UPSC2M: Benchmarking Adaptive Learning from Two Million MCQ Attempts

Kevin Shi

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2025-74
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2025/EECS-2025-74.html

May 15, 2025

UPSC2M: Benchmarking Adaptive Learning from Two Million MCQ Attempts

by Kevin Shi

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Signed by:

Jitendra Malik

16D473A00244427...

Professor Jitendra Malik
Research Advisor

5/14/2025

(Date)

* * * * * * *

DocuSigned by:

Kurt Keutzer

44443E7EACAC4D6...

Professor Kurt Keutzer
Second Reader

5/15/2025

(Date)

UPSC2M: Benchmarking Adaptive Learning from Two Million MCQ Attempts

Kevin Shi

UC Berkeley
kevinxpshi@berkeley.edu

Abstract

We present **UPSC2M**, a large-scale dataset comprising two million multiple-choice question attempts from over 46,000 students, spanning nearly 9,000 questions across seven subject areas. The questions are drawn from the Union Public Service Commission (UPSC) examination, one of India's most competitive and high-stakes assessments. Each attempt includes both response correctness and time taken, enabling fine-grained analysis of learner behavior and question characteristics. Over this dataset, we define two core benchmark tasks: **question difficulty estimation** and **student performance prediction**. The first task involves predicting empirical correctness rates using only question text. For this, we benchmark several baselines and introduce **LLM-Guided Feature Regression (LFR)**, a content-based regression pipeline that leverages question features extracted by large language models. The second task focuses on predicting the likelihood of a correct response based on prior interactions. Here, we evaluate standard approaches and propose **Subject Knowledge Tracking (SKT)**, a lightweight knowledge-tracking algorithm for subject-level proficiency modeling. Together, the dataset and benchmarks offer a strong foundation for building scalable, personalized educational systems. We release the dataset and code to support further research at the intersection of content understanding, learner modeling, and adaptive assessment: github.com/kevins-hi/upsc2m.

1 Introduction

As digital learning platforms become increasingly central to education, there is growing demand for intelligent systems that can adapt to individual learners, curate relevant content, and deliver targeted assessments (Woolf, 2009). At the heart of such systems lie two fundamental modeling tasks: estimating the difficulty of educational content (Lord, 1980; Blum and Corter, 2014) and predicting student performance (Corbett and Anderson, 1994; Pavlik Jr et al., 2009; Piech et al., 2015). These capabilities underpin a wide range of applications—from personalized question selection to real-time learner diagnostics. When combined, they serve as the foundation for fully automated adaptive learning systems that dynamically tailor instruction based on both content complexity and learner proficiency.

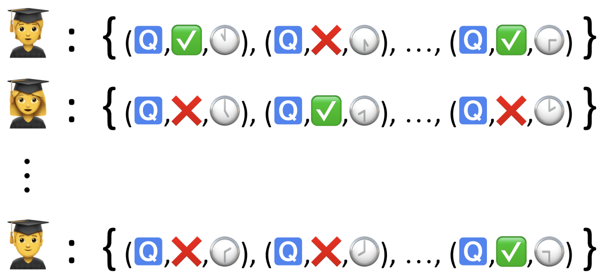

Figure 1: UPSC2M visualized as a list of students, each associated with a set of question attempts. Each attempt records the student ID, question ID, selected answer, whether it was correct, and the time taken to answer.

Statistic	Count
Unique Students	46,235
Unique Questions	8,973
Total Interactions	1,962,573

Table 1: Summary statistics for the UPSC2M dataset.

Much of the existing work in educational modeling has relied on small-scale classroom data or narrow subject domains, limiting the development and evaluation of models suited to real-world settings (Stamper et al., 2011; Corbett and Anderson, 1994; Pavlik Jr et al., 2009). To bridge this gap,

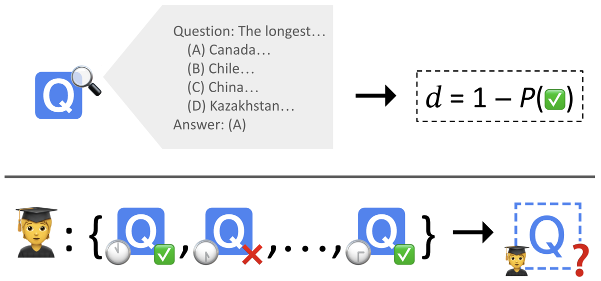

Figure 2: Illustration of the two benchmark tasks: question difficulty estimation (top) and student performance prediction (bottom). In the first task, the goal is to estimate the difficulty of a question—defined as one minus the empirical probability of a correct response—based solely on its text. In the second task, given a student's prior question attempts, predict whether the student will correctly answer a new, unseen question.

we introduce UPSC2M, a large-scale dataset comprising 1,962,573 question attempts from aspirants preparing for the Union Public Service Commission (UPSC) examination—one of India's most competitive standardized tests. Covering 8,973 questions across seven subjects, UPSC2M captures correctness and timing data from 46,235 students.

