Evaluating Predictive Models of Student Success: Closing the Methodological Gap

Josh Gardner
School of Information
The University of Michigan
jpgard@umich.edu

Christopher Brooks
School of Information
The University of Michigan
brooksch@umich.edu

January 26, 2018

Abstract

Model evaluation – the process of making inferences about the performance of predictive models – is a critical component of predictive modeling research in learning analytics. In this work, we present an overview of the state-of-the-practice of model evaluation in learning analytics, which overwhelmingly uses only naïve methods for model evaluation or, less commonly, statistical tests which are not appropriate for predictive model evaluation. We then provide an overview of more appropriate methods for model evaluation, presenting both frequentist and a preferred Bayesian method. Finally, we apply three methods – the naïve average commonly used in learning analytics, frequentist null hypothesis significance test (NHST), and hierarchical Bayesian model evaluation – to a large set of MOOC data. We compare 96 different predictive modeling techniques, including different feature sets, statistical modeling algorithms, and tuning hyperparameters for each, using this case study to demonstrate the different experimental conclusions these evaluation techniques provide.

1 Introduction

The past decade has seen an explosion in the use of data science methods in general, and predictive modeling in particular, across nearly every domain. Learning analytics itself is, at least in part, an outgrowth of the spread of predictive modeling, where the increased use of digital learning tools and the widespread adoption of Student Information Systems (SIS) has made it easier to replicate, transfer, and analyze data.

Predictive modeling of student success has become a central task in learning analytics research. A recent survey by Hu, Cheong, Ding, and Wu examined 39 works published between 2002 and 2016 which developed prediction models for student learning outcomes and tested their predictability on empirical data

[27]. In the current work, we survey more than 80 peer-reviewed works which construct and evaluate predictive models of student success in MOOCs.

Predictive modeling in learning analytics typically consists of the following steps:

1. **Feature Extraction**: raw data (i.e., clickstream logs or database tables) are processed to extract structured data suitable for predictive modeling. Selection of informative features (also called *feature engineering*) is regarded as one of the most critical elements in constructing effective predictive models.

2. **Algorithm Selection**: either before or after feature extraction, appropriate statistical models are identified (i.e., logistic regression, decision trees, etc.). "Appropriate" here can refer to the goals of the research (i.e., interpretable models which provide estimates of the magnitude of an effect or decision rules) or to other desiderata, most often including predictive accuracy.

3. **Hyperparameter Tuning**: parameters which control model fit are selected and tested (i.e., the number of nodes in a neural network).

4. **Model Evaluation**: From the full set of $featureextraction \times algorithm \times hyperparameter$ examined, the optimal model is selected according to some objective function (typically, each model is evaluated on either the training data or unseen test data).

In this work, we focus on the final step of this pipeline, the task of model *evaluation* or model *selection*. This is a critical stage of any predictive modeling experiment, and is often the "headline" finding. However, we argue that there is a significant methodological gap in current practice in the learning analytics and educational data mining communities with respect to model selection. In particular, we take issue with the lack of statistical testing when comparing models and making a final recommendation or decision as to the best performing one. In an effort to advance the field's methodology with respect to these decisions, this work attempts to answer the following questions:

- What is the current practice in the learning analytics and related (i.e., educational data mining) communities with respect to predictive model evaluation and selection?

- What procedures are effective (and ineffective) for predictive model evaluation, according to the broader machine learning and statistical model evaluation literature?

- How do different procedures, applied to experimental data in learning analytics, produce different conclusions?

In this paper, we first present data from a comprehensive literature review of MOOC research to establish the "state of the practice" in the field in Section 2. In Section 3, we present a brief overview on methods for model evaluation from the machine learning and statistical research communities, with an emphasis on (a) methods widely used in the learning analytics field based on our findings in Section 2, many of which are not suited for the settings in which they are used, and (b) methods which are most effective for common model evaluation tasks in learning analytics. In Section 4, we conduct a predictive modeling experiment on a large sample of MOOCs, applying three techniques to demonstrate different kinds of inferences one can make to highlight the importance of constructing knowledge using statistically rigorous, well-calibrated methods that are suited to predictive model evaluation and robust to multiple comparisons. Our results also demonstrate specific empirical findings which are relevant to future modeling research in MOOCs. We provide conclusions with an eye to practice within the LA and EDM fields in Section 6.

2 State of the Practice: A Survey of Model Evaluation in Learning Analytics

2.1 Predictive Modeling in MOOCs

Prior work on predictive modeling in MOOCs has evaluated a wide variety of outcomes, data sources, feature extraction methods, and modeling algorithms.

Outcomes for measuring student success are as diverse as learners' goals and intentions for taking MOOCs, but commonly include dropout [52], pass/failure [9, 23], completion/certification [32], assignment and exam scores [49, 43], or career-related objectives [53].

The data sources used for predictive modeling of these outcomes typically consist of a subset of the data collected by the MOOC platform directly. The most common data source used is platform clickstream exports, which record each user interaction with course resources (page or discussion forum viewing, video play/pause/rewind behavior, assignment or quiz submission, etc.). Raw clickstream logs are typically large text files in JavaScript Object Notation (JSON), with each line representing an individual interaction. An ordinary MOOC clickstream file contains millions of interaction-level records. Raw clickstream records are not suitable for use as inputs to most predictive models; preprocessing and extraction of structured data from the clickstream file is typically required. This process of extracting structured information from raw data is known more generally as *feature extraction* or *feature engineering*.

Predictive modeling experiments in MOOCs have used a wide variety of feature engineering approaches. Feature extraction is considered one of the most important, and also one of the most difficult, tasks in predictive modeling in MOOCs [52, 48, 36, 44, 41, 48], and methods for feature extraction are commonly highlighted as one of the primary innovations of published experiments. Often, these features are derived from raw platform data (either click-

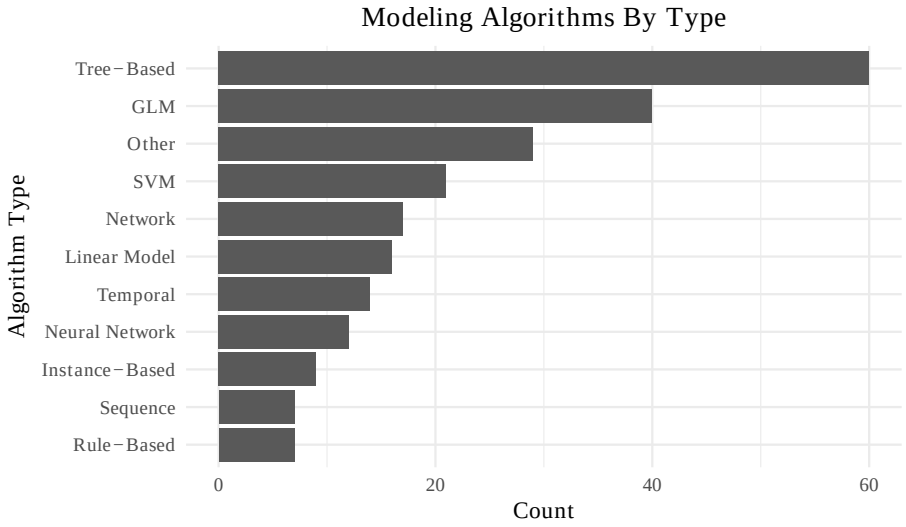

Figure 1: Modeling algorithms used in works surveyed by model type. Where multiple algorithms were used, each algorithm was counted.

stream files, discussed above, or other database exports provided by MOOC platforms, including detailed information about forum posts, course resources, assignments, learner surveys, and course metadata). These features range from simple counting-based features [58, 33]; natural language processing metrics [44, 55, 14], social networks or subcommunities [45, 8, 46], demographic information [10], academic performance on course assignments [34, 3], data about learners' emotional states [18], and other information.

From these feature sets, most experiments construct multiple statistical models and report their performance. There is no broad consensus in practice regarding which algorithms are used to construct these models, though it is not unusual for a given work to suggest one algorithm is "better" than others for a given task and feature set. In Figure 1, we show the most common broad categories of algorithms used acoss our survey of MOOC research, and note that the most common statistical modeling techniques include tree-based methods (including bot simple decision trees and ensembles of trees, such as boosted trees and random forests), logistic regression (both with and without regularization), support vector machines, and network-based models (such as Naïve Bayes and Bayesian Networks), with several other algorithms being popular as well.

Each of the algorithms listed above typically requires selecting and, optionally, tuning hyperparameters which control elements of model fit, such as parameters related to cost, convergence, and feature selection or regularization. Our survey finds that in practice, hyperparameter tuning is often not described in published research; in 19 out of 85 (22%) of the work surveyed, methods for

hyperparameter tuning or selection were not reported or even mentioned, and in a remaining 7 out of 85 (8%), hyperparameter tuning was reported as being performed manually, often with no description of the full range of comparisons conducted an no reproducible procedure offered. In cases where hyperparameter is not reported, either the hyperparameter tuning is (a) simply not performed, and default settings are used; (b) performed by the experimenter, but not reported, or (c) performed automatically by a machine learning package (such as Weka, RapidMiner, caret in R, or scikit-learn in Python). This is important because, as we will describe in detail below, cases (b) and (c) require model evaluation methods which are robust to multiple comparisons even *without* knowing or reporting how many comparisons are conducted in an experiment. In case (a), we might question whether an experiment sufficiently explored the performance of a given modeling approach. In any case, we argue that hyperparameter tuning is a critical, but often ignored, element of predictive model evaluation. Additionally, when an experiment considers hyperparameter tuning *and* algorithm selection *and* multiple feature extraciton methods or data sources, the number of potential models considered by the experiment increases multiplicatively, as our experiment in Section 4 demonstrates in practice.

The brief survey in this section demonstrates the broad array of techniques used for the first three components of the predictive modeling pipeline (feature extraction, algorithm selection, and hyperparameter tuning) in MOOCs. Given the breadth of techniques that have already emerged, the need for effective model evaluation can be considered particularly urgent.

2.2 Model Evaluation in MOOCs

The methodological consensus on techniques for predictive model evaluation in MOOCs is surprisingly strong, considering the breadth of approaches taken to the rest of the predictive model-building pipeline described above. A predictive modeling technique must choose a *prediction architecture* – the procedure by which training and testing datasets are partitioned and model predictions are obtained – in order to evaluate the statistical models of interest. Results of the prediction architectures used in the works included in our literature survey ($N = 85$) are shown in Figure 2.2.

These results demonstrate the prevalence of various forms of cross-validation in evaluating models (typically 2-, 5-, or 10-fold cross-validation; only two experiments surveyed use leave-one-out cross-validation). In almost all cases, this cross-validation is performed on a single course dataset and average performance reported; we found no cases, for example, where fold-level data was reported or evaluated (for example, to produce estimates of the variance of model performance across each fold). Cross-validation by itself presents no major methodological concerns with respect to model selection (although it may generate "optimistically" biased model performance estimates [21]). The use of cross-validation particularly relevant in light of concerns about statistical testing when applied to draw inferences from cross-validated model performance data, discussed in Section 3. Additionally, we note that in nine experiments (nearly

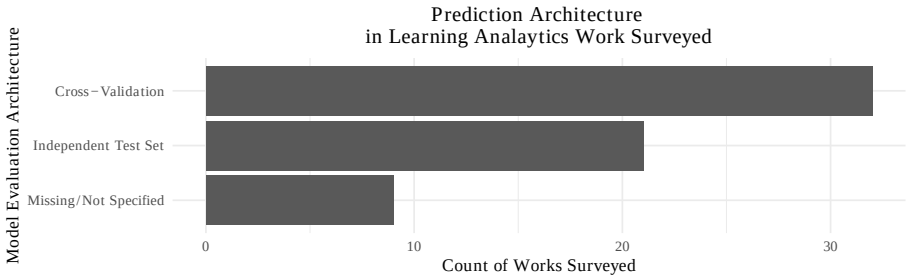

Figure 2: Prediction architecture in works surveyed; this is the method by which estimates of model performance on unseen data are obtained (for further analysis, i.e., by applying statistical tests do the data). Cross-validation is the most common technique and is used in over one third of the work surveyed. Note that in nine works, the evaluation architecture was not reported at all, despite predictive results being reported.

10% of the experiments surveyed), the model evaluation method was not reported, despite the presentation of predictive results (i.e., accuracy of multiple models) – recall that all of the work surveyed appeared in peer-reviewed publications. This further demonstrates the lack of emphasis on model evaluation in predictive models of student success.

In addition to selecting a prediction architecture, an experiment must also choose a model *evaluation* method, whereby the observed differences in performance on the test data are formally evaluated or tested. We present data on model evaluation methods in Table 1.[1] We note that the data in this table reflects any experiments where multiple predictive models were evaluated and compared, and where the experimenters reported some conclusion about which model was a "better" predictor of the outcome of interest; strictly explanatory modeling experiments where an evaluation of the model's predictive accuracy is secondary to inspection of the model itself (although predictive accuracy is one way to assess model fit in explanatory experiments) are included in the "No model comparison or explanatory" column. While our categorization of which experiments should be considered "explanatory" versus those which compare models based on their predictions cannot be perfect based only on the published descriptions of work, the results in Table 1 make clear the dearth of *any* significance testing in most published predictive modeling research in learning analytics. Additionally, because all of the works reviewed were peer-reviewed, these results also reflect the consensus of the reviewers, and the field as a whole. This review included work from several of the field's flagship journals and conferences (Computers and Human Behavior; Journal of Educational Data Mining; International Conference on Learning Analytics and Knowledge;

[1]For the complete results of this literature review, with citation and categorization information for each study evaluated, see [24].

Student's t-test	5
Other NHST (Chi-Square, etc.)	5
No model comparison or explanatory	27
Model comparison with no statistical test	50
Total	83

Table 1: Testing procedures for model evaluation in works surveyed.

The International Conference on Educational Data Mining; Learning at Scale; etc.). We can clearly conclude that rigorously testing and evaluating predictive models is currently *not* considered a necessary component of scientific practice for predictive modeling researchers in learning analytics and educational data mining.

In particular, statistical model evaluation using appropriate procedures (i.e., hypothesis-based testing procedures using well-calibrated statistical tests, or model-based evaluation procedures), is important to scientific inquiry for several reasons:

1. **Guard against spurious results**: Formal statistical testing and the concept of a p-value can be traced back more than a century to the work of Fisher [20] and Pearson [42]. Fisher introduced hypothesis testing to provide an objective approach to inductive inference [28]. This null hypothesis significance testing (NHST) was developed to provide an evidentiary basis for differentiating between results which were spurious and simply due to chance, and those which were likely to be genuine effects, over the course of multiple experiments or observations (of course, this approach is based on several assumptions, such as the truth of the null hypothesis, which we discuss in Section 3). Other model-based methods, such as the use of Generalized Linear Models and Bayesian models, are also used for this purpose. Despite differences in the inferences we draw from these different procedures, using *no* method for evaluating the results of an experiment provides no basis for concluding whether an observed result reflect some true relationship in nature, or whether it is simply due to chance or stochastic error.

