable [18], but may not always be easy to integrate into existing pipelines.

Partially Interpretable.

A possible way to achieve partial interpretability in Random Forests and other ensemble approaches is to visualize each constituent classifier, or to characterize them in terms of their feature importances. Typically, an ensemble can have hundreds, if not thousands of classifiers and as such make it tedious to interpret each component every week, which makes feature importances a more convenient choice. A compact metaphor to visualize feature importances is shown in Figure 7 in the form of a colormap. Though we cannot make granular inferences like we did from Figure 8, we can still comment on the broad-stroke role the features play in dropout prediction. This high-level treatment still echoes lessons learned from Logistic Regression (Figure 2) and Decision Trees (Figure 8). In both Logistic Regression and Decision Trees, we see that the percentage of rewatched videos starts gaining importance from Week 2 and 4 onwards.

Interpreting the uninterpretable.

Model-agnostic black-box approaches operate after-the-fact on trained models. They approximate the behavior of the original classifier using an interpretable classifier and provide an interpretability layer, much the same as this paper aims to do with dropout detection as a whole. Popular approaches include decision-tree approximations [6], linear-model approximations [21], and perturbation based model inference [1]. In cases where it is impossible to swap out the uninterpretable predictor, we recommend that these wrapper-based approaches be used.

4.3 Instance Level Interpretability

The goal of instance level interpretability is to identify key feature values which contributed towards a predicted outcome for a given instance. Readily interpretable models give rise to readily interpretable predictions. In the case of Decision Trees, the path-to-leaf in a decision tree is the explanation of an instance's prediction and the terminal leaf node is the predicted outcome label. Linear Models can be interpreted by multiplying feature values with the learned coefficients and adding the intercept. The highest magnitude of coefficient multiplied by feature value is the most informative of an instance's prediction. Other instance-level interpretations are discussed in [9].

In the interest of being an interpretability *layer*, this paper uses Locally Interpretable Model-Agnostic Explanations (LIME) [21], which, as the name suggests is a model-agnostic wrapper technique. LIME trains an interpretable linear approximation of the black-box classifier around the test instance. It then returns the approximate linear model coefficients, intercept and probability scores to offer an explanation that is faithful to the black-box model for that given instance. We use LIME at each stage of the dropout prediction pipeline to provide explanations for *any* student's predicted outcome from *any* classifier model.

Using LIME on a given student's dropout predictions over time gives us probability scores for likelihood of dropout and possible linear model explainers. Figure 9 shows how we can extract a student-centric storyboard from these explained predictions. From the probability score, we see that the student initially only shows leanings towards completion and then exhibits stronger signs of completing the course. Barring a minor hiccup in Week 2, this student's comple-

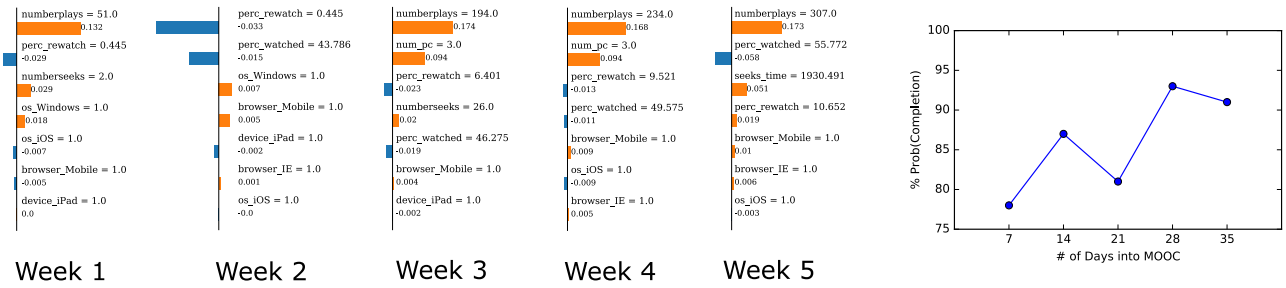


Figure 9: **Longitudinal Explainers throughout the MOOC for a sample student with their predicted probabilities of Completion**. We see the various feature values specific to this student who persists through the MOOC till Week 5 and completes the course.

tion is largely owing to adherence to watching videos (the high number of video play events points to this). This student tends to watch a small portion of the video content and hence rewatches a certain portion of those videos. This nudges the probability score away from Completion, but not enough to lower it enough to predict Dropout.

5. CONCLUSION

Our paper is a first-step in the direction of incorporating interpretability in MOOC dropout prediction. We realize that the problem of MOOC dropout has its roots in student engagement and cannot be studied in isolation, spurring the need to answer the much needed *why?* in student dropout.

How can MOOC dropout prediction pipelines be made interpretable? An interpretable pipeline needs to begin with interpretable features. Features which encompass elements of student engagement need to be chosen such that they are available for the entire set of students we wish to predict on. Preprocessing steps like PCA which obfuscate interpretability should not be used. Given that student dropout is an imbalanced class problem, resampling techniques help alleviate its negative effect on prediction.

What machine learning techniques does one need to consider? We show how model selection should be implemented to select the best possible model at each stage. We provide guidelines and metrics to identify the best mix of interpretable and high-performance models. Interpretability can be introduced using various approaches. We demonstrate implementations ranging from intrinsically interpretable to black-box model inference approaches to interpret models and predictions.

Can this be generalized to other MOOCs? We show how our analyses could be extended to MOOCs which differ from ours in format and platform (Coursera, Udacity, Udemy, etc).

Our solution to interpretable dropout prediction pipelines enables us to analyze both models and predictions longitudinally. In the future, we aim to expand this research to evaluating its effectiveness on intervention efforts in a live MOOC.