We propose two core tasks supported by this dataset. The first is *Question Difficulty Estimation*, where models predict empirical difficulty using only question text. The second is *Student Performance Prediction*, where models forecast whether a student will answer a question correctly, given their prior interactions. These tasks reflect core challenges in real-world adaptivity and serve as modular building blocks for intelligent tutoring and assessment systems. To benchmark progress, we implement several baselines and introduce two novel methods: *LLM-Guided Feature Regression (LFR)* for interpretable content-based prediction, and *Subject-Knowledge Tracking (SKT)* for lightweight, subject-specific proficiency modeling.

Our contributions are threefold: (1) We release UPSC2M, a large-scale dataset capturing both question content and behavioral interaction data in a high-stakes, multi-subject testing context. (2) We define and benchmark two core tasks—difficulty estimation and performance prediction—using a diverse set of baseline and proposed models. (3) We demonstrate how interpretable, lightweight modeling approaches (LFR and SKT) can match or exceed the performance of more complex alternatives while maintaining transparency and extensibility.

Together, UPSC2M and its benchmark tasks provide a robust foundation for research in scalable personalized education. By supporting more accurate models of question difficulty and student performance, this work helps lay the groundwork for educational platforms that adapt to individual needs at scale, expanding access to high-quality, personalized learning for students regardless of background or location.

2 Related Work

Large-scale Interaction Datasets A number of publicly available datasets have driven progress in student modeling and adaptive learning. The PSLC DataShop repository provides tens of thousands of student–problem interactions across diverse domains (Stamper et al., 2011), and the ASSISTments dataset offers fine-grained logs of middle-school mathematics practice. More recently, EdNet—a hierarchical dataset of over 130 million interactions from an online tutoring platform—has enabled deep sequence models at unprecedented scale (Choi et al., 2020). Our dataset, UPSC2M, complements these by focusing on a highly competitive, multi-subject examination context, capturing both correctness and response-time signals for UPSC aspirants.

Question Difficulty Estimation Classical item response theory (IRT) models difficulty as a latent parameter estimated from response patterns (Lord, 1980), but they rely solely on interaction counts. Recent work has explored textual and semantic features to predict question difficulty directly from content (Blum and Corter, 2014). By pairing a large, annotated UPSC question bank with empirical accuracy rates, UPSC2M supports both purely content-based difficulty regression and hybrid approaches that integrate behavioral priors.

Feature Extraction with LLMs Recent work has explored using LLMs to extract features that capture deeper cognitive and linguistic aspects of question difficulty, such as generating reasoning traces and modeling response uncertainty (Feng et al., 2025), deriving linguistic and cognitive features for down-

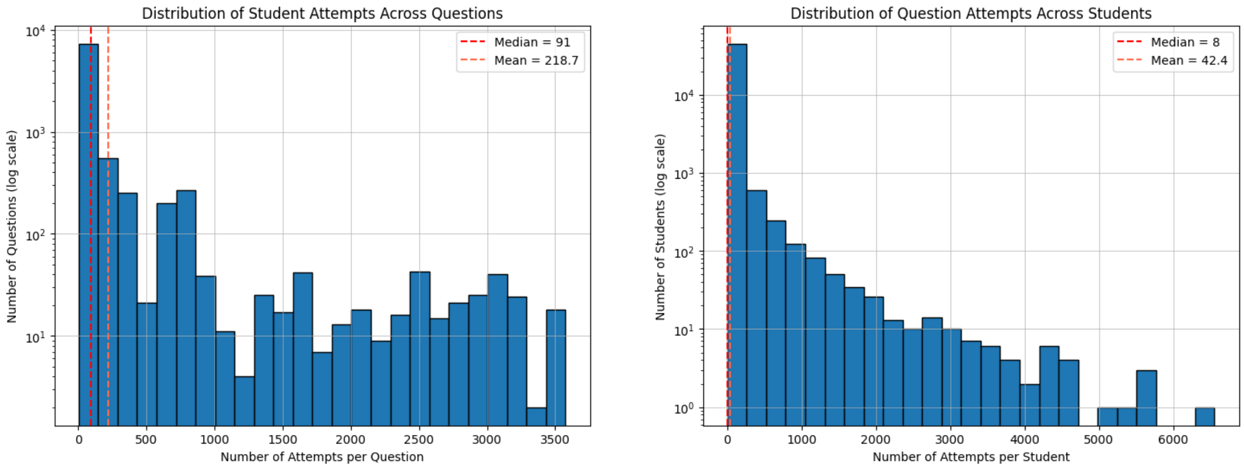

Figure 3: Distributions of interaction counts in the UPSC2M dataset. The left panel shows the number of student attempts per question, and the right panel shows the number of question attempts per student. Both distributions are plotted on a log scale and annotated with their respective median and mean values, illustrating a long-tailed pattern in which many questions and students have relatively few interactions.

stream regression (Razavi and Powers, 2025), and predicting reading comprehension difficulty aligned with IRT-based scores (Jain et al., 2025). LFR builds on these advances, combining structured LLM-derived features with empirical accuracy data to support interpretable and scalable difficulty estimation in the UPSC setting.