2. **Quantify the confidence of results**: Statistical testing procedures provide typically provide information about the confidence of a result, or the degree to which we expect it to be true (conditional on certain assumptions, such as the truth of the null hypothesis, or the data we observed). This is largely the method by which (1) above is accomplished (the measures of confidence, such as a p-value or Bayesian posterior probability, are used to determine whether a results of a comparison are likely to be spurious by comparing them to a predetermined threshold).

3. **Provide a basis for comparison across experiments**: Statistical testing procedures usefully allow us to compare and (potentially) reconcile

the results of different experiments, guiding the process of scientific inquiry and consensus formation. For example, two studies that find small, but statistically insignificant effects in opposite directions can be compared on the basis of the significance and magnitude of their effects (and might not be particularly concerning where the confidence is low or the observed effect sizes small). Without any evaluation of the observed results of an experiment, the scientific community simply has no basis for comparison beyond prior knowledge or assumptions.

This list is not meant to be exhaustive. Instead, we hope it simply makes clear to the reader that statistical testing is good scientific practice and necessary to the construction of robust scientific knowledge.

The consensus on techniques for predictive model evaluation in the learning analytics and educational data mining research communities is somewhat surprising. Cross-validation with no statistical testing is overwhelmingly adopted in the LA/EDM predictive modeling research, to the point where the results of this procedure are considered sufficiently acceptable by authorities in the field to be published widely. We have to wonder whether this would also be the case for non-predictive model comparisons – would an experiment which compared two subpopulation means and identified one as "better" without looking at the distribution of differences through a t-test (or another appropriate method) be accepted for publication? We highly doubt that it would be. Yet this is essentially the same issue we face when comparing predictive models – and many papers compare large numbers of models across many different configurations. This issue has not gone unnoticed by the broader machine learning community, and in the next section we describe approaches used by this community which may be suitable for predictive modeling in learning analytics and educational data mining.

3 Prior Research: Predictive Model Evaluation and Selection

In this section, we survey the broader research on the statistical evaluation of predictive models in order to demonstrate that there is a robust theoretical and empirical research base identifying both effective and ineffective methods for predictive model evaluation. Our main goal in this section is to highlight the most effective methods for specific tasks common to learning analytics research, and to identify and weigh the criteria for using these various procedures where there are multiple useful or effective methods for a given task. We also outline methods which have been demonstrated to be *ineffective* for predictive model evaluation, focusing on methods which were implemented in experiments surveyed in our literature review. We conclude by advocating a Bayesian procedure which is preferable for several reasons, particularly its ability to estimate the *posterior* probability of an observed result (a theoretical/inferential benefit) and its robustness to multiple comparisons (a practical benefit), but we argue

that adopting any of the effective procedures outlined below would advance the scientific practice of the field of learning analytics.

3.1 Model Evaluation: Naïve Average Method

The most common technique for comparing the performance of predictive models in the fields of learning analytics and educational data mining is what we refer to as the "naïve average" method. In this approach, models are evaluated by comparing their average performance. As we show in Table 1, 50 out of the 85 studies (59%) surveyed conducted model comparison with no statistical test, using the naïve average approach, despite attempting to draw inferences about some type of comparison between multiple predictive models (either implicitly, by presenting the predictive results from several models, or explicitly, by referring to models as "more accurate" or having the "best performance"). In any experiment where multiple statistical models are being compared, an appropriate statistical evaluation method is necessary to produce accurate inferences about observed model performance. Drawing inferences about which model may be "best" based on a simple sample average is inappropriate for machine learned models the same way it is inappropriate for comparing any other average, without statistical testing – it provides no context for this comparison, failing to account for variability in the observed data, the magnitude of the observed difference, and the size of the sample, among other contextual factors. We call this method the "naïve average" method because, by taking the average and simply choosing the "best" average performance, this approach naïvely assumes that any differences in average performance observed must be (a) due to genuine differences in model performance (and not, for instance, random variation), and (b) that these differences must be (in frequentist terms) statistically significant, and large enough to be important.

Indeed, making objective comparisons like these are precisely why statistical testing was developed – to draw principled, reliable inferences about data under uncertainty. This is particularly important when evaluating the complex performance data from predictive models of student success, which itself reflects underlying samples of student populations, randomized resamples of subpopulations (for instance, via cross-validation), and other stochastic procedures inherent to many modeling algorithms (such as random feature selection or parameter initialization methods).

Furthermore, averaging itself – even without the use of statistical testing – may be uninformative or misleading. Demsar notes that "[i]f the results on different data sets are not comparable, their averages are meaningless" [15]. In the case of predictive models of student success, there might be good reason to think that performance across different datasets would not be comparable: some courses might be easier or more difficult to predict on for a variety of reasons, including different student subpopulations, varying level of difficulty and quality of instruction,, different course durations and requirements, etc. If any of these factors make a substantial difference to the accuracy of predictive models, then simply averaging performance across many datasets might be especially

misleading.

Averages are also susceptible to outliers, which particularly distort experimental results when relatively small populations are used (as is the case in our review, where nearly 50% of studies evaluated only a single course).

We believe that at least one reason the naïve average approach has gained wide acceptance despite its shortcomings is that it quickly narrows the field of potential "best" models in an experiment to one – the model with the best overall average performance (assuming the absence of any ties). In Section 4, we recognize that this is a common use case, and we demonstrate that a Bayesian procedure in particular can identify a small (4 out of 96) family of best models with high confidence while using a data-driven and inferentially sound procedure.

A second potential reason the naïve average approach might be preferred is that it is computationally efficient. However, none of the procedures considered here are particularly computationally expensive and are tractable using a modern laptop computer. For the 4560 pairwise comparisons in the case study below, the NHST-based method discussed below executes in seconds, and the Bayesian method executes in minutes.

We therefore see no reason for the continued use of the naïve average method for model evaluation in learning analytics–all this approach does is muddy the waters and contribute to uninformed decisions about the "embarrassment of riches" available to researchers interested in constructing predictive models of student success.

3.2 Model Evaluation: NHST-Based Methods

In this section, we discuss model evaluation procedures based on null hypothesis statistical testing (NHST). As Table 1 demonstrates, this is the second most common family of evaluation procedures used to evaluate machine learned models in learning analytics.

3.2.1 Two Models, Single Domain

NHST procedures have a relatively long history of being applied for the evaluation of predictive models. In particular, a broad array of statistical techniques have been examined for the task of comparing two learning algorithms over a single domain or dataset. One of the earliest and most authoritative treatment of several NHST-based model evaluation methods is Dietterich [17], which evaluates several statistical testing and resampling procedures in the context of predictive model evaluation. Dietterich evaluated several evaluation procedures based on the t-test commonly used in learning analytics, including (i) a paired t-test applied with resampling (where repeated random subsampling is used to create train/test partitions from a dataset $\mathcal{D}$), (ii) paired t-test with 10-fold cross-validation, and (iii) paired t-test with 5 x 2 cross-validation (5 iterations of 2-fold CV) and an adjusted test statistic $\tilde{t}$.

The paired t-test in (i) and (ii) uses the test statistic

$$t = \frac{\bar{p} \cdot \sqrt{n}}{\sqrt{\frac{\sum_{i=1}^{n}(p^{(i)} - \bar{p})^2}{n-1}}} \tag{1}$$

where $\bar{p} = \frac{1}{n}\sum_{i=1}^{n} p^{(i)}$. The adjusted test statistic in (iii) is

$$\tilde{t} = \frac{p_1^{(1)}}{\sqrt{\frac{1}{5}\sum_{i=1}^{5} s_i^2}} \tag{2}$$

where $p_1^{(1)}$ is the observed difference in performance from the first fold of the first replication and s_i^2 is the variance computed from the ith replication.

Of these three, Dietterich finds that only (iii), the 5 x 2 cross-validation approach using $\tilde{t}$, has an acceptable Type I error rate where the probability of Type I error is less than or equal to the significance threshold α. Dietterich recommends using either 5 x 2 CV with a paired t-test, or a McNemar's test when 10 iterations of model training are impractical. McNemar's test is a simple test based on a χ^2 test for goodness of fit; we do not discuss it in detail here but refer the reader to [17] for details. We note that this test is particularly useful because, in addition to only requiring a single iteration of training and testing for each model, it can be calculated using only aggregate model performance data (a simple confusion matrix).

Dietterich does, however, find that using 10-fold CV with a t-test has the highest power of any of the tests evaluated (ability to detect a true difference when it exists); together with its high Type I error rate, this might explain its frequent use in both the learning analytics community and the broader applied machine learning community. These properties, along with the unfortunate property that increasing the sample size can magnify the p-value of any observed difference using this method, translate into a high propensity to produce "statistically significant" results [54].

Further work has confirmed Dietterich's finding that paired t-tests with resampling schemes other than 5 x 2 CV lead to elevated Type I error rates [40, 6]. This work has also explored corrections to the t-test when used with resampled[2] model performance data. Nadeu and Bengio propose one such correction in [40]. Nadeu and Bengio find that resampling produces flawed estimates of the variance of model performance – the t-test underestimates the variance of the cross-validation estimator by failing to account for overlapping training sets – and propose a corrected t-test which accounts for this overlap. Specifically, Nadeau and Bengio propose an adjustment to the estimated variance of a traditional t-statistic, from

$$\hat{\sigma}^2 = \frac{1}{n_2} S_L^2 \tag{3}$$

[2]We take "resampling" to refer to any method which creates multiple experimental subpopulations $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$ from a single overarching dataset $\mathcal{D}$; resampling methods include bootstrap resampling and cross-validation.

to

$$\hat{\sigma}^2 = \frac{1}{J} + \frac{n_2}{n_1} S^2_{\mu_j} \tag{4}$$

where the number of samples $J = k \cdot r$, $S^2_{\mu_j}$ is the sample variance of the estimates, n_1 is the number of samples used for training, and n_2 is the number of samples used for testing. This modification is intended to account for the correlation between resamples, which the resampled t-test "boldly" assumes to be zero, by instead estimating that correlation as $\rho = \frac{n_2}{n_1+n_2}$ [40][3]. This modified test statistic is used with the Student's t-distribution and $n-1$ degrees of freedom, in otherwise exactly the same way as an unadjusted Student's t-statistic. This adjustment has been shown to have acceptable levels of Type I error and high replicability, particularly when using many ($R = 100$) runs [6]. Additionally, the correction avoids another critical flaw with the unadjusted resampled t-test: that the significance of the test statistic can be increased without bound simply by increasing k (the number of resamples), mentioned above. This procedure can be used with a variety of resampling approaches, including bootstrap resampling and cross-validation. However, we note that this procedure is rarely used in practice (and was not used in any of the works surveyed here) [57].

We note that this correlation between the training data in cross-validation folds was the original motivation for the 5x2 cross-validation approach proposed in [17]. Instead of accounting for correlation between overlapping folds, however, Dietterich's approach eliminates it by using only two folds, which results in nonoverlapping training sets within each run. Despite the acceptable Type I error rate, it has been widely recognized that Dietterich's 5x2 CV approach achieves low power [40, 17, 4]. Additionally, the test has been shown to have low replicability due to the large random variation inherent in using only 2 folds per run and the fact that the modified test statistic $\tilde{t}$ depends only on the difference $x1, 1$ and not on the full set of differences $x_{1,1}, ..., x_{r,k}$ [4, 6, 17]. Furthermore, despite the improved performance, the choice of $R = 5$ is somewhat arbitrary, and has been criticized as ad hoc and lacking in theoretical justification [40, 57]. We note, however, that 5x2 CV might be considered a useful replacement for 10-fold CV, the most commonly-used form of cross-validation, because 5x2 CV requires the same number of train-test iterations (10 in both cases) but does so with nonoverlapping folds within each iteration. In situations where many models are being considered or model training is computationally expensive, 5x2CV incurs no extra training burden (and actually might lead to *faster* training because each model is only trained on half of the dataset, not $\frac{9}{10}$ of the dataset as in 10-fold CV).

This same corrected test statistic can be used for cross-validation in a general approach we refer to as **corrected cross-validation** [57]. Cross-validation differs from bootstrap resampling in at least two ways that are relevant to sta-

[3] specifically see sections 3 and 4 for discussion and theoretical justification of this modification

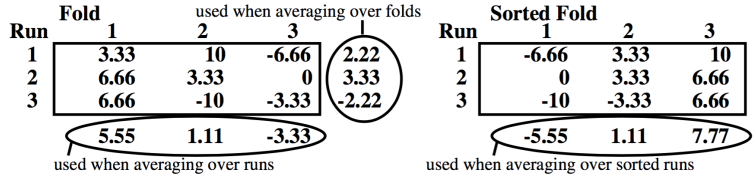

Figure 3: Example from [4] illustrating averaging over sorted runs (right) compared to averaging over folds or over runs (left) in the sorted runs sampling scheme, a seldom-used model evaluation technique with acceptable Type I Error, high power, and high replicability.

tistical testing. First, the test and training sets are guaranteed not to overlap within each fold using cross validation. Second, while training sets do overlap across folds (each training set shares $k-2$ folds of data with the others), they do so in a consistent way under cross-validation (in contrast, the overlap is random when using bootstrap resampling). Together, due at least in part to these two factors, it has been empirically demonstrated that a corrected cross-validation approach achieves superior performance relative to the corrected resampled t-test [5, 6]. Several proposals have offered slightly modified versions of the corrected cross-validation approach with different corrections to t, paired with specific values for R and k.

One such proposal is 10x10 CV with a variance correction, known as the **corrected repeated k-fold cv test**. This test utilizes a variance correction similar to the corrected resampled t-test, but pairs the modified test statistic with a cross-validation approach. Together, these modifications achieve an acceptable Type I error rate (equal to α, as expected) and better statistical power than 5x2 CV [6]. 10x10 CV with the corrected repeated k-fold CV test has been shown to have higher replicability than corrected resampling with $R = 100$ and 10x10 CV (both of which require 100 iterations of model training and thus the same computational effort) in a direct experimental comparison by [6].