6. ACKNOWLEDGMENTS

This research was supported in part by the NSF Grant IIS-1447795.

7. REFERENCES

[1] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. MÃžller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.

[2] G. Balakrishnan and D. Coetzee. Predicting student retention in massive open online courses using hidden markov models. *Electrical Engineering and Computer Sciences University of California at Berkeley*, 2013.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[4] N. V. Chawla, N. Japkowicz, and A. Kotcz. Editorial: special issue on learning from imbalanced data sets. *ACM Sigkdd Explorations Newsletter*, 6(1):1–6, 2004.

[5] D. Clark. Moocs: taxonomy of 8 types of mooc. *Donald Clark Paln B*, 2013.

[6] M. W. Craven and J. W. Shavlik. Extracting tree-structured representations of trained networks. *Advances in neural information processing systems*, pages 24–30, 1996.

[7] J. Dillon, N. Bosch, M. Chetlur, N. Wanigasekara, G. A. Ambrose, B. Sengupta, and S. K. D'Mello. Student emotion, co-occurrence, and dropout in a MOOC context. *The 9th International Conference on Educational Data Mining*, pages 353–357, 2016.

[8] M. Fei and D.-Y. Yeung. Temporal models for predicting student dropout in massive open online courses. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 256–263. IEEE, 2015.

[9] A. A. Freitas. Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter*, 15(1):1–10, 2014.

[10] P. J. Guo, J. Kim, and R. Rubin. How video production affects student engagement: An empirical study of mooc videos. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 41–50. ACM, 2014.

[11] C. Gütl, R. H. Rizzardini, V. Chang, and M. Morales. Attrition in mooc: Lessons learned from drop-out students. In *International Workshop on Learning Technology for Education in Cloud*, pages 37–48. Springer, 2014.

[12] S. Halawa, D. Greene, and J. Mitchell. Dropout prediction in moocs using learner activity features.

Experiences and best practices in and around MOOCs, 7:3–12, 2014.

[13] J. He, J. Bailey, B. I. Rubinstein, and R. Zhang. Identifying at-risk students in massive open online courses. In *AAAI*, pages 1749–1755, Austin Texas, USA, 2015. AAAI Press.

[14] G. Kennedy, C. Coffrin, P. de Barba, and L. Corrin. Predicting success: how learners' prior knowledge, skills and activities predict mooc performance. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 136–140, New York, NY, USA, 2015. ACM, ACM New York, NY, USA 2015.

[15] H. Khalil and M. Ebner. Moocs completion rates and possible methods to improve retention - a literature review. In J. Viteli and M. Leikomaa, editors, *Proceedings of EdMedia: World Conference on Educational Media and Technology 2014*, pages 1305–1313, Tampere, Finland, June 2014. Association for the Advancement of Computing in Education (AACE).

[16] M. Kloft, F. Stiehler, Z. Zheng, and N. Pinkwart. Predicting mooc dropout over weeks using machine learning methods. In *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*, pages 60–65, Doha, Qatar, 2014.

[17] B. Kocol. Web page link-tracking system, Mar. 10 2009. US Patent 7,502,994.

[18] H. Lakkaraju, S. H. Bach, and J. Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1675–1684, New York, NY, USA, 2016. ACM.

[19] C. Márquez-Vera, A. Cano, C. Romero, A. Y. M. Noaman, H. Mousa Fardoun, and S. Ventura. Early dropout prediction using data mining: a case study with high school students. *Expert Systems*, 33(1):107–124, 2016.

[20] D. F. Onah, J. Sinclair, and R. Boyatt. Dropout rates of massive open online courses: behavioural patterns. *EDULEARN14 Proceedings*, pages 5825–5834, 2014.

[21] M. T. Ribeiro, S. Singh, and C. Guestrin. " why should i trust you?": Explaining the predictions of any classifier. *arXiv preprint arXiv:1602.04938*, abs/1602.04938, 2016.

[22] M. T. Ribeiro, S. Singh, and C. Guestrin. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*, 2016.

[23] R. Rivard. Measuring the mooc dropout rate. *Inside Higher Ed*, 8:2013, 2013.

[24] T. Sinha, P. Jermann, N. Li, and P. Dillenbourg. Your click decides your fate: Inferring information processing and attrition behavior from mooc video clickstream interactions. *arXiv preprint arXiv:1407.7131*, 2014.

[25] R. M. Stein and G. Allione. Mass attrition: An analysis of drop out from a principles of microeconomics mooc. *Social Science Research Network*, pages 1–19, 2014.

[26] J. K. Tang, H. Xie, and T.-L. Wong. A big data framework for early identification of dropout students in mooc. In *International Conference on Technology in Education*, pages 127–132. Springer, 2015.

[27] C. Taylor, K. Veeramachaneni, and U.-M. O'Reilly. Likely to stop? predicting stopout in massive open online courses. *arXiv preprint arXiv:1408.3382*, 2014.

[28] J. Whitehill, J. J. Williams, G. Lopez, C. A. Coleman, and J. Reich. Beyond prediction: First steps toward automatic intervention in mooc student stopout. *Social Science Research Network*, 2015.

[29] W. Xing, X. Chen, J. Stein, and M. Marcinkowski. Temporal predication of dropouts in moocs: Reaching the low hanging fruit through stacking generalization. *Computers in Human Behavior*, 58:119–129, 2016.

[30] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2(3):5, 2015.

[31] D. Yang, T. Sinha, D. Adamson, and C. P. Rose. Turn on, Tune in, Drop out: Anticipating student dropouts in Massive Open Online Courses. In *Proceedings of the 2013 NIPS Data-driven education workshop*, volume 11, page 14, Lake Tahoe, Nevada, USA, 2013.