Student Performance Prediction Predicting learner outcomes has a long history in educational data mining. Bayesian Knowledge Tracing (BKT) (Corbett and Anderson, 1994) and Performance Factor Analysis (PFA) (Pavlik Jr et al., 2009) established early probabilistic frameworks for tracking mastery. The advent of neural methods—e.g. Deep Knowledge Tracing (DKT) (Piech et al., 2015) has further improved sequence-based prediction. The UPSC2M dataset, with its detailed question content, student attempt outcomes, and rich temporal metadata, offers a new testbed for benchmarking such models on high-stakes exam data.

Applications for Adaptive Testing Adaptive testing algorithms—such as computerized adaptive testing (CAT) (Weiss, 2011)—depend critically on calibrated item difficulties and real-time performance estimates. Datasets that combine content features with large-scale attempt logs enable the development of more responsive and personalized CAT systems. We anticipate that UPSC2M will spur new advances in adaptive exam design, question selection strategies, and real-time learner diagnostics.

3 Proposed Dataset

3.1 The UPSC Exam

The Union Public Service Commission Examination is a highly competitive annual assessment conducted in India to recruit candidates into prestigious government positions such as the Indian Administrative Service, Indian Police Service, and Indian Foreign Service. Each year, approximately one million aspirants take the exam, highlighting its widespread appeal among graduates from diverse educational backgrounds. Given the limited number of available positions—typically around one thousand—the UPSC exam is regarded as one of India's most challenging and high-stakes competitive assessments.

The exam consists of three primary stages—Preliminary, Main, and Interview—conducted over several months. This paper specifically focuses on Paper I of the Preliminary Examination, also known as the General Studies Paper. Paper I is a two-hour objective assessment consisting of 100 multiple-choice questions designed to evaluate candidates' factual recall and analytical reasoning abilities. The questions cover a variety of topics, including Indian history, geography, polity, economics, science and technology, environmental issues, and current affairs.

		Students per Question			Questions per Student		
Subject	**Question Count**	Mean	Median	Max	Mean	Median	Max
Current Affairs	1793	127.79	112	3576	20.13	5	1502
Polity	1487	348.00	79	3284	19.31	5	1425
History	1449	259.72	77	2559	20.94	5	1227
Economy	1111	183.86	72	1728	20.17	5	1069
Science	1094	139.81	19	2869	11.48	5	1008
Environment	1022	181.63	104	2801	11.70	4	913
Geography	1017	291.82	145	3055	19.93	5	956
Overall	**8973**	**218.72**	**91**	**3576**	**42.45**	**8**	**6553**

Table 2: Per-subject statistics in the UPSC2M dataset, including the number of questions and summary statistics for student and question engagement—measured as students per question and questions per student.

The UPSC exam offers a rich testbed for evaluating adaptive learning technologies due to its large-scale, diverse participant base and well-structured, high-coverage question design. With one million annual aspirants—drawn from varied educational backgrounds and engaging deeply with standardized multiple-choice questions spanning multiple domains—it enables fine-grained modeling of learner behavior, question difficulty, and performance trends. The exam's high-stakes nature further ensures genuine learner engagement, yielding robust data for developing and validating personalization algorithms and predictive analytics in educational contexts.

3.2 Data Collection

To support research on adaptive learning algorithms, we developed and deployed Padhai, a learning platform specifically tailored for UPSC aspirants. Through Padhai, students interacted with a curated repository of approximately 10,000 multiple-choice questions closely aligned with the style and format of the UPSC examination. Over a two-year period, we collected question attempts from around 50,000 learners, resulting in over two million interactions.

To enhance dataset quality and reduce noise from repeated question exposure, interactions were deduplicated by retaining only the first encounter each student had with a given question. Following additional filtering to remove students and questions with insufficient interactions, the final cleaned dataset comprises 1,962,573 question attempts from 46,235 students across 8,973 questions. The dataset has been thoroughly anonymized to protect student privacy while preserving essential patterns and signals crucial for downstream modeling tasks.

3.3 Dataset Schema

We release UPSC2M, a large-scale dataset comprising two components: an *attempts dataset* and a *questions dataset*. Each row in the attempts dataset represents a single interaction between a student and a question, capturing key fields including `user_id`, `question_id`, `user_answer`, `user_correct`, and `time_taken`. The accompanying questions dataset provides metadata for each question, including its `id`, `subject`, question `stem`, multiple-choice `options`, and the correct `answer`. While no student metadata is included, the dataset enables rich behavioral analysis: the `user_answer` field supports investigations into distractor effectiveness and common misconceptions, while the `time_taken` field—measured in seconds—offers a proxy for question engagement and fluency under time pressure. Each question is constrained to a 60-second limit, mirroring the real-world pacing of the UPSC exam.