A novel and seldom used sampling-based approach to the model comparison task, evaluated in [4], shows acceptable Type I Error, high power, and high replicability: the **sorted runs sampling scheme**. This is a modified approach to cross-validation, in which the results $P_{R,1}, ..., P_{R,k}$ for each run are first sorted, and then the averages of the ordered folds are taken according to their respective rank across runs. This results in a sample of k estimates for each statistical model, where $\overline{P_1} = \frac{1}{R}\sum_{i=1}^{R} min(P_1...k)$ is the average of the lowest scores for a given learner in each of the R runs, P_2 is the average of the second-lowest score for a learner in each run, etc. An illustration of this scheme is shown in Figure 3.

The sorted runs sampling scheme has been shown to achieve a better Type I Error rate than 10x10cv, but with a slightly lower power [4]. The sorted runs sampling scheme also has the appealing property that it "results in a sample for which the independence assumption is not heavily violated, so that

Test	Description	Sampling Method	Test Statistic	Degrees of Freedom
(1) 5x2cv paired t-test [17]	Adjusted t-test calibrated for use with 5x2cv. Lower power than (2,3,4).	2-fold cross-validation	$t = \frac{x_{11}}{\sqrt{\frac{1}{5}\sum_{j=1}^{5}\hat{\sigma}_j^2}}$	5
(2) Corrected resampled t-test [40]	Standard t-test with adjusted variance. Avoids significance "inflation" for increasing number of resamples. Lower reproducibility than (3).	Random Subsampling	$t = \frac{\frac{1}{n}\sum_{j=1}^{n} x_j}{\sqrt{(\frac{1}{n}+\frac{n_2}{n_1})\hat{\sigma}^2}}$	$n-1$
(3) Corrected repeated k-fold cv test [6]	$r \cdot k$-fold CV, with the same variance correction as the corrected resampled t-test above.	Cross-validation	$t = \frac{\frac{1}{k\cdot r}\sum_{i=1}^{k}\sum_{j=1}^{r} x_{ij}}{\sqrt{(\frac{1}{k\cdot r}+\frac{n_2}{n_1})\hat{\sigma}^2}}$	$k \cdot r - 1$
(4) Paired t-test with sorted runs sampling scheme [4]	Adjusted sampling scheme with standard t-test. Note that x_i is summed across sorted runs.	Sorted runs sampling scheme	$t = \frac{\frac{1}{n}\sum_{i=1}^{n} x_i}{\sqrt{\hat{\sigma}^2}/\sqrt{df+1}}$	$r-1$

Table 2: Overview of statistical tests for the two-model model comparison task with acceptable error rates. Each uses a Student's t-distribution with the specified degrees of freedom.

no correction in variance or degrees of freedom is required." This allows the sorted runs sampling scheme to be evaluated with an uncorrected t-statistic and $df = n - 1$.

We conclude this section by noting that the tests considered in this section are limited in that they are only statistically appropriate for use with comparing two models on a single dataset or domain. They are not, for example, robust to multiple comparisons, or suited for use across datasets, both of which can also lead to increased probability of error, as we will discuss below. However, these tests give a clear sense of the issues in statistical model evaluation from the frequentist perspective. These include, most importantly, (a) achieving accurate estimates of model performance and its variance; and (b) finding statistical tests that are calibrated to produce the expected error distributions (i.e., Type I error rate equal to α) when applied to model performance data.

3.2.2 Multiple Models, Multiple Datasets

In practice, comparisons of only two models on a single dataset are rare. The space of potential models in almost any experiment is so large that it would be far too restrictive to only compare two; furthermore, in many cases, statistical models are tested across several datasets (i.e., several different MOOCs). In this section, we extend the discussion of Section 3.2.1 to evaluate NHST methods for the evaluation of multiple models across multiple datasets, which is far more common in the field of learning analytics (for example, when comparing predictive models across several MOOCs).[4] These procedures are slightly more complex, but reflect the core issues of (a) estimating and (b) accurately testing model performance which drove the discussion in Section 3.2.1.

As Section 3.2.1 makes clear, comparing the results of predictive models is not analogous to comparing the results of normal random population samples. The assumptions of traditional statistical tests used for data analysis, such as the paired t-test, are often strongly violated by predictive model performance data (i.e., non-normality; independence; noncommensurability across datasets/domains). This also holds in the case of multiple-model-multiple-dataset comparison. The classical statistical procedure to determine whether there was a difference between several experimental subsamples – in this case, the results of several predictive models across many datasets – would be analysis of variance (ANOVA) to determine whether a difference existed [20, 15], perhaps with a post-hoc test if groupwise differences were detected. This procedure is unfit for predictive model evaluation, however, for at least two reasons. First, ANOVA assumes a normal distribution of the data – an assumption which is not guaranteed in the case of machine learned models. The degree to which this assumption is problematic depends on the task, datasets, and evaluation metrics being used; however, it is far from guaranteed in a model evaluation context and is difficult to know *a priori* when selecting a model evaluation procedure. Second, repeated-measures ANOVA assumes sphericity, the condition where the variances of the differences between all possible pairs of within-subject conditions (i.e., between the performance of different learning algorithms) are equal. This assumption is likely to be violated when comparing predictive models; there are wide differences between the variability of even commonly-used learning algorithms such as logistic regression and classification trees (due to the well-known bias-variance tradeoff in predictive modeling). Third, ANOVA is unable to account for non-independence of overlapping samples due to repeated resampling (this is equivalent to evaluating multiple non-independent samples per cell) [15]; avoiding this between-folds correlation this was the motivation for the corrected cross-validation approach discussed above. We do not discuss the fitness of ANOVA for model evaluation further because it was not used in any of the work surveyed; however, we refer the reader to [15, 29] for further discussion.

[4] We do not specifically discuss the task of comparing two classifiers over multiple datasets, which is a subset of this task; however, we note that a Wilcoxon signed ranks test is recommended for this task in [15].

Nonparametric procedures are appropriate when the strong assumptions of parametric tests such as ANOVA are likely to be violated. In this section, we focus on describing a *recommended* approach, a two-stage procedure utilizing the Fisher test and Nemenyi post-hoc test; because other multiple-hypothesis testing techniques are not widely used in predictive modeling (and were not used in any of the work surveyed here), we do not discuss or critique invalid techniques for this task.

The Friedman test is a non-parametric version of the ANOVA test, and compares the average rankings of the k algorithms across each of N datasets, calculating a test statistic measuring the probability of the observed rankings under the null hypothesis of all algorithms having equivalent performance (and therefore equal expected average rankings). The observed value of the Friedman statistic

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right] \tag{5}$$

where R_i^j is the rank of the jth of k algorithms on N datasets and the statistic is distributed according to a chi-square distribution with $k-1$ degrees of freedom, is compared to a critical value for the given values of N and k [22].

If the null hypothesis is rejected at the selected significance level ($\alpha = 0.05$ in this experiment), the post-hoc Nemenyi test is used to compare all classifiers to each other. The Nemenyi is similar to a nonparametric version of the Tukey test for ANOVA, and uses a critical difference

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{6}$$

as a threshold to determine whether the performance between any two classifiers is significantly different, where the critical value q_α is based on the Studentized range statistic divided by $\sqrt{2}$.

The results of the Friedman and Nemenyi tests are often visualized using a Critical Difference (CD) diagram. An example is shown in 4. Models are plotted on a number line according to their average rank across all datasets, and bold CD lines are used to link models which are statistically indistinguishable at α. As we will note below, this procedure does not scale well with large k, as plotting many labeled observations on a number line can be difficult to interpret. We introduce a novel plot adapted frmo another work to interpret the results of both Bayesian and frequentist procedures.

One advantage of this method is that because the Friedman test uses only the *rankings* of the algorithms on each dataset, it does not require estimates of the variance of model performance – recall that inaccurate estimates of variance were one of the primary confounding issues with the tests described in Section 3.2.1. Instead, it only requires that the estimates of model performance and the measured rankings they produce are reliable and "...that enough experiments were done on each data set and, preferably, that all the algorithms were

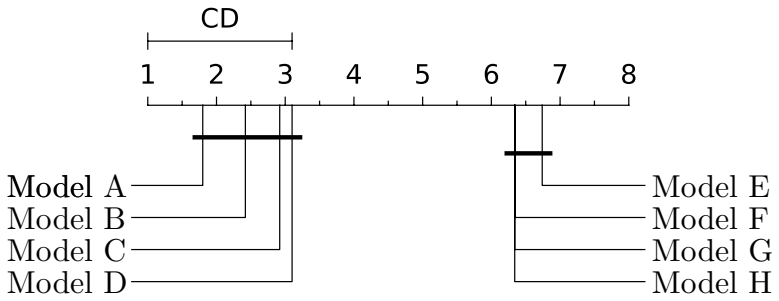

Figure 4: A Critical Difference (CD) diagram, which is used to visualize the results of the Friedman + Nemenyi testing procedure. Models are plotted according to their average ranks, with bold CD lines connecting models which are statistically indistinguishable at α.

evaluated using the same random samples..." [15] and that the datasets, and therefore the rankings of the algorithms across each dataset, are independent. In contrast to many other statistical approaches to comparing model performance, such as ANOVA, the Friedman test makes no further assumptions about the sampling scheme.

Additionally, the Friedman and Nemenyi test accounts for multiple comparisons – the number of models compared, k, is accounted for in both the Friedman statistic (Equation 5) and the post-hoc Nemenyi test (Equation 6). This accounts for the many comparisons often conducted in the course of a model evaluation experiment, which can grow quite large even with modest numbers of feature sets, algorithms, and hyperparameter settings (for example, in the case study in Section 4, $k = 96$). However, this procedure is only able to account for k if the total number of candidate models is tracked, reported, and accounted for in the calculation of the test statistic. In practice, we expect that the reported values of k in many published works are lower than the true number of models considered, because either (a) the large number of intermediate model comparisons conducted "under the hood" by auto-tuning toolkits in popular modeling packages, including caret in R and WEKA or autoWEKA, are not tracked or even made available to researchers, or (b) the full scope of these model comparisons is not included in the final statistical tests reported in many works, due to their tendency to widen the critical difference (and diminish potentially "statistically significant" differences) or because many models considered along the "garden of forking paths" [26] are simply discarded without regard to their implications for statistical testing.

The tracking and reporting of multiple comparisons might seem like statistical finger-wagging, but we believe that dismissing these concerns is a dire mistake, particularly for MOOC researchers, for several reasons.

First, in a nascent field where replication is difficult and rare, having accurate estimates of the confidence of a given finding is important. Any given

predictive experiment may be the only study to evaluate its method or dataset, so determining the generalizability or confidence of those findings is entirely reliant on the use of proper evaluation techniques on part of the author.

Second, while concerns about multiple comparisons might seem unimportant when the number of comparisons N is small, in most machine learning experiments N is quite large – with hundreds or thousands of candidate models being not uncommon in the MOOC modeling research surveyed (i.e., [56] compares at least 1400 models with various architectures and window formulations; [48] evaluates over 10,000 candidate models; neither correct or even acknowledge the large number of multiple comparisons in the context of their significance testing). John Tukey describes the problem in [50]:

> A man or woman who sits and deals out a deck of cards repeatedly will eventually get a very unusual set of hands. A report of unusualness would be taken quite differently if we knew it was the only deal ever made, or one of a thousand deals, or one of a million deals, etc.

We need to know the number of "deals" in our evaluation of machine learning experiments in order to know how to interpret the results. Experiments where only a few "hands" are dealt are rare in the era of computational science and in the specific body of work surveyed here, and we expect the number of candidate models in each experiment to grow as advances in statistical learning bring new modeling techniques, and advances in computational software make these models easily available to researchers.

While the Friedman and Nemenyi procedure is theoretically effective (and certainly preferable to the naïve average approach), we note two concerns about the method in practice. The first, raised above, is that this procedure requires tracking and reporting all comparisons conducted. We believe this practice is rare in machine learning research, and perhaps it is even unreasonable for researchers to collect and save this data in the course of an experiment. Unfortunately, few tools even exist to track these types of comparisons in the course of model exploration (a notable and promising exception is the modelDB project [51]).

Second, there are larger concerns about the use of null hypothesis significance testing (NHST) for predictive model evaluation, suggesting that such tests may be ineffective even under ideal conditions:

1. **NHST misinterpretation.** The results of NHST are often misinterpreted. This is true in the broader scientific literature (for example, see an early discussion in [11]) as well as in the MOOC literature. It is not uncommon to see the results of significance testing interpreted as the probability that one algorithm is better than another [16] , the probability of successful replication [11], or proof that an algorithm is "significantly better" (e.g. [47, 39, 34, 37, 43]; other examples with similar language abound in the subset of work which utilizes NHST). The p-value generated by a NHST is none of these. In fact, the result of a hypothesis test estimate

the probability of the experimental results D if the hypothesis H_0 is correct, $P(D|H_0)$, which is **not** the same as the probability of correctness of hypothesis given the experimental results, $P(H_0|D)$ [11, 16]. A NHST produces a *conditional* probability estimate; in the case of a comparison between two (or more) prediction algorithms, this is conditional on the supposed equivalent performance of these algorithms.

2. **False equivalence testing.** Here we consider specifically the condition of this conclusion, H_0: that there is exact equivalence between the performance of two models, or a difference of zero. This condition is almost always false [1]; Jensen (Jensen1997-cn, p.1000) refers to such hypotheses as "the *nil* hypothesis" after the probability that this hypothesis is true: *nil* (see also [50]). Particularly in the case of machine learning models, it is unlikely that a null hypothesis of exact *equivalence* between any two algorithms is genuinely true. Thus, these conditional estimates, even when properly interpreted, are loosely indicative of any potential true state of the world, at best, and potentially even uninformative or misleading.

3. **Increasing data leads to significance.** As a direct consequence of (ii), detecting a "significant" difference under the null hypothesis typically only requires collecting enough data: if the hypothesized equivalence of H_0 is actually false (as it almost always is, particularly in the case of machine learning algorithms), we merely need to collect enough data to "reval" it [16, 17, 11, 57, 38]. As discussed in Section 3.2.1 above, many hypothesis testing methods, including the paired t-test commonly used in the MOOC research surveyed, decrease the estimated variance – effectively increasing the significance of a given result – as the sample size increases. Most NHST methods fail to account for this; even if they do, the reported p-value itself does not allow the reader differentiate between the effect of the sample size and the effect size [54, 1].

4. **Arbitrary α.** There is no statistical theory to guide the selection of α, which is used as the accept-reject cutoff in hypothesis testing, and its determination is based on a mix of convention and convenience (choosing the lowest round threshold above the observed significance levels) [38]. Without using formal preregistration, there is no way to bind researchers to a fixed significance level prior to observing their experimental results. The use of confidence intervals, recommended by Cohen in [11] – which are typically obtained by simply inverting the rejection regions for test statistics such as – might reduce the black-and-white thinking produced by reporting only significance/non-significance relative to an α. However, there is also no principled way to choose the size of a confidence interval, and the same objections apply. Jensen highlights Birnbaum's confidence curves [2] as a potential solution to this problem. Confidence curves are continuous plots of confidence region bounds over the bounds of the confidence region $CR \in [0.5, 1]$. We agree that the more widespread adoption of confidence curves would be a beneficial practice.

At the beginning of this section, we noted that effective procedures for evaluating multiple models across multiple datasets were needed in the field of learning analytics. We described a theoretically effective NHST-based method, the Friedman test paired with a post-hoc Nemenyi test, to address this case. However, in light of the concerns highlighted above – which are not only applicable to this procedure, but to *any* null hypothesis significance test – we believe that other methods are worth exploring for predictive model evaluation in learning analytics.

3.3 Model Evaluation: Bayesian Testing

In this section, we discuss Bayesian methods for predictive model evaluation. Similar to our discussion in Section 3.2.2, we do not attempt an exhaustive survey – there is little need to discuss ineffective methods, because we are aware of no prior learning analytics research which utilizes such methods. We introduce a Bayesian method for model evaluation, briefly discussing its theoretical underpinnings and how it avoids many of the concerns about NHST mentioned above. After introducing this method, we discuss why we believe it is the most useful method for model evaluation in learning analytics in 3.4

3.3.1 Bayesian Parameter Estimation

The application of Bayesian statistical methods to model evaluation is relatively new, and have risen in prominence particularly over the past decade or so as scientific consensus around the concerns outlined above has grown, and as Bayesian modeling techniques in general have become more widely used and accessible to researchers. There are several approaches for model evaluation which use Bayesian techniques, and we refer the reader to the excellent overview provided by Benavoli et al. in [1] for a detailed review[5]. In this work, we focus on a family of approaches which Benavoli et al. refer to as *Bayesian parameter estimation* or simply *Bayesian analysis*. We will refer to this approach as *Bayesian model evaluation* to highlight the specific application of interest in the current work, but occasionally use Benavoli et al.'s nomenclature when the meaning is clear.

In this work, we specifically consider the Bayesian hierarchical correlated t-test of Coriani et al. [12]. This is a test used to compare the results of two classifiers over multiple datasets, and it can be extended to comparisons of multiple classifiers over multiple datasets without adjustment.

The Bayesian hierarchical correlated t-test uses a hierarchical model, a Bayesian technique for modeling distributions over a range of potential parameters, to account for the mean, variance, and correlation of the results of cross-validated model performance data across multiple datasets. Specifically, the Bayesian hierarchical correlated t-test uses the following model:

[5]For interested readers, we note that Benavoli et al. highlight specific Bayesian methods for the task, discussed in section 3.2.1, of analyzing cross-validation results which accounts for the correlation due to the overlapping training sets [12]; we note that this method makes use of the correction proposed by Nadeau and Bengio in [40].

$$\mu_{\cdot}...\mu_k \sim t(\mu_0, \sigma_0, \nu), \tag{7}$$

$$\sigma_1...\sigma_k \sim unif(0, \hat{\sigma}), \tag{8}$$

$$x_i \sim MVN(\mathbb{1}\mu_i, \Sigma_i), \tag{9}$$

Equation (9), where x_i is the vector of differences between models, captures the correlation between cross-validation measures on the ith dataset by model these as draws from a multivariate normal distribution with mean μ_i, and correlation ρ. $\mathbb{1}$ is a vector of ones, and the covariance matrix Σ has diagonal elements σ_i^2 and off-diagonal elements $\rho\sigma_i^2$ where $\rho = \frac{n_te}{n_tr}$, again following Nadeau and Bengio [12, 40]. Equation (8) allows each dataset to have its own estimation uncertainty, standard deviation σ_i, drawn from a common uniform distribution where $\hat{\sigma} = 1000 \cdot \sum_i^q \frac{\hat{\sigma}}{q}$. We refer the reader to [1, 12] for further details.

Equation (7) models the differences between two classifiers on each dataset, and thus accurately models the real-world multiple-dataset task where some classifiers might perform better or worse on certain datasets, leading to variability in the difference μ_i on each dataset i. Equation (7) also models the fact each observed mean difference on a single dataset, μ_i, depends on the average difference of accuracy between the two classifiers on the population of data sets, μ_0. This parameter, μ_0, is typically the quantity of interest, and is modeled with a Student distribution with variance σ_0^2 and degrees of freedom ν. The use of a Student distribution here makes the model more robust to outliers [35], and slightly more conservative than its frequentist counterpart [12].

Equation (8) allows each dataset to have its own standard deviation σ_i, drawn from a common uniform distribution, where $\hat{\sigma} = 1000 \cdot \sum_i^q \frac{\hat{\sigma}}{q}$. We refer the reader to [1, 12] for further details.

Distributions for the hyperparameters are given by:

$$\sigma_0 \sim Uniform(0, \bar{s_0}), \tag{10}$$

$$\mu_0 \sim Uniform(-1, 1), \tag{11}$$

$$\nu \sim Gamma(\alpha, \beta) \tag{12}$$

$$\alpha \sim Uniform(\underline{\alpha} = 0.5, \bar{\alpha} = 5) \tag{13}$$

$$\beta \sim Uniform(\underline{\beta} = 0.05, \bar{\beta} = 0.15) \tag{14}$$

Equations (10) and (11) represent the hyperparameters for the standard deviation and difference, where the prior for the standard deviation σ_0, is unifnromly distributed on $[0, \bar{s_0}]$ with $\hat{s_0} = 1000 s_{\bar{x}}$, and the prior for the difference μ_0 is uniformly distributed on $[-1, 1]$. The $[-1, 1]$ range for μ_0 is appropriate for

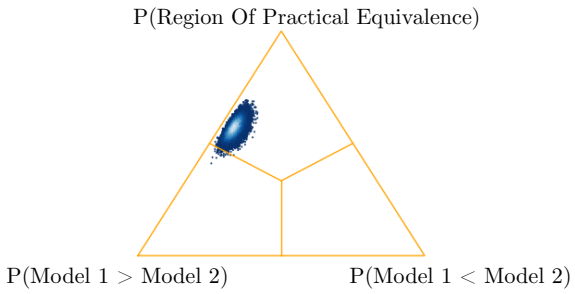

Figure 5: A Bayesian posterior plot. The plot visualizes the results of Markov-Chain Monte Carlo (MCMC) sampling applied to the comparison of two predictive models across multiple datasets. The estimated probability of each outcome is the proportion of samples that fall in each section of the plot, producing estimates for each of the three outcomes and plotting them in barycentric coordinates. Inspection of these plots can be useful for small comparisons, but for experiments with large models spaces, inspecting all $\frac{k(k-1)}{2}$ pairwise plots is impractical.

most measures of predictive model performance, including AUC, the measure used in the experiment below.

Equation (12) gives the prior degrees of freedom for the t-distribution used to model the mean differences $\mu_1 ... \mu_k$, and (13) and (14) give the distribution of its hyperparameters. These are chosen to allow the Gamma distribution for ν to accommodate a wide range of different prior beliefs [12].

For each pair of candidate models, this Bayesian hierarchical model is used to make inferences about the observed and overall average differences in model performance μ_i and μ_0 using Markov-Chain Monte Carlo (MCMC) sampling. From the dataset-level model performance data, MCMC is used to generate samples of $(\theta_l, \theta_e, \theta_r)$, where θ_i represents the posterior probability that model A is better, the models are equivalent, and model B is better, respectively. These samples represent the hypothetical differences in performance on a future unseen dataset [12]; generating $N = 50,000$ samples on a typical laptop computer takes only a few seconds, and conducting this comparison for all $\frac{96*95}{2} = 4560$ pairwise comparisons in the experiment below takes less than 10 minutes using the `BayesianTestsML` Python library[6].

These samples are then used to compute the posterior probability of each of these hypotheses by simply counting the proportion of MCMC samples for which θ_i has the highest posterior probability. Additionally, the results of this sampling can be visualized by projecting the $(\theta_l, \theta_e, \theta_r)$ triplets onto barycentric coordinates to produce a posterior plot, shown in Figure 5.

This method is able to account for the different uncertainty which characterizes each dataset by estimating unique parameters for each dataset, and, because

[6]https://github.com/BayesianTestsML

the heirarchical model applies shrinkage to the μ_i values when estimating them jointly, it estimates them more accurately than previous approaches which use maximum likelihood estimation [12].

The result of Bayesian analysis is different from a NHST procedure. In Bayesian analysis, the goal is to estimate the *posterior probability* of the null hypothesis by applying Bayes' rule to models of the likelihood of observing the data under each possible set of parameters, along with a model of the prior probabilities of these parameters. This allows a Bayesian procedure to correctly lead to inferences such as: "based on the observed data, there is a 95% probability that Model A is more accurate than Model B". In contrast, NHST only allows such statements conditional on the null hypotheses: "there is a 5% probability of observing the data, if the null hypothesis is true" (as we noted above, it is common for experimenters to misinterpret or misstate the results of NHST in the form of a Bayesian posterior). Additionally, NHST can never "prove" the null hypothesis, H_0, that two models have equivalent performance, despite the fact that we are often interested in this possibility. This is an important difference between the two procedures: they justify different inferences; only a Bayesian procedure allows us to make inferences about the posterior probability of differences in model performance, and only the Bayesian procedure allows us to draw substantive conclusions about models having similar or equivalent performance.

3.4 Recommended Approach: Bayesian Model Evaluation

In this section, we compare the Bayesian approach described above to the NHST procedure described in section 3.2.2. We present a series of criteria for an ideal procedure for predictive model evaluation in learning analytics, and demonstrate that only the Bayesian procedure meets these criteria.

Evaluating predictive models in learning analytics is a complex task. This typically involves considering (a) a massive space of potential models, because there is a lack of consensus about the most effective models for even the most common prediction tasks, such as dropout; (b) relatively small collections of datasets, for example, even the largest prior MOOC studies of which we are aware evaluate around 40 MOOCs (i.e., [56, 19]) and (c) large individual datasets, which make repeated model-fitting undesirable, if not intractible.

Additionally, more general features of statistical testing procedures also place constraints on the model comparison task. This includes the fact that (d) many model evaluation procedures entail statistical assumptions about the data which need to be reasonably met in expectation; (e) statistical testing often requires adjustment for multiple comparisons; and (f) different procedures support inferences of a different kind (i.e., inferences about conditional probabilities from NHST versus the posterior inferences from Bayesian analysis).

These considerations collectively point toward a series of criteria which we believe would describe the ideal model evaluation scheme:

1. **Computationally tractible**: Statistical testing should require as little

computational overhead as possible. Ideally, any model evaluation procedure would use data collected during model training and testing, and would not require additional data collection. It should scale well with the number of datasets N, the number of models k, and the dimensionality of each individual dataset.

2. **Calibrated for use with predictive models**: As discussed in section 3.2, not all forms of statistical testing are appropriate for the unique task of predictive model evaluation. This includes, for example, accounting for overlapping cross-validation training sets (when cross-validation is used) and accounting for noncommensurability across datasets.

3. **Impose minimal assumptions on the data**: An ideal model evaluation method would make few, if any, assumptions about the underlying data and about the model performance data collected from it, such as normality, symmetry, commensurability across datasets, constant variance, etc .

4. **Robust to multiple comparisons**: Identifying high-performing models involves searches of large candidate model spaces with few, if any, *a priori* assumptions about which might perform best on a given task. Any procedure should allow making many comparisons – particularly when used for pairwise comparisons, where the number of possible comparisons is $\frac{k(k-1)}{2}$.

5. **Test an informative** H_0: We would like our model evaluation procedure to test realistic and informative null hypotheses. Even concrete, confident inferences are useless if they are drawn about an H_0 which is extremely unlikely.

6. **Provides direct evidence about** H_0: When conducting model evaluation, we are interested in directly drawing inferences about H_0, the null hypothesis that two models have equivalent performance. Inferences that are conditional on H_0 being true are not considered informative in the absence of information about the probability of H_0 (and we never have this information).

7. **Account for magnitude and uncertainty**: Model evaluation should clearly account for the magnitude of observed differences and the uncertainty of its estimates. These effects should be separable in the subsequent analysis of model performance testing.

Item 1 does not particularly count in favor of any of the schemes considered, but also does not rule any out. All are computationally tractible with consumer-level computing hardware. The MCMC sampling method used by Bayesian methods incurs a higher computational overhead than any of the NHST procedures, but particularly for the relatively small numbers of datasets considered in most learning analytics reserach (dozens of courses), the difference is negligible.

For equally effective schemes, item 1 suggests that we can choose the one which can provide the most efficient estimates.

Item 2 above excludes the statistical testing procedures, such as the paired t-test, described in section 3.2.1, which are unable to accurately account for the expected variance of model performance estimates (particularly when used with cross-validation), or which allow for monotonically increasing the significance of a comparison simply by increasing the sample size. We note that this test was the most common NHST procedure used in the work surveyed (shown in Table 1).

Item 3 further excludes approaches, such as ANOVA, which impose strict assumptions on model performance data (such as normality). However, this does not provide a clear reason to prefer nonparametric NHST procedures over Bayesian analysis procedures.

Items 4, 6, 5, and 7 all count against any NHST approach, and favor a Bayesian model evaluation procedure. In particular, 4 is not fully addressed by any NHST procedure. We noted that the explosion of machine learning techniques has created a massive space of potential models to consider, and we would prefer a model evaluation technique which does not hinder exploration of an arbitrarily large space of these models. While NHST procedures can be adjusted for multiple comparisons, these adjustments are approximate at best and often impractically conservative with large k and small to moderate N, as our case study demonstrates. In contrast, the Bayesian approach does not "accept" hypotheses and is generally unconcerned with Type I errors as it only estimates posterior probabilities. A Bayesian hierarchical model can directly account for the uncertainty from multiple comparisons [25]. Further, we discussed 5 in detail in section 3.2.2; we simply note here that NHST procedures for model evaluation utilize null hypotheses of equivalence which are, to put it plainly, almost always false in the case of machine learned models [16, 1]. There is nothing to be learned by rejecting such a hypothesis. What we *would* like to do is draw inferences directly about an informative H_0, as stated in 5. We discussed in section 3.3.1 that only a Bayesian procedure allows for direct inferences about posterior probabilities; NHST strictly permits inferences conditional on the truth of H_0. Finally, 7 also supports only a Bayesian technique over any NHST model evaluation procedure. By using a region of practical equivalence (ROPE), the Bayesian analysis technique we adopt from [12, 1] allows us to account for the magnitude of a difference in performance, and to conduct analyses where small differences in performance – even if "statistically significant" – are not considered practically significant. Additionally, we would like a measure of the uncertainty of our conclusions. A p-value which reflects a mix of effect size and uncertainty [38] is far less informative than a direct posterior estimate of the probability of a hypothesis.