3.4 Dataset Statistics

UPSC2M exhibits substantial scale and diversity in learner behavior across content categories. As shown in Table 2, each question is attempted by an average of 219 students, with some questions receiving over 3,000 attempts. This breadth of coverage stems from both the temporal dynamics of question exposure—where older or more prominently featured questions accumulate more interactions—and varying levels of learner interest across subject areas. Such variation necessitates models capable of generalizing across both high-frequency and low-frequency questions.

The average student attempted 42 questions, with the most active student answering over 6,500. This long-tailed distribution, typical of open educational platforms, supports modeling across a wide range of engagement levels. However, the low median number of questions per student indicates that many students engage only briefly, emphasizing the need for models that are robust to cold-start scenarios and sparse interaction histories.

4 Question Difficulty Estimation

4.1 Problem Formulation

We propose a task to estimate the empirical difficulty of a multiple-choice question using only its textual content. Each question is represented as a tuple `(id, subject, stem, options, answer)`, where `stem` denotes the question prompt, `options` is a list of four candidate choices, and `answer` specifies the index of the correct option.

The empirical difficulty of a question is defined as $1 - p_{\text{correct}}$, rounded to two decimal places, where p_{correct} denotes the proportion of students in UPSC2M who answered the question correctly among those who attempted it. This definition reflects the intuition that more difficult questions are associated with lower observed accuracy.

Setup To support reproducible evaluation, the questions dataset includes a predefined `split` field designating train, validation, and test partitions in a 70/15/15 split. Each question is also annotated with a precomputed `difficulty` score based on the formulation above.

4.2 Text Embedding Regression

As a baseline for question difficulty estimation, we adopt a simple regression approach. Specifically, we encode the question using a frozen pretrained text encoder (Devlin et al., 2019; Reimers and Gurevych, 2019; Neelakantan et al., 2022) and train a small MLP to predict the associated difficulty.

Each question is serialized as a single string combining the stem and options, which is then passed through OpenAI's `text-embedding-3-large` model—a general-purpose text embedding model. The resulting fixed-dimensional embedding serves as input to an MLP trained to minimize mean squared error against the ground-truth difficulty scores provided in the dataset. This approach offers a lightweight text-to-score mapping that sets a lower bound for models leveraging richer representations.

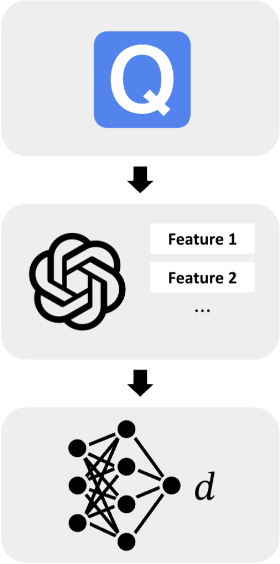

Figure 4: The LLM-Guided Feature Regression (LFR) pipeline. A question is processed by an LLM to extract a set of interpretable features. These features are then used by a small neural network to predict the question's difficulty score d.

4.3 LLM-Guided Feature Regression (LFR)

We introduce *LLM-Guided Feature Regression*, a pipeline that augments content-based difficulty estimation by leveraging LLMs and standard NLP techniques to extract interpretable, domain-informed features from each question (Feng et al., 2025; Razavi and Powers, 2025; Jain et al., 2025). These features serve as inputs to a small MLP trained to regress onto the empirical difficulty score defined in Section 4.1. We hypothesize that this approach taps into the LLM's world knowledge and reasoning ability to capture deeper aspects of question difficulty.

The extracted features reflect common intuitions behind human judgments of question difficulty, including the obscurity of required knowledge, clarity of phrasing (Boyce-Jacino and DeDeo, 2019), and quality of distractors (Rezigalla et al., 2024). This feature-driven approach is model-agnostic and extensible: future improvements in LLM capabilities or prompting strategies may yield more accurate or

Method	**RMSE**	**MAE**	**R^2**
Training Mean	0.2057	0.1699	-0.0001
Text Embedding	0.1910	0.1543	0.1375
LFR	**0.1786**	**0.1447**	**0.2457**

Table 3: Test set performance of text-based regression models for question difficulty estimation. The *Training Mean* baseline predicts the mean difficulty for all training samples. *LFR* refers to *LLM-Guided Feature Regression*.

nuanced representations, and the feature set can be expanded without significant changes to the downstream regression architecture.

We extract the following features:

Obscurity The LLM is prompted to identify the minimal set of external knowledge facts needed to answer the question and rate each fact on a 3-point obscurity scale: 1 (common knowledge), 2 (general knowledge), and 3 (specialized knowledge). We compute the mean, sum, and max obscurity scores across the identified facts. Higher obscurity values may indicate questions that require rarer or more specialized knowledge, making them more difficult for the average test-taker.

Ambiguity The LLM rates the phrasing of the question on a 1–5 scale of linguistic ambiguity, where 1 indicates a highly straightforward question and 5 indicates a highly ambiguous or tricky one. Ambiguous wording can mislead test-takers and increase the likelihood of incorrect answers, even when relevant knowledge is present.