Together, these criteria collectively mount a strong imperative in favor of Bayesian model evaluation. We believe that a demonstration of the sample averaging, NHST, and Bayesian techniques side-by-side can make an even stronger case for its adoption in the learning analytics community. We provide such a comparison in the next section with the aim of convincing readers that the

Bayesian method is preferable and providing a demonstration of its implementation and interpretation in the context of predictive models of student success.

4 Case Study: Evaluating MOOC Dropout Models

In this section, we present the results of an experiment which constructs and compares several dropout models across a large sample of MOOCs. Our goal in this section is twofold. First, we wish to demonstrate, side-by-side and on the same sample data and experimental results, the three broad approaches to evaluating predictive models (naïve average; frequentist null hypothesis significance testing; Bayesian model evaluation). Such a comparison stands to illuminate not only procedural and inferential differences between these approaches in general, but it also demonstrates very practically how those differences can lead researchers to different conclusions, even with the same underlying data.

Second, we hope to demonstrate interesting and useful results in this experiment: the data, feature extraction, and statistical modeling methods used here are common in applied predictive modeling research including the fields of learning analytics and educational data mining; these results can provide important evidence of the effectiveness of the data sources, feature extraction methods, and modeling algorithms considered. We believe the current experiment may be particularly illuminating because prior research typically evaluates the many parameters of a modeling experiment separately (data source/feature extraction from data; modeling algorithm; hyperparameter tuning). Here, we instead consider these procedures together, to fully evaluate potential synergy between these different approaches and to determine whether they are effective in modeling with the entire population of a course.

4.1 Data

The data used in this experiment are a large and diverse sample of MOOCs offered by the University of Michigan on Coursera. In total, the experiment evaluated $N = 48$ sessions of courses. These courses were on a broad variety of topics in several domains, including technology (Inside the Internet; Social Network Analysis), science (Introduction to Thermodynamics; Sleep: Neurobiology, Medicine, and Society; Introduction to Cataract Surgery), finance (Principles of Valuation: Time Value of Money; Valuation: Alternative Methods), and others (Digital Democracy; AIDS: Fear and Hope; Model Thinking). A summary of the data for the courses used in shown in Table 3.

This large, diverse MOOC dataset makes this one of the largest MOOC studies to date in terms of number of courses evaluated. The largest-scale work on constructing predictive models using raw MOOC data of which the authors are aware to date are [**?**, 36], which each build predictive models on 39 XuetangX MOOCs; [56], which builds predictive models on 40 HarvardX MOOCs; and [19], which performs explanatory modeling on a set of 44 MOOCs. In particular, the

Metric	Value
Number of Sessions	48
Number of Unique Courses	17
Total Number of Active Students	117,028
Total Number of Interactions	2,479,900
Average Number of Active Students (SD)	2490 (2391)
Average Length (SD) in weeks	9.8 (2.6)
Average Number of Unique Forum Posters (SD)	507 (447)
Average Number of Course Assignments (SD)	1.2 (0.9)
Average Number of Quizzes (SD)	18 (16.8)
Average Number of Human-Graded Quizzes	1.6 (1.5)

Table 3: Summary statistics for courses used in case study.

size of this dataset is relevant because it represents a reasonable upper bound for the number of datasets, N, that a MOOC predictive modeling experiment might utilize (which serves as a limiting factor in the ability of frequentist methods to identify differences between models).

4.2 Experimental Setup

We were interested in using the three broad methods outlined above – naïve average, NHST, and Bayesian – for evaluating a set of predictive models for predicting dropout early in a MOOC. There are many components of predictive model-building, including feature extraction from raw data, algorithm selection, and hyperparameter tuning, which are often tested individually but rarely evaluated together. In this experiment, we seek to collectively evaluate feature sets, algorithms, and hyperparameters *together*. We do so for several reasons: (a) this more realistically demonstrate the large numbers of comparisons that such a procedure entails, because researchers are likely to conduct at least some of these comparisons in an experiment outside of the final or published experiment. This full number of comparisons is essential, for example, in the multiple comparisons corrections made by many NHST procedures; we sought to demonstrate what happened in such procedures when they were used at scale. (b) this procedure also stands to be more informative than indivudlaly experimenting with only one model element (i.e., the algorithm), while constraining the choice of the others (i.e., only using discussion forum-based features). By considering the collective space of all of our feature-algorithm-hyperparameter combinations, we believe that this experiment stands to identify more effective approaches for dropout modeling.

We consider the following elements for each aspect of our predictive models:

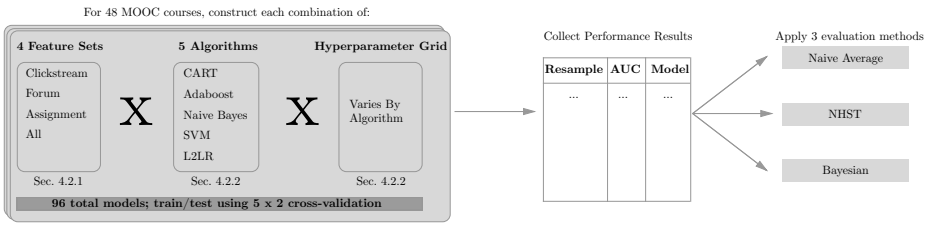

Figure 6: Experiment overview. Using a large, highly diverse sample of MOOCs, we test 96 unique feature-algorithm-hyperparameter combinations using 5x2 cross-validation. Each model is trained on the same random folds of data. The performance of each model on each fold (or *resample*) is collected, yielding more than 80,000 observations, each representing one iteration of model training/testing on a specific resample. These results are compared using three different methods: the "naïve average" approach often used in published data mining research; a NHST-based method using the Friedman and post-hoc Nemenyi tests; and a Bayesian method.

4.2.1 Feature Extraction/Data Source

MOOCs are particularly notable for the rich and varying sources of data they generate, and MOOC platforms typically offer several raw data sources (Coursera, for example, offered seven different data exports, covering interactions to forum posts, assignments, and demographics, on their Spark platform which is used in the current experiment [13]). We were thus interested in exploring which of these data sources might be most useful for student success prediction, particularly considering prior work that has suggested that forum posts [14] and assignments [52] might be highly effective dropout predictors. We thus consider sets of features extracted from the following data sources, with the specific features based on prior work:

- **Clickstream**: We use a relatively simple set of counting-based features, which count the number of accesses to various course pages, number of forum views, and number of video views. These features are common in activity-based dropout modeling, and are most similar to those from [58]. This is the simplest and smallest feature set of the three evaluated.
- **Forum Posts**: We use a diverse set of features which contain rich natural language, sentiment, and text complexity metrics, gathered from several previous works which demonstrated that these were effective predictors of dropout or student performance more generally [44, 14, 55].
- **Assignments**: We use a set of features derived from students' quizzes, peer-graded assignments, in-video quizzes, and exams, computing both simple features (such as average grade, average raw points, and number of submissions each week) as well as more complex features (which track

changes in each quiz type over time, or their number of submissions relative to the highest number of submissions by any student that week). Again, these features have been demonstrated to be effective dropout predictors in previous work [52]. Where courses used no assignments, models using this method defaulted to majority-class prediction.

- **All:** This feature set is the union of the three features sets above.

All features and their definitions are shown in Table 4. In addition to referencing distinct "types" of features (activity, forum/language, performance/learning), these different feature sets are drawn from distinct data sources (clickstream, forum database, assignments database, respectively), which allows this experiment to potentially inform future development efforts by identifying which data source might contain the most useful student performance effort and therefore be a reasonable use of scarce developer time to extract feature sets.

4.2.2 Algorithms and Hyperparameters:

Predictive model selection and tuning is the context in which the model evaluation problem is most often discussed. In the current experiment, we sought to mimic the choices learning analytics researchers often face in a predictive modeling experiment by considering a space of candidate models which plausibly matched the space of models considered in a dropout modeling experiment. This approach allows this case study to (i) genuinely demonstrate the inferential effects of testing this large model space in conjunction with multiple feature sets in a realistic environment, (ii) demonstrate the performance of both NHST and Bayesian methods in the face of large k and moderate N, and (iii) to demonstrate empirical results which are more generally informative about effective methods for dropout modeling in MOOCs beyond a mere methodological demonstration.

We consider the following models, tuning the listed hyperparameter settings for each:

- **Decision Tree**: Decision trees are one of the most popular algorithms for predictive modeling in MOOCs, both due to demonstrated performance on a variety of prediction tasks and their interpretability. We fit classical decision trees based on the original implementation of Breiman et al. [7], tuning the cost-complexity parameter which regularizes the tree depth based on the purity of the splits achieved.

- **Regularized Logistic Regression**: Logistic regression is second only to decision trees in its prevalence in prior MOOC research, and is effective, interpretable, and highly efficient to fit to most datasets. We use an L2 (ridge) regularized logistic regression, which penalizes the overall size of model coefficients and shrinks them toward zero using an L2 penalty term, λ, which is the only hyperparameter tuned in this model. L2 logistic regression is also more effective when collinearity (correlation among predictors) is present, which is common in user-level MOOC data.

- **Gradient Boosted Tree (Adaboost)**: Gradient boosted trees are a powerful method that constructs ensembles of iteratively reweighted "weak learners" to the data, learning a set of weights based on the observations misclassified at each iteration. The adaboost method has been shown to achieve high accuracy in many machine learning tasks with little tuning; Leo Breiman referred to Adaboost as "the best off-the-shelf classifier in the world' [21]. We tune both the number of iterations of the boosting algorithm performed, and the algorithm used for the boosting. Simple classification trees (the same Breiman et al. algorithm used above) are used as the base learning for the Adaboost models, as is conventional. We use the Adaboost algorithm as a a stand-in for the much more common random forest method (see Figure 1); the random forest method did not allow us to test consistent values of the *mtry* parameter, number of variables to consider at each split, because this value depends on the number of variables in a dataset.

- **Support Vector Machine**: Support Vector Machines were one of the first statistical techniques widely used for dropout modeling in MOOCs [33]. SVMs construct a "maximum margin hyperplane" to identify the boundary which best separates different outcome classes in the data, and are noted for being less susceptible to outliers because of their focus only on points close to the separating boundary. We use only a linear kernel SVM in this experiment, tuning the gamma (cost) parameter. We center and scale the predictors when using the SVM algorithm in this experiment.

- **Naïve Bayes**: Naïve Bayes models are probabilistic classifiers that make the relatively strong assumption of independence between features, but are often used as a baseline classifier in predictive modeling experiments. Figure 1 shows that network-based models, including Naïve Bayes, are among the most popular in learning analytics models. We tune both the Laplace smoothing correction and the kernel for the Naïve Bayes classifiers in this experiment. For both Naïve Bayes classifier and Support Vector Machines, we perform the additional preprocessing step of dropping predictors which display zero variance within each outcome class.

A summary of the models considered is shown in Table 5. Together, these algorithms represent a broad spectrum of model types, including relatively simple, high-bias parametric models and complex, flexible, non-parametric models. They also constitute a representative sample of the models most often used for dropout modeling tasks, as shown in Figure 1. In total, our experiment considered 96 candidate models (where each model is a fixed algorithm-hyperparameter arrangement).

Algorithm	**Hyperparameters Tuned**	**Total Models**
Classification Tree	Cost-complexity parameter (cp)	4
Logistic Regression	L2 (ridge) penalty term (λ)	5
Adaboost	Boosting Algorithm ("Real Adaboost", M1) Number of iterations	6
SVM	Cost (γ)	5
Naive Bayes	Laplacian smoothing (fL) Kernel	4

Table 5: Algorithms used and hyperparameters tuned for each algorithm. Note that each configuration of each algorithm was used with each of the four feature types, yielding a total of $24 \times 4 = 96$ models; see also Table 4 and Figure 6.

The experimental setup, shown in Figure 6, is as follows: For each course session, we extract all four sets of features (clickstream, forum, assignments, all). We train each of the 96 candidate model on each feature set, using 5x2-fold cross-validation, using the same random cross-validation folds for every model, and recording the Area Under the Receiver Operating Characteristic (AUC) as the evaluation metric for each model. We retain the fold-level results (the model performance on the held-out model fold), which provides 10 estimates of model performance for every model on each session of the course (a sample of this data is shown in Table 6). We then apply three methods for model evaluation to this fold-level data (naïve average, null-hypothesis statistical testing or NHST, and Bayesian model evaluation), both to demonstrate the family of "best" models selected by each approach, and to reveal the pairwise comparisons across all models (as a measure of the ability of the NHST and Bayesian methods' respective ability to make decisions about classifier performance comparisons).

ROC	Resample	Course	Session	Model
0.597	Fold1.Rep1	1	1	adaboost ($NIter = 50$ Boosting = M1 Features = Quiz(
0.594	Fold1.Rep1	Digital Democracy	1	adaboost ($NIter = 100$, Boosting = M1, Features = Quiz)
0.597	Fold1.Rep1	Digital Democracy	1	adaboost ($NIter = 500$, Boosting = M1, Features = Quiz)
0.467	Fold1.Rep1	Digital Democracy	1	adaboost ($NIter = 50$, Boosting = Ada, Features = Quiz)
0.572	Fold1.Rep1	Digital Democracy	1	adaboost ($NIter = 100$, Boosting = Ada, Features = Quiz)
0.514	Fold1.Rep1	Digital Democracy	1	adaboost ($NIter = 500$, Boosting = Ada, Features = Quiz)
0.601	Fold2.Rep1	Digital Democracy	1	adaboost ($NIter = 50$, Boosting = M1, Features = Quiz)
0.601	Fold2.Rep1	Digital Democracy	1	adaboost ($NIter = 100$, Boosting = M1, Features = Quiz)
...	...	...	...	...
0.719	Fold1.Rep1	Digital Democracy	1	L2LR ($\lambda = 1$, Features = Clickstream)
0.748	Fold1.Rep1	Digital Democracy	1	L2LR ($\lambda = 0.1$, Features = Clickstream)
0.742	Fold1.Rep1	Digital Democracy	1	L2LR ($\lambda = 0.01$, Features = Clickstream)
0.743	Fold1.Rep1	Digital Democracy	1	L2LR ($\lambda = 0.001$, Features = Clickstream)
0.743	Fold1.Rep1	Digital Democracy	1	L2LR ($\lambda = 0$, Features = Clickstream)
0.701	Fold2.Rep1	Digital Democracy	1	L2LR ($\lambda = 1$, Features = Clickstream)
...	...	...	...	...
0.523	Fold1.Rep1	Digital Democracy	1	SVM ($\gamma = 10$, Features = Forum)
0.529	Fold1.Rep1	Digital Democracy	1	SVM ($\gamma = 1$, Features = Forum)
0.517	Fold1.Rep1	Digital Democracy	1	SVM ($\gamma = 0.1$, Features = Forum)
0.518	Fold1.Rep1	Digital Democracy	1	SVM ($\gamma = 0.01$, Features = Forum)
0.521	Fold1.Rep1	Digital Democracy	1	SVM ($\gamma = 0.001$, Features = Forum)
0.507	Fold2.Rep1	Digital Democracy	1	SVM ($\gamma = 10$, Features = Forum)
0.508	Fold2.Rep1	Digital Democracy	1	SVM ($\gamma = 1$, Features = Forum)

Table 6: A sample of the experimental data generated from the modeling process. Each feature/algorithm/hyperparameter is trained and tested using the same random cross-validation folds on each individual course dataset, resulting in fold-level model performance data for each unique model.

5 Results Analysis

With four feature sets, 96 candidate models, and 10 iterations of training/testing on 85 course sessions, this resulted in a total of $4 \times 96 \times 10 \times 85 = 184,320$ total observations of model performance. This indeed demonstrates an "embarassment of riches" we face in the field of learning analytics. In this section, we present three approaches to sifting through this abundance of model performance data to choose the "best" models – those with the best generalization performance across our broad sample of MOOCs, or the best predicted performance on a future (unseen) dataset. Identifying these models is a primary goals of most predictive modeling experiments.

Below, we present three approaches to identifying the family of models with the best generalization performance. All models provide similar recommendations about the highest-performing model (because each method is applied to

the same model performance data, and the top models had relatively consistent and strong performance across multiple datasets and measures of performance). However, these models vary significantly in terms of (a) the inferences we van draw from each method; (b) the size of the family of "best" models, and (c) the ability of each technique to make decisions about comparisons between the entire space of models. We demonstrate (i) the NHST-based method's limitations with the large (but realistic) number of comparisons being performed on this dataset, which results in an inability to discern differences between more than 50% of the models considered and a family of nearly 20 "best" models; (ii) that the Bayesian approach provides the most useful, robust, and reliable statistical inferences about the model performance; and (iii) that the results of the Bayesian technique allow us to draw useful conclusions about the best sets of features and algorithms for the dropout prediction problem (and which features and algorithms are practically equivalent).

5.1 Naïve Average Method

In the naïve average model evaluation approach, the experimental results are simply averaged, sorted according to performance, and the model(s) with the best average performance are selected. Due to space constraints, we present only the top and bottom 20 rows of the complete results table (which contains 96 entries); these results are shown in Table 7.

Algorithm	Feature Type	Hyperparameters	Average AUC
CART	All	0.001	0.9010
Adaboost	All	50 Boosting = M1	0.8981
Adaboost	All	100 Boosting = M1	0.8979
Adaboost	All	500 Boosting = M1	0.8969
CART	All	0.01	0.8936
Adaboost	Clickstream	50 Boosting = M1	0.8928
Adaboost	Clickstream	100 Boosting = M1	0.8923
CART	Clickstream	0.001	0.8922
Adaboost	Clickstream	500 Boosting = M1	0.8908
CART	Clickstream	0.01	0.8800
NB	Clickstream	0 Kernel = True _1	0.8727
NB	Clickstream	1 Kernel = True _1	0.8727
NB	Clickstream	1 Kernel = False	0.7911
NB	Clickstream	0 Kernel = False	0.7911
L2LR	All	0_0.01	0.7799
L2LR	All	0_0.001	0.7791
L2LR	All	0_0	0.7791
L2LR	All	0_0.1	0.7791
L2LR	Clickstream	0_0	0.7520
L2LR	Clickstream	0_0.001	0.7520
...	...	...	...

Table 7: Average model performance results. Using the "naïve average" method, we would select the model configuration with the best average performance as the "best" model in this experiment; in this case, this is a classifiation tree with complexity parameter 0.001 and all feature types (Clickstream + Forum + Assignments).

Applying naïve average method, we would conclude that the decision tree (CART) algorithm, with all features and a cost-complexity parameter of 0.001, is the model with the best generalization performance in this experiment: the decision tree with these features and hyperparameters achieves the highest average AUC. The "family" of best models is a model of one, and every pairwise difference, where any discernible difference exists, is considered significant. Using the naïve average method, we do not consider the magnitude of the observed difference (which is smaller than 0.003, in the case of the first and second models), the total number of models considered ($k = 96$), the number of observations over which the experiment was conducted ($N = 48$ courses $\times$ 5 runs of 2-fold CV = 480 observations per model). We simply assume that the observed differences are accurate, that they are not spurious, and that they are significant enough to be of interest.

This experiment, which matches the realistic conditions under which many experiments are conducted (many models relative to the number of datasets, with small observed differences in performance), makes clear how problematic

the simple average method is. The observed difference between the CART algorithm and the next-highest performing algorithm is quite narrow; this difference is small enough to be spurious, and certainly small enough to be practically useless. The simple average does not consider this. Additionally, the simple average method simply ignores the fact that we compared 96 different models. We might expect to observe some differences in model performance with so many comparisons, perhaps even due to randomness in the cross-validation folds. Thinking in terms of Tukey's "deals" of hands of cards, the question of how many "deals" we made, and therefore how surprised we should be by the hand we have been dealt, is swept under the rug by the naïve average method.

Furthermore, the naïve average method provides no estimate of the confidence or significance of our results. While this might not be considered an issue in a single study, in particular, it provides no basis for comparison for future work which attempts to replicate these findings on new data: is our confidence low, in which case a different result be surprising? Or is our confidence quite high, in which case we would expect similarly strong results in replications? The simple average method provides no answer to these questions.

Finally, by failing to provide any basis for identifying potential equivalence between models, the naïve average method does not allow us to easily introduce other considerations, such as model interpretability or training time, into our decision. If we had a "family" of equally effective models to choose from, we might select one model based on these other considerations; however, under the naïve average method, we simply choose the best average model – even when the differences may be small enough to warrant choosing based on some other relevant criteria. If we do decide to choose some group of best models, there is no principled way to do so.

5.2 NHST-Based Method: Frequentist Nemenyi Test

An alternative, more principled approach, to model evaluation is based on the Nemenyi test, described in Section 3.2.2. This procedure simultaneously compares all of the algorithms considered in the experiment, with a two-stage, nonparametric approach. First, on each dataset, the Friedman test (Equation 5) is applied. The Friedman test can be thought of as a nonparametric version of the ANOVA test statistic, and determines whether the observed rankings deviate significantly from their expected distribution under the null hypothesis of equivalent average performance. If this test indicates a significant difference (as it did in this experiment), we proceed with a post-hoc Nemenyi test to conduct pair-wise comparisons and determine where significant differences between individual models may exist (similar to ANOVA, the Friedman test only indicates whether the data collectively show a deviation from the expected distribution; the Friedman test does not itself identify which models differ and is not calibrated to do so).

As we discuss above, this procedure is preferable to the naïve average approach because it uses some controls to avoid spurious results by accounting for both k and N in calculating the test statistic. The Nemenyi test is preferable

over other frequentist procedures for significance testing because this nonparametric procedure can be applied even to non-normal model performance data where the distributions may vary and be incommensurable across datasets. Typically, the results of this procedure are reported for all pairwise comparisons using a Critical Difference (CD) diagram, which is shown in Figure 7. However, the CD diagram is difficult to interpret with a large number of models; we instead present a "windowpane plot" of the results in Figure 8.

Clickstream	
Forum Views	Number of pageviews of forum pages.
Active Days	Number of days for which user registered any clickstream activity (maximum of 7).
Quiz Views	Number of pageviews of quiz attempt pages, as measured by clickstream features.
Exam Views	Number of pageviews of exam-type quiz pages, as measured by clickstream features.
Human-Graded Quiz Pageview	Number of pageviews of human-graded quiz pages, as measured by clickstream features.
Assignments	
Pre-Submission Lead Time	Time between a quiz submission and deadline for all submissions; discretized buckets for $t \geq 7$ days, $3 \leq t < 7$, $1 \leq t < 3$, $0 \leq t < 1$, and late.
Total Raw Points	Sum of total raw points earned on quizzes.
Average Raw Score*	Average raw score on all assignments.
Raw Points Per Submission	Total raw points divided by total submissions.
Total quiz submissions	Total count of quiz submissions.
Percent of allowed submissions	Total count of quiz submissions as a percent of the maximum allowed submissions.
Percent of max student submissions	A student total number of quiz submissions as a percent of the maximum number of submissions made by any student in the course.
Correct submissions percent*	Percentage of the total submissions that were correct.
Change in weekly average*	Difference between current week average and previous week average quiz grade.
Forum	
Number of Posts	Total number of posts.
Number of Replies	Number of posts by user which were replies to other users (i.e., not to themselves, and not first post in thread).
Average Post Sentiment	Average net sentiment of posts (positive - negative).
Average Post Length	Average length of posts, in characters.
Positive Posts	Number of posts with net sentiment ≥ 1 standard deviation above thread average.
Negative Posts	Number of posts with net sentiment ≤ -1 standard deviation below thread average.
Neutral Posts	Number of posts with net sentiment within 1 standard deviation of thread average.
Sentiment Relative to Thread	Average of (post sentiment - avg sentiment for thread).
Threads Started	Total number of threads initiated by student.
Unique Words/Bigrams	Count of unique words/bigrams used across all posts.
Flesch Reading Ease	Flesch Reading Ease score, discretized into separate features in increments of 10 from 0 to 100.
Flesch-Kincaid Grade Level	Flesch-Kincaid grade level, discretized into separate features in increments of 1 from 0 to 20.
Note Votes Received	Total net upvotes users' posts received (positive - negative).

Table 4: Feature name and definition by category. Each feature is calculated at the student-week level, resulting in $p \cdot n$ features at week n with one observation for each unique student. Sentiment was extracted using the `VADER` sentiment analyzer [**?**]. Fleisch-Kincaid readability and grade level scores calculated in accordance with [31]. Features marked with a (*) were calculated by quiz type (homework, quiz, and video), resulting in three different features, one per quiz type, using that definition.

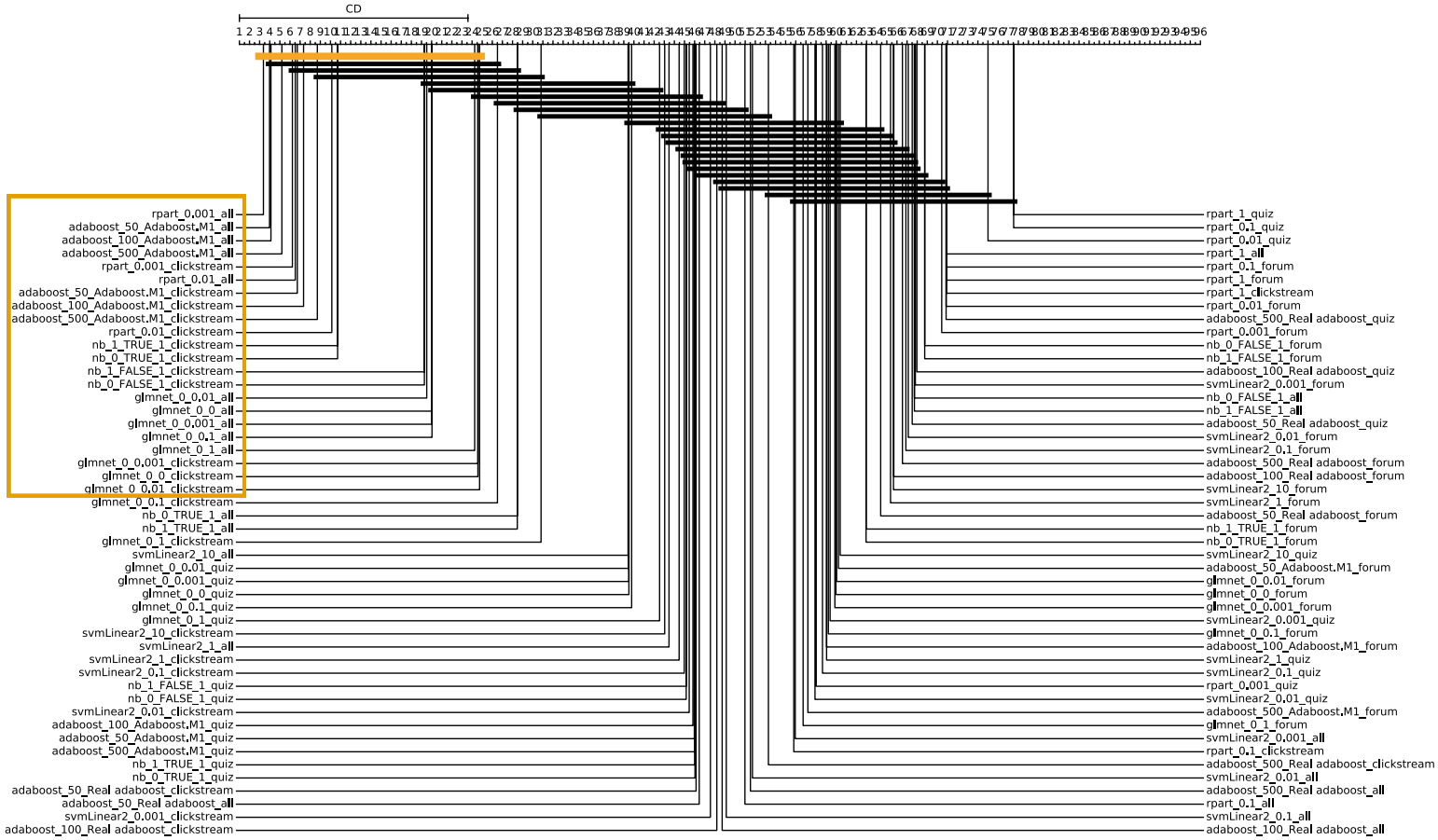

Figure 7: Critical Difference (CD) diagram for models evaluated in this experiment. The bold CD bars connect models statistically indistinguishable at $\alpha = 0.05$; the family of "best" models under the frequentist procedure is indicated in orange.