Distractor Quality (DQ) The LLM evaluates each of the three distractors on a 1–5 scale of plausibility, where 1 indicates an implausible option and 5 indicates a highly plausible one. We include the average and maximum plausibility scores as features, omitting the sum due to redundancy. High-quality distractors make a question more challenging by increasing the cognitive effort required to eliminate incorrect options.

Reading Difficulty We compute the Flesch Reading-Ease of the question text to approximate its reading complexity (Flesch, 1948). Complex sentence structure or vocabulary can hinder comprehension, especially under time pressure, thus increasing effective difficulty.

Negation Presence A binary feature indicating whether the question stem contains a negation (e.g., "not", "never"), which often introduces subtlety or increases the potential for confusion. Negations can can easily lead to misinterpretation, particularly when paired with tricky answer choices.

Named Entities The number of unique named entities mentioned in the question, serving as a proxy for factual density. A higher count suggests greater demands on recall.

Together, these features form a compact, interpretable question representation that incorporates both linguistic and conceptual signals. While lightweight compared to end-to-end modeling with LLMs, this approach benefits from the LLM's reasoning capacity in a structured and controllable manner, providing a middle ground between black-box embeddings and handcrafted heuristics.

4.4 Results

As shown in Table 3, the *Text Embedding* baseline achieves modest improvements over the *Training Mean* predictor, reducing RMSE by 7.2% and MAE by 9.2%. This suggests that general-purpose semantic embeddings encode some information relevant to question difficulty, though the gains remain limited—highlighting the challenge of predicting question difficulty from surface-level textual features alone.

Incorporating structured LLM-derived features yields further improvement. *LFR* achieves the lowest error, with a 13.2% RMSE reduction and 14.8% MAE reduction relative to the *Training Mean* predictor. The R^2 score also rises substantially, indicating that interpretable linguistic and conceptual features explains a greater portion of the observed variance in difficulty.

While these results demonstrate the utility of LLM-guided feature engineering, a substantial gap remains to be closed. Much of the difficulty signal remains unmodeled, motivating future work that incorporates richer or more targeted features capable of capturing deeper pedagogical or cognitive cues.

4.5 Ablations

To better understand the contribution of individual components in the LFR pipeline, we conduct a series of ablations. First, we assess the impact of each

Ablated Feature	RMSE	MAE	R^2
Ambiguity Score	0.1898	0.1553	0.1488
Distractor Quality	0.1824	0.1491	0.2133
Obscurity Score	0.1815	0.1490	0.2212
Negation Presence	0.1801	0.1465	0.2331
Named Entities	0.1794	0.1462	0.2391
Reading Difficulty	0.1786	0.1450	0.2461
None (All Features)	**0.1786**	**0.1447**	**0.2457**

Table 4: Feature-wise ablation results for the LLM-Guided Feature Regression (LFR) model on the question difficulty estimation task. Removing any single feature generally reduces performance, with the largest degradation observed when omitting the ambiguity, obscurity, or distractor quality features.

feature by independently removing it from the input set and retraining the regression model. Across all ablations, performance degrades relative to the full-feature model, confirming that each feature contributes non-trivially to prediction accuracy.

Second, we evaluate the effect of using two OpenAI models for feature generation: the more capable `gpt-4.1` and the lighter-weight `gpt-4o-mini`. In general, features derived from the stronger model yield superior performance, especially for *obscurity* and *distractor quality*, which likely benefit from richer world knowledge and more accurate factual reasoning. A surprising exception arises with the *ambiguity* feature: replacing the weaker model with the stronger one leads to a noticeable drop in overall performance. Closer inspection reveals that the stronger LLM consistently assigns low ambiguity scores, resulting in low-variance, less informative features. In contrast, the weaker model produces a more differentiated distribution of ambiguity scores, better capturing the variability needed for the regression model to learn useful distinctions.

Finally, we experimented with several classical regression algorithms—including linear regression, random forests, and support vector regression—but found that a shallow MLP consistently achieved the best performance.

4.6 Discussion

While presented here as a benchmark task, automatic estimation of question difficulty has broad practical value for educational applications. In adaptive learning platforms, accurate difficulty prediction enables dynamic content personalization—matching questions to a learner's current proficiency to maintain engagement and promote effective learning. This is especially critical in open educational environments with diverse student populations and wide variability in prior knowledge (Dagunduro et al., 2024).

LLM Assignment	RMSE	MAE	R^2
Swap Ambiguity	0.1913	0.1558	0.1354
Swap Obscurity	0.1822	0.1489	0.2150
Swap DQ	0.1819	0.1495	0.2180
All `gpt-4o-mini`	0.1838	0.1508	0.2014
Default (Hybrid)	**0.1786**	**0.1447**	**0.2457**

Table 5: Ablation results for the choice of LLM used to generate each feature in the LFR pipeline. *DQ* denotes distractor quality. The *Default (Hybrid)* configuration uses `gpt-4.1` for obscurity and distractor quality, and `gpt-4o-mini` for ambiguity. The first three rows show the effect of individually swapping each LLM-based feature from the default setting. Notably, replacing `gpt-4o-mini` with `gpt-4.1` for ambiguity—effectively using `gpt-4.1` for all features—substantially degrades performance, indicating that `gpt-4o-mini` is better suited specifically for generating ambiguity scores.