Algorithm	Feature Type	Hyperparameters	Avg. Rank	Avg. AUC	Diff. In Ranks	Diff. In AUC
CART	All	$cp = 0.001$	3.376	0.901	NA	NA
Adaboost	All	$NIter = 50$, Boosting = M1	3.978	0.899	-0.602	0.002
Adaboost	All	$NIter = 100$, Boosting = M1	4.118	0.899	-0.742	0.002
Adaboost	All	$NIter = 500$, Boosting = M1	5.198	0.897	-1.822	0.004
Adaboost	Clickstream	$NIter = 50$, Boosting = M1	6.725	0.89	-3.349	0.011
Adaboost	Clickstream	$NIter = 100$, Boosting = M1	7.344	0.889	-3.968	0.012
Adaboost	Clickstream	$NIter = 500$, Boosting = M1	8.704	0.887	-5.328	0.014
Naïve Bayes	Clickstream	$fL = 1$, Kernel = True	10.708	0.872	-7.332	0.029
Naïve Bayes	Clickstream	$fL = 0$, Kernel = True	10.708	0.872	-7.332	0.029
Naïve Bayes	Clickstream	$fL = 1$, Kernel = False	19.288	0.788	-15.911	0.113
Naïve Bayes	Clickstream	$fL = 0$, Kernel = False	19.288	0.788	-15.911	0.113
L2LR	All	$\lambda = 0.01$	19.521	0.78	-16.145	0.121
L2LR	All	$\lambda = 0$	20.036	0.779	-16.66	0.121
L2LR	All	$\lambda = 0.001$	20.036	0.779	-16.66	0.121
L2LR	All	$\lambda = 0.1$	20.039	0.778	-16.662	0.123
L2LR	All	$\lambda = 1$	24.258	0.752	-20.882	0.149
L2LR	Clickstream	$\lambda = 0.001$	24.578	0.75	-21.202	0.151
L2LR	Clickstream	$\lambda = 0$	24.578	0.75	-21.202	0.151
L2LR	Clickstream	$\lambda = 0.01$	24.744	0.75	-21.368	0.151

Table 8: Family of models which are statistically indistinguishable from the "best" model, according to the Friedman and Nemenyi Test procedure. These models all have differences in average rank (across each dataset) that is less than the critical difference of $CD = 22.5936$ relative to the highest-ranked model (CART with complexity parameter = 0.001). All differences are relative to this model, which had an average rank of 3.376, and average AUC of 0.901.

This analysis is not, however, strictly concerned with all 96 models and their relative performance. In particular, it is concerned with the realistic goal of identifying the best models for dropout prediction: that is, the feature-algorithm-hyperparameters which achieve the highest AUC. We therefore identify the collection of models which are statistically indistinguishable from the highest-performing model as the "family of highest-performing models". These models are shown in Table 8, and correspond to all of the models in Figure 7 which are connected by an orange CD bar. These models are also shown in the windowpane plot in Figure 8 as the model numbers which are colored white in the top row (which indicates that the models are statistically indistinguishable from the top-ranked model).

While we would like to conclude that the models shown in Table 8 are the "best" models, and while this is often implied by the results of frequentist analyses of model performance results, to do so would be incorrect. Instead, we

are not able to draw any conclusions regarding the differences in performance between these models using this procedure. As Demsar puts it in [15], these results indicate that, for this family of models, "the experimental data is not sufficient to reach any conclusion" (14). This is the inferential equivalent of not being able to reject the null hypothesis. A frequentist approach can never *prove* the null hypothesis of no effect, and instead simply produces inconclusive results; an NHST can never prove the equivalence of classifiers [35, 12]. This is a scientifically unsatisfying result – and, in particular, it provides no evidence about which of this "family" we should use.

In addition, as noted above, there are several ways in which NHST procedure's discrimination between significant and non-significant differences in model performance are unsatisfying. In sections 3.2 and 3.3.1, we discussed how the inferences of NHST procedures fail to differentiate between the *magnitude* and the *uncertainty* of the effects under examination. In this experiment, this is reflected in the best model family in Table 8. The NHST procedure does not discriminate between models with large differences in performance (high magnitude) but high variability (high uncertainty), and models with small differences in performance (low magnitude) but also low variability (low uncertainty). This results in a family of best models which fall along a spectrum; some models show large differences in average performance, but this difference is too variable to conclude with certainty that it constitutes a real effect. For these models, we might be interested in further exploring whether additional testing could reveal a true effect, which could result in large improvements in model performance (or avoid large *decreases* in model performance from choosing an "equivalent" model which is actually much worse at predicting dropout).

For example, consider all of the logistic regression models shown in the lower portion of Table 8; each of these models, on average, had an AUC more than 0.1 *worse* than the highest-performing model. This difference constitutes a practical and potentially important difference in model performance, and we might like to be aware of this difference when applying a procedure to select the ideal dropout prediction model. In contrast, consider the Adaboost models in Table 8 (the six highest-performing models after the decision tree). For these models, the magnitude is quite small, with the difference in average AUC relative to the best model never exceeding 0.01. However, the models are also considered statistically indistinguishable from the best model under the NHST procedure, because the observed difference in their ranks was small. This difference in performance might be considered *practically* less useful – an improvement in AUC of 0.003 or 0.004 would be of little import in most cases – but the NHST is not able to take the magnitude of this difference into consideration.

What if we wanted to consider the procedure's decisions across the entire space of models considered, not simply the family of "best" models – perhaps to learn something about which feature types were most important, or whether hyperparameter tuning significantly influenced the performance of specific models? Figure 8 answers this question, and shows the decisions of the NHST for each pairwise model comparison. Unfortunately, the procedure is only able to make decisions in 44.08% of the pairwise comparisons conducted, which provides little

information about many pairwise model comparisons (this procedure will make fewer decisions as the number of comparisons k grows relative to the number of datasets N). Figure 8 demonstrates how the frequentist method is only able to make decisions when the observed differences in performance are large, with significant differences only detected between the highest- and lowest-performing models. This leaves little room for interpretation of the relative difference between, for example, models using quiz vs. forum features (and leads us to conclude that, in most cases, no difference in performance can be detected with the available data).

The final problem with the NHST may seem pedantic, but is actually crucial to the argument of this work and the procedure of predictive model selection: the NHST does not tell us about what we want to know about our experiment. In fact, it allows us only to draw conclusions conditional on a hypothesis that is almost certainly false! As discuss in section 3.2 and 3.3.1, null hypothesis significance testing yields an estimate of the probability of observing the data (in this case, the difference in average rankings) *conditional on the truth of* H_0; that is, the probability of observing these data if all models actually had equivalent average performance: $P(Y|H_0)$. As many scholars have pointed out, this hypothesis is almost always incorrect [30, 25]. Estimating the probability of observing our results, conditional on a hypothesis that is false, yields little useful information about the actual probability of observing these results, and about the data themselves. What we would like to know is, quite simply, the probability of the null hypothesis itself, given the data: $P(H_0|Y)$. However, a NHST cannot estimate this probability, and therefore gives little statistically valid information about which model is best, or about pairwise comparisons between more than half of the models in our experiment; to do so, we must apply Bayesian model evaluation.

5.3 Bayesian Model Evaluation Method

We apply the Bayesian model evaluation method described in Section 3.3.1 to the same, fold-level model performance data. Recall that the Bayesian model evaluation procedure uses the difference in fold-level performance estimates for each pair of models (X, Y) to construct simulated draws from a Dirichlet distribution, and then draws repeated samples from this distribution to produce posterior estimates of $(P(X > Y), P(ROPE), P(X < Y))$, where ROPE is the "region of practical equivalence," and denotes that the difference between X and Y is smaller than some pre-specified threshold. For this experiment, as in previous work using this procedure [1], we use ROPE = 0.01; that is, models are considered practically equivalent if the difference in their AUC is smaller than 0.01.

The Bayesian procedure thus allows us to differentiate between two cases in which models which are clearly different: cases where the magnitude is small (the difference $|X - Y| \in ROPE$, and therefore the models are "practically equivalent"), and cases where this magnitude is large ($|X - Y| \notin ROPE$, the model performance is meaningfully different). This is useful, because we tend

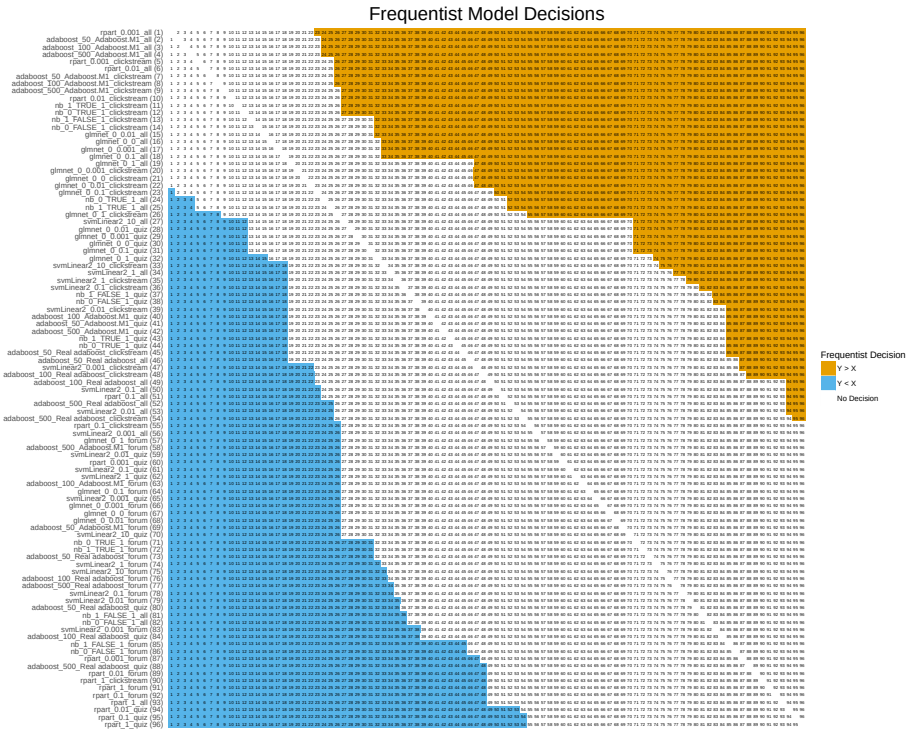

Figure 8: A "windowpane" plot showing decisions for frequentist method (adapted from [25]). All models are shown along each axis in identical order of decreasing performance. Each cell is labeled with the rank of the model along the x axis, and compares to the corresponding model listed on the y axis. Models colored orange or blue are places where the Frequentist method is able to decide as to one being significantly different than the other. White squares indicate a lack of decidability due to models being in the same Critical Difference Region. Due to the large number of comparisons, the Frequentist method is unable to draw conclusions about more than half of the pairwise comparisons (detecting significant differences in 44.08% of the 4560 comparisons).

Algorithm	Feature Type	Hyperparameters	Avg. Rank	Avg. AUC	Diff. In Ranks	Diff. In AUC
CART	All	$cp = 0.001$	3.376	0.901	-	-
Adaboost	All	$NIter = 50$, Boosting = M1	3.978	0.899	-0.602	0.002
Adaboost	All	$NIter = 100$, Boosting = M1	4.118	0.899	-0.742	0.002
Adaboost	All	$NIter = 500$, Boosting = M1	5.198	0.897	-1.822	0.004

Table 9: Family of models practically equivalent to the "best" model, according to the Bayesian model evaluation procedure. All models shown have a P(ROPE) greater than the decision threshold of 0.95 . Note that this threshold is somewhat arbitrary and adopted here to replicate [1]); however, adjustments of this threshold even to 0.999 had minimal effects on the Bayesian decision in most cases. In all cases here, P(ROPE) ≈ 1 according to the results of the MCMC sampling procedure described in Section 3.3.1. All differences are relative to the highest-ranked model (CART with complexity parameter = 0.001, which had an average rank of 3.376, and average AUC of 0.901. The Bayesian procedure's ability to directly test the hypothesis of equivalent performance allows it to make decisions on many model comparisons which the frequentist procedure cannot, yielding a far more precise and practically useful family of best models.

only to care about the latter; if the difference in performance of two models is small, we might consider other aspects of the model, such as training time or interpretability, in order to select which to use.

We present Bayesian equivalents of the results shown in previous sections in Table 9 and Figure 9. The Bayesian model evaluation procedure returns a more precise family of "best" models than the frequentist procedure, concluding that only 3 models are practically equivalent to the best overall model (where the region of practical equivalence is set to 1%).

Additionally, there is an important epistemic distinction between the best model family recommended by the Bayesian procedure and its frequentist equivalent. In the case of the frequentist models, recall that we were simply not able to conclude anything about the models shown – we simply cannot reject the null hypothesis of equivalence, but we do not *prove* or have strong reason to believe that these models are equivalent. The conclusion that "we cannot prove that these models are different" is simply the best we can do under the frequentist paradigm with the given data. However, the Bayesian procedure directly estimates the probability of two models being equal: recall that the Dirichlet model yields probability estimates for each of three potential outcomes for every (X,Y) pair of models: P(X better than Y), P(ROPE, or X and Y are practically equivalent), and P(Y better than X). Being able to model this practical equivalence gives the Bayesian approach a substantial advantage in this task: there are many comparisons for which the model performance is reliably within the region of practical equivalence (ROPE), and the Bayesian approach can assign

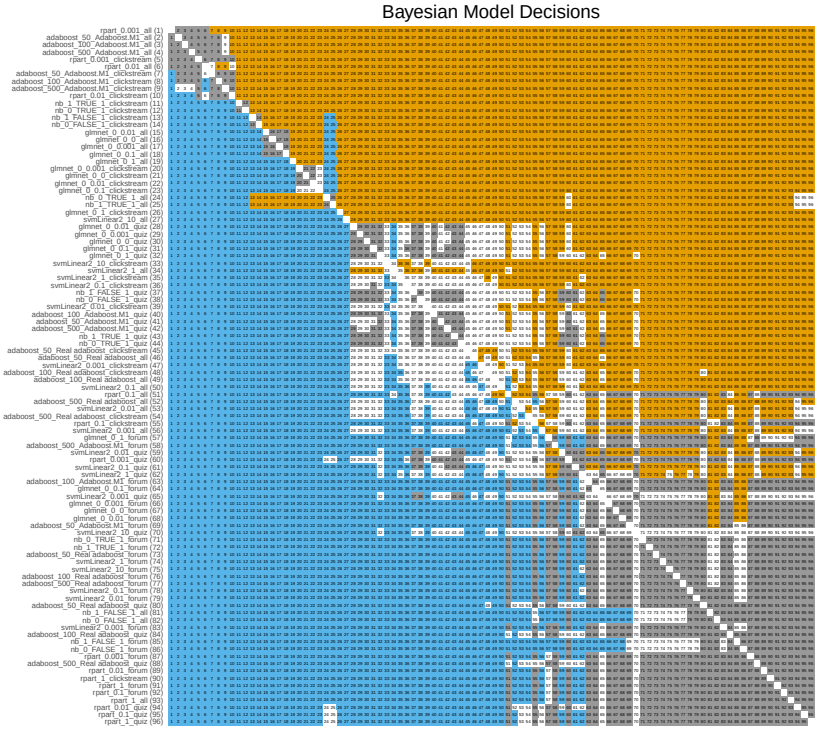

Figure 9: Windowpane plot showing decisions for Bayesian method. The Bayesian method is able to directly compute posterior probability that two models are within the "region of practical equivalence," a third possibility which is not directly estimable using a Frequentist method. The Bayesian method is able to make decisions in over 89% of pairwise comparisons.

high probability to ROPE in such cases. In contrast, the frequentist paradigm can never conclude that no difference exists; indeed, given enough data, as noted above, the frequentist paradigm will *always* conclude that a significant different exists (even when this difference is minimal).