Beyond personalization, automatic difficulty estimation supports large-scale content management. Automated models can assist in auditing question banks for redundancy, identifying overly easy or hard items, and calibrating the difficulty distribution of assessments. Such capabilities streamline content curation, helping educators organize lesson plans and construct balanced practice sets with minimal manual effort.

In generative settings, where LLMs are used to create new multiple-choice questions (Raina and Gales, 2022), difficulty estimation models can serve as lightweight verifiers. By flagging questions that fall outside a desired difficulty range, these models help ensure the pedagogical utility of generated content.

As educational platforms scale across diverse curricula and learner populations, accurate question difficulty estimation will become a cornerstone of personalized adaptive learning infrastructure.

5 Student Performance Prediction

5.1 Problem Formulation

We propose a task to predict whether a student will answer a given multiple-choice question correctly, based on their prior interaction history. Each row in the attempts dataset represents a single interaction and is formatted as a tuple (`user_id`, `question_id`, `user_answer`, `user_correct`, `time_taken`), where `user_correct` is a binary label indicating whether the response was correct.

For evaluation, the fields `user_answer`, `user_correct`, and `time_taken` are treated as target variables—models may access them during training but must not use them as input features at inference time. At test time, each example is defined solely by the pair (`user_id`, `question_id`), and the model must predict whether the student answers the question correctly.

Formally, this task involves estimating the conditional probability that a student answers a question correctly, given their historical behavior. This formulation mirrors real-world scenarios in adaptive learning systems, where predicting a learner's performance is essential.

Setup To facilitate reproducible evaluation, the attempts dataset includes a predefined `split` field that assigns each interaction to the training, validation, or test set, following an 80/10/10 ratio. The split is randomized at the interaction level, with post-processing to ensure that all students and questions in the validation and test sets also appear in the training set. This constraint ensures that models are evaluated on their ability to generalize to new interactions, rather than on cold-start cases with unseen students or questions.

5.2 Baselines

To contextualize the performance of more sophisticated models, we evaluate several simple baselines for this task.

Random and Zero Predictors As naive reference points, we consider two trivial classifiers. The *Random* baseline predicts correctness by sampling from the empirical label distribution in the training set, which shows a slight class imbalance (59.81% incorrect). The *Zero Predictor* always predicts the majority class (0 for incorrect), thereby serving as a worst-case lower bound on accuracy and calibration. While uninformative, these baselines are useful for verifying that more complex models exploit meaningful structure in the data.

Difficulty-Based Heuristic As a simple yet informative baseline, we ignore the student's interaction history and estimate the probability of a correct response based solely on the difficulty of the target question. Specifically, we compute the predicted probability as $1 - d$, where d denotes the difficulty score of the question. This formulation assumes that all students have an equal chance of answering a question correctly, modulated only by how empirically difficult the question is for the population.

We evaluate this heuristic using two variants: one based on the ground-truth difficulty labels defined in Section 4.1, and another using the predicted difficulty scores from the best-performing question difficulty estimation model described in Section 4.3. The former represents an oracle, reflecting perfect knowledge of item-level performance, while the latter provides a more realistic assessment of how automated difficulty estimation can inform student response prediction. Comparing these variants highlights the impact of difficulty estimation accuracy on this task.

Despite its simplicity, this baseline captures coarse priors over questions and highlights the influence of item difficulty on student performance. Comparing it to history-aware models underscores the value of incorporating personalized behavioral signals.

5.3 Collaborative Filtering

To assess the utility of standard recommender system techniques for modeling student performance, we evaluate several collaborative filtering (CF) (Goldberg et al., 1992; Sarwar et al., 2001; Su and Khoshgoftaar, 2009) methods that treat the task as a matrix completion problem. The student-question interaction matrix is constructed from observed correctness labels, and models are trained to predict whether a student will answer a given question correctly.

We include matrix factorization methods such as Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF), which learn low-dimensional embeddings for students and questions based on historical responses. We also evaluate a bias-only model that estimates correctness using ad-

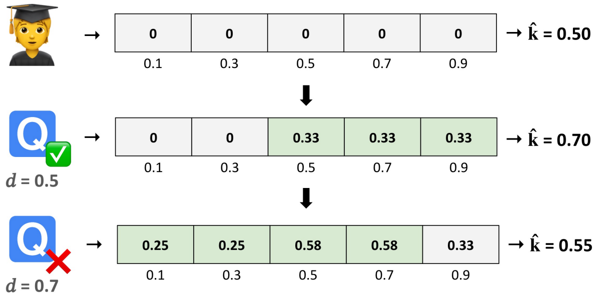

Figure 5: Three steps of the Subject Knowledge Tracking (SKT) algorithm. Each row represents the student's knowledge estimate as a distribution over discretized difficulty levels. When the student answers a question, SKT updates the distribution: if the answer is correct, bins with midpoints higher than the question's difficulty are increased; if incorrect, bins with lower midpoints are increased. The total update mass of 1 is evenly distributed across the highlighted (green) bins. The estimated proficiency $\hat{k}$ is then computed as a weighted mean using Equation 2, defaulting to 0.5 when no observations are available. The figure simplifies SKT by showing 5 bins and a single distribution per student, while our full model uses 10 bins per distribution and maintains 7 subject-specific distributions per student.

ditive student and item biases, as well as a K-nearest neighbors (KNN) approach that aggregates correctness labels from similar students. These methods represent a range of personalization strategies, from simple bias modeling to latent and neighborhood-based techniques.

Together, they provide a classical baseline family for student performance prediction, offering insight into how much signal can be captured from past interactions alone, without access to question content.

5.4 Subject Knowledge Tracking (SKT)

We introduce *Subject Knowledge Tracking* (SKT), a knowledge-tracking baseline that models each student's subject-specific proficiency using interpretable scalar values. The core idea is to estimate, for each subject, a latent knowledge parameter $k \in [0, 1]$ representing the student's probability of success on questions of that difficulty level.

Each student is assigned a separate coarse-grained probability density vector over the unit interval for each of the seven subject categories. The range $[0, 1]$ is discretized into 10 equal-width bins. Each bin stores the estimated probability mass that the student's true k-value for that subject lies within that range. These vectors are initialized uniformly and updated over the training set using the student's question outcomes.

When a student answers a question correctly, SKT adds mass to all bins with midpoints greater than or equal to the difficulty of the question. If the student answers incorrectly, the update is applied to all bins with midpoints less than or equal to the difficulty. The total update magnitude is fixed at 1 and is evenly distributed among the affected bins:

$$\Delta = \frac{1}{\text{\# bins to update}} \tag{1}$$

This mechanism gradually concentrates probability mass in regions of the k-vector consistent with observed performance.

To make predictions, SKT computes the expected knowledge value $\hat{k}$ for a student in the question's subject by taking the weighted average of the bin midpoints under the current probability distribution:

$$\hat{k} = \frac{\sum_{i=1}^{N} w_i \cdot m_i}{\sum_{i=1}^{N} w_i} \tag{2}$$

where w_i is the mass in bin i, m_i is the midpoint of bin i, and N is the number of bins. The probability that the student answers a given question correctly is then estimated by comparing $\hat{k}$ against the difficulty of the question:

$$\hat{y}_{u,q} = \mathbb{I}[\hat{k}_{u,\text{subject}(q)} \geq d_q] \tag{3}$$

where d_q is the question's difficulty and $\mathbb{I}$ is the indicator function.

SKT is evaluated using both ground-truth question difficulties and predictions from the best-performing question difficulty estimation model. This allows us to assess how sensitive SKT is to the quality of difficulty estimates and to isolate the contribution of the knowledge-tracking mechanism itself.

This approach offers a lightweight and interpretable model of student knowledge that captures subject-specific proficiency and its interaction with item difficulty. SKT can be extended in several directions, such as learning richer knowledge representations beyond a single scalar, incorporating subcategory-specific tracking, or adapting the update rule based on question properties or recency.

Method	**Accuracy**	**Precision**	**Recall**	**F1**	**AUC**	**Brier**
Random	0.5204	0.4005	0.4020	0.4012	0.5000	0.2400
Zero Predictor	0.6002	0.0000	0.0000	0.0000	0.5000	0.3998
Heuristic (GT)	0.6698	0.6231	0.4405	0.5161	0.7118	0.2080
Heuristic (Pred)	0.5688	0.4379	0.2771	0.3394	0.5232	0.2492
SKT (GT)	**0.6783**	0.6266	**0.4832**	**0.5456**	–	–
SKT (Pred)	0.5795	0.4614	0.3091	0.3702	–	–
KNN CF	0.6429	0.5817	0.3802	0.4599	0.6461	0.2330
SVD CF	0.6755	**0.6319**	0.4508	0.5262	0.7133	**0.2076**
NMF CF	0.6757	**0.6867**	0.3471	0.4612	**0.7157**	0.2100
Bias Only CF	**0.6788**	0.6312	**0.4731**	**0.5408**	**0.7210**	**0.2051**

Table 6: Test set performance of baseline methods on the student performance prediction task. *CF* denotes collaborative filtering. **Bolded** entries indicate the top two performing models for each metric. *GT* denotes models evaluated using ground-truth question difficulty, while *Pred* refers to those using predicted difficulty scores. *SKT* (Subject Knowledge Tracking) variants do not produce probability estimates and therefore omit AUC and Brier Score.

5.5 Results

Table 6 reports the performance of all baseline models on the student performance prediction task. The *Heuristic (GT)* model substantially outperforms trivial baselines, demonstrating that question difficulty alone provides a strong prior for estimating student success. This suggests that well-estimated item-level difficulty can serve as a meaningful signal, even without any personalization.

Among collaborative filtering methods, *Bias Only* yields the highest overall performance, while more expressive models such as *SVD*, *NMF*, and *KNN* fail to produce significant gains. The high sparsity of the student-question matrix (99.62%) likely limits the ability of these models to learn effective representations or student neighborhoods, constraining their ability to capture student-specific patterns beyond simple item and student-level tendencies.

The *SKT (GT)* model performs comparably to the strongest CF baseline, ranking among the top two models in accuracy, recall, and F1 score. Its competitive performance is notable given its simplicity and parameter-free design. Rather than learning latent vectors, SKT relies on interpretable, subject-specific bin updates to track student knowledge. That it matches the performance of matrix factorization approaches suggests that structured knowledge tracking—grounded in difficulty comparisons—can serve as a viable alternative to classical collaborative filtering. Moreover, SKT can be naturally extended with more expressive representations of student knowledge.

Performance drops substantially when predicted difficulty scores are used in place of ground-truth values, as seen in both the *Heuristic (Pred)* and *SKT (Pred)* variants. This highlights the sensitivity of downstream models to the quality of difficulty estimates: inaccurate priors diminish the value of both item-based and personalized approaches. Improving difficulty estimation is thus critical for enabling effective student performance prediction in fully automated pipelines.

Overall, the results underscore the value of combining high-quality item priors with interpretable, student-specific modeling. Simple but principled approaches like SKT offer a promising and extensible alternative to data-hungry latent factor models, particularly in sparse educational settings.

5.6 Ablations

We conduct a series of ablations to understand the sensitivity of SKT to its core design choices. These experiments shed light on how each component contributes to model stability, interpretability, and predictive performance.

Bin Resolution SKT discretizes the $[0, 1]$ interval into 10 equal-width bins by default. We ablate the number of bins and observe that both finer (e.g., 20 or 50 bins) and coarser (e.g., 5 bins) settings lead to reduced accuracy. Finer discretizations in-

Ablation	Accuracy	Precision	Recall	F1 Score
5 Bins	0.6777	0.6256	0.4824	0.5447
20 Bins	0.6779	0.6260	0.4823	0.5449
Zero Initialization	0.6757	0.6207	0.4856	0.5449
Gaussian Initialization	0.6783	0.6265	0.4832	0.5456
Update Size = 2	0.6780	0.6258	0.4839	0.5458
Update Size = 5	0.6773	0.6241	0.4847	0.5457
Update Size = 10	0.6767	0.6229	0.4848	0.5453
Unnormalized Magnitude	0.6764	0.6434	0.4276	0.5138
Default	**0.6783**	**0.6266**	**0.4832**	**0.5456**

Table 7: Ablation results for the Subject-Knowledge Tracking (SKT) model on the student performance prediction task. Performance generally declines when altering the number of bins, increasing the update size, or removing normalization. Gaussian and uniform initializations perform comparably and both outperform zero initialization; the uniform prior is favored for its simplicity and neutrality. The *Default* setting uses 10 bins, a uniform initialization with 0.5 mass per bin, and a total update size of 1. In the *Unnormalized Magnitude* variant, the full update size is applied to each updated bin individually rather than distributed evenly.

cur additional computational and storage overhead, while coarser ones lose resolution. The 10-bin setting achieves the best balance, offering the highest accuracy alongside interpretability and efficiency.

Uniform Initialization We experiment with several initialization schemes, including zero, uniform, and Gaussian priors. Uniform initialization—assigning 0.5 to each bin—yields the highest validation accuracy, though its performance is nearly identical to that of the Gaussian prior. Compared to the sharp early updates caused by zero initialization and the rigidity of peaked priors, the uniform prior provides a simple, neutral starting point.

Update Size We experiment with scaling the total update magnitude beyond the default value of 1. Larger update sizes cause the bin distributions to converge more slowly, as updates shift probability mass more aggressively, often overshooting the true knowledge region and requiring additional corrections. However, we observe no meaningful gains in predictive performance across a range of values. This suggests that slower convergence does not necessarily lead to better generalization, and a moderate update size strikes a desirable balance between responsiveness and stability.

Normalized Update Magnitude By default, SKT distributes a fixed total mass of 1 across all bins affected by each update. This normalization ensures that no single interaction disproportionately alters the probability distribution. Removing this normalization—i.e., adding a unit mass to each affected bin—causes instability. In particular, rare or inconsistent observations (e.g., a single incorrect response after many correct ones) lead to abrupt shifts in the estimated knowledge distribution, degrading overall prediction accuracy. Normalizing the update magnitude mitigates this effect by smoothing the impact of outliers.

5.7 Discussion

Student performance prediction is central to adaptive educational systems, enabling platforms to tailor instruction to each learner's evolving proficiency (Woolf, 2009). By estimating the likelihood of