The Bayesian model evaluation approach not only provides us with a smaller, more targeted set of models as our "best" dropout prediction model which is sensitive to the magnitude of the difference between models (only those with small differences in performance relative to the best model are considered; those with large, but highly variable, differences are excluded); it also allows us to make substantive conclusions about those models. Our conclusion is not conditional on the assumption of a null hypothesis of equivalent performance (which is almost certainly incorrect in most cases), and is instead only conditional on the data we observe.

We also briefly note that, while the naïve average approach may often be

preferred for its simplicity, or its ability to yield a single (or small set of) best model(s), the Bayesian approach nearly matches the precision of using a naïve average approach in this case, but instead of making strong assumptions about the significance of observed differences – recall that the naïve average method simply assumes that *all* differences are significant – the Bayesian approach uses a hierarchical modeling approach to actually estimate the probability that each pair of models is practically equivalent, given their observed performance.

We discussed previously the concerns about multiple comparisons which are often not addressed in model comparison experiments, and we briefly return to this consideration here. The frequentist method used here directly considers the number of comparisons conducted, as well as the number of datasets. We noted above that, when this controlling does occur, it makes this procedure especially sensitive to having a large number of datasets in order to control the size of the cricial difference (CD; Equation 6). Additionally, this requires researchers to track, report, and account for the full scope of comparisons performed in the course of an experiment – despite the fact that increasing the number of comparisons will reduce their ability to detect a statistically significant effect (by increasing the size of the CD). For Bayesian model evaluation, large numbers of comparisons are generally not considered to be a concern [25], because the concept of a Type I error does not exist under a Bayesian framework. Additionally, for cases in which we would like to specifically model the multiple comparisons, we could fit a Bayesian hierarchical model which can specifically account for the variability within and across models. The use of such a model was beyond the scope of this work, and our intent was to replicate an existing model evaluation method. However, we note that when hierarchical models are used, they tend to apply shrinkage to the estimates, reducing the estimate of effect sizes (in this case, differences in observed performance) [25]. However, given the level of confidence of most of the Bayesian decisions in this experiment, it is likely that a hierarchical model would not substantially change the Bayesian decision in most model comparisons.

Finally, analyzing the experimental results using the Bayesian method allows us to draw several useful and substantive conclusions from this modeling experiment which are relevant to future predictive modeling research in MOOCs. For example, Table 9 and Figure 9 both show that the family of four best models use *all* features – clickstream, forum, and assignments. This suggests that models can use these combined features to potentially exploit the information contained in each feature set to make better predictions than with only one of the feature sets. Additionally, Figure 9 provides further information on which individual feature sets might be most informative: by inspecting the model rankings, and the large "blocks" of models which are practically equivalent, we see that models using clickstream features consistently outperformed models using forum or assignment features, and in many cases clickstream-only models were competitive with otherwise-equivalent models using all features. In most cases, models with forum or assignment features were almost all indistinguishable from others using the same feature type, regardless of algorithm and hyperparameters used. Finally, Figure 9 demonstrates that, for many algorithms, hyperparameter tun-

ing appears to have had little effect, especially relative to the obvious effects of feature type. This suggests that future modeling efforts might achieve the most substantial performance improvements from feature engineering and using the correct data types (such as the activity data contained in clickstream data), not from developing sophisticated algorithms on uninformative feature sets (such as assignments which relatively few students complete).

6 Conclusion

In this work, we are concerned with advancing the state of the learning analytics field with respect to the evaluation of predictive models. We presented the results of a comprehensive literature review to assess the state of the practice in the field, and presented an overview of several techniques for model evaluation. By applying these results to a realistic and comprehensive case study using a large set of MOOC data, we demonstrated the differences in substantive conclusions supported by each method. In particular, we demonstrated the power of Bayesian model evaluation to draw highly precise, informative conclusions about the performance of feature-algorithm-hyperparamter combinations, even across a large space of candidate models. Under the Bayesian model evaluation method, our case study specifically demonstrated the importance of feature extraction to model performance, and in particular demonstrated the predictive performance of clickstream-based features for dropout prediction (and the relatively poor performance achieved by using only forum- or assignment-based features, regardless of the statistical model used).

While the fields of learning analytics and educational data mining are only a decade old[7] the techniques we have presented here borrow from the work of others done over the last 15 years in the broader machine learning community. With the growth in the size of datasets available (e.g. MOOCs), and the ability to run thousands of permutations of analyses on desktop hardware, we are concerned with the lack of rigor when selecting and reporting on the "best" predictive model. This is especially important as there a number of pragmatic elements when operationalizing predictive models – computational speed, robustness in the face of missing data, and interpretability – all of which might influence adoption ability. In this work we have shed light on techniques which can help inform these choices, and we look forward to the growth of rigor in the community as a result.

References

[1] A. Benavoli, G. Corani, J. Demsar, and M. Zaffalon. Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis. June 2016.

[7] Based on annual conferences in the area, though we note that educational data mining itself has roots in the much older Artificial Intelligence in Education (AIED) community.

[2] A. Birnbaum. Confidence curves: An omnibus technique for estimation and testing statistical hypotheses. *J. Am. Stat. Assoc.*, 56(294):246–249, June 1961.

[3] M. L. Bote-Lorenzo and E. Gómez-Sánchez. Predicting the decrease of engagement indicators in a MOOC. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, LAK '17, pages 143–147, New York, NY, USA, 2017. ACM.

[4] R. R. Bouckaert. Choosing between two learning algorithms based on calibrated tests. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, 2003.

[5] R. R. Bouckaert. Estimating replicability of classifier learning experiments. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 15–, New York, NY, USA, 2004. ACM.

[6] R. R. Bouckaert and E. Frank. Evaluating the replicability of significance tests for comparing learning algorithms. In *Advances in Knowledge Discovery and Data Mining*, pages 3–12. Springer, Berlin, Heidelberg, May 2004.

[7] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[8] C. G. Brinton and M. Chiang. MOOC performance prediction via clickstream data and social learning networks. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 2299–2307. ieeexplore.ieee.org, Apr. 2015.

[9] C. Brooks, C. Thompson, and S. Teasley. A time series interaction analysis method for building predictive models of learners using log data. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 126–135. ACM, Mar. 2015.

[10] C. Brooks, C. Thompson, and S. Teasley. Who you are or what you do: Comparing the predictive power of demographics vs. activity patterns in massive open online courses (MOOCs). In *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*, L@S '15, pages 245–248, New York, NY, USA, 2015. ACM.

[11] J. Cohen. [PDF]The earth is round (p ¡ .05). *American Psychologist*, 49(12):997–1003, Dec. 1994.

[12] G. Corani, A. Benavoli, J. Demšar, F. Mangili, and M. Zaffalon. Statistical comparison of classifiers through bayesian hierarchical modelling. Sept. 2016.

[13] Coursera. *Coursera Data Export Procedures*. Coursera, June 2013.

[14] S. Crossley, L. Paquette, M. Dascalu, D. S. McNamara, and R. S. Baker. Combining click-stream data with NLP tools to better understand MOOC completion. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, LAK '16, pages 6–14, New York, NY, USA, 2016. ACM.

[15] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7(Jan):1–30, 2006.

[16] J. Demšar. On the appropriateness of statistical tests in machine learning. In *Workshop on Evaluation Methods for Machine Learning in conjunction with ICML*, 2008.

[17] T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput.*, 10(7):1895–1923, Sept. 1998.

[18] 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. In *The 9th International Conference on Educational Data Mining*, pages 353–357. pnigel.com, 2016.

[19] B. J. Evans, R. B. Baker, and T. S. Dee. Persistence patterns in massive open online courses (MOOCs). *J. Higher Educ.*, 87(2):206–242, Mar. 2016.

[20] R. A. Fisher. *Statistical methods for research workers*. Genesis Publishing Pvt Ltd, 1925.

[21] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[22] M. Friedman. A comparison of alternative tests of significance for the problem of m rankings. *Ann. Math. Stat.*, 11(1):86–92, 1940.

[23] D. García-Saiz and M. Zorrilla. A meta-learning based framework for building algorithm recommenders: An application for educational arena. *J. Intell. Fuzzy Syst.*, 32(2):1449–1459, 2017.

[24] J. Gardner and C. Brooks. Student success prediction in MOOCs. Nov. 2017.

[25] A. Gelman, J. Hill, and M. Yajima. Why we (usually) don't have to worry about multiple comparisons. July 2009.

[26] A. Gelman and E. Loken. The garden of forking paths: Why multiple comparisons can be a problem, even when there is no "fishing expedition" or "p-hacking" and the research hypothesis was posited ahead of time. *Department of Statistics, Columbia University*, 2013.

[27] X. Hu, C. W. L. Cheong, W. Ding, and M. Woo. A systematic review of studies on predicting student learning outcomes using learning analytics. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, LAK '17, pages 528–529, New York, NY, USA, 2017. ACM.

[28] R. Hubbard and M. J. Bayarri. P values are not error probabilities. *Available in Internet*, 2003.

[29] N. Japkowicz and M. Shah. *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, Jan. 2011.

[30] D. D. Jensen and M. D. Schmill. Adjusting for multiple comparisons in decision tree pruning. In *KDD*, pages 195–198. ocs.aaai.org, 1997.

[31] J. P. Kincaid, R. P. Fishburne, Jr, R. L. Rogers, and B. S. Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch, 1975.

[32] R. F. Kizilcec and S. Halawa. Attrition and achievement gaps in online learning. In *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*, L@S '15, pages 57–66, New York, NY, USA, 2015. ACM.

[33] 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. aclweb.org, 2014.

[34] S. Kotsiantis, K. Patriarcheas, and M. Xenos. A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education. *Knowledge-Based Systems*, 23(6):529–535, Aug. 2010.

[35] J. K. Kruschke. Bayesian estimation supersedes the t test. *J. Exp. Psychol. Gen.*, 142(2):573–603, May 2013.

[36] W. Li, M. Gao, H. Li, Q. Xiong, J. Wen, and Z. Wu. Dropout prediction in MOOCs using behavior features and multi-view semi-supervised learning. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 3130–3137, July 2016.

[37] X. Li, T. Wang, and H. Wang. Exploring n-gram features in clickstream data for MOOC learning achievement prediction. In *Database Systems for Advanced Applications*, pages 328–339. Springer, Cham, Mar. 2017.

[38] B. B. McShane, D. Gal, A. Gelman, C. Robert, and J. L. Tackett. Abandon statistical significance. Sept. 2017.

[39] Y. Meier, J. Xu, O. Atan, and M. v. der Schaar. Predicting grades. *IEEE Trans. Signal Process.*, 64(4):959–972, Feb. 2016.

[40] C. Nadeau and Y. Bengio. Inference for the generalization error. *Mach. Learn.*, 52(3):239–281, 2003.

[41] S. Nagrecha, J. Z. Dillon, and N. V. Chawla. MOOC dropout prediction: Lessons learned from making pipelines interpretable. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 351–359, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.

[42] K. Pearson. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, 1900.

[43] J. Qiu, J. Tang, T. X. Liu, J. Gong, C. Zhang, Q. Zhang, and Y. Xue. Modeling and predicting learning behavior in MOOCs. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, pages 93–102, New York, NY, USA, 2016. ACM.

[44] C. Robinson, M. Yeomans, J. Reich, C. Hulleman, and H. Gehlbach. Forecasting student achievement in MOOCs with natural language processing. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, LAK '16, pages 383–387, New York, NY, USA, 2016. ACM.

[45] C. P. Rosé, R. Carlson, D. Yang, M. Wen, L. Resnick, P. Goldman, and J. Sherer. Social factors that contribute to attrition in MOOCs. In *Proceedings of the First ACM Conference on Learning @ Scale Conference*, L@S '14, pages 197–198, New York, NY, USA, 2014. ACM.

[46] T. C. Russo and J. Koesten. Prestige, centrality, and learning: A social network analysis of an online class. *Commun. Educ.*, 54(3):254–261, July 2005.

[47] T. Sinha, N. Li, P. Jermann, and P. Dillenbourg. Capturing "attrition intensifying" structural traits from didactic interaction sequences of MOOC learners. Sept. 2014.

[48] C. Taylor, K. Veeramachaneni, and U.-M. O'Reilly. Likely to stop? predicting stopout in massive open online courses. Aug. 2014.

[49] C. Tucker, B. K. Pursel, and A. Divinsky. Mining student-generated textual data in MOOCs and quantifying their effects on student performance and learning outcomes. *Computers in Education Journal*, 5(4):84–95, 2014.

[50] J. W. Tukey. The philosophy of multiple comparisons. *Stat. Sci.*, 6(1):100–116, 1991.

[51] M. Vartak, H. Subramanyam, W.-E. Lee, S. Viswanathan, S. Husnoo, S. Madden, and M. Zaharia. M odel DB: a system for machine learning model management. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, page 14. dl.acm.org, 2016.

[52] K. Veeramachaneni, U.-M. O'Reilly, and C. Taylor. Towards feature engineering at scale for data from massive open online courses. July 2014.

[53] Y. Wang. *Demystifying Learner Success: Before, During, and After a Massive Open Online Course*. PhD thesis, Teachers College, Columbia University, Mar. 2017.

[54] R. L. Wasserstein and N. A. Lazar. The ASA's statement on p-values: Context, process, and purpose. *Am. Stat.*, 70(2):129–133, Apr. 2016.

[55] M. Wen, D. Yang, and C. Rose. Sentiment analysis in MOOC discussion forums: What does it tell us? In *Educational data mining 2014*. educationaldatamining.org, 2014.

[56] J. Whitehill, K. Mohan, D. Seaton, Y. Rosen, and D. Tingley. Delving deeper into MOOC student dropout prediction. Feb. 2017.

[57] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, Oct. 2016.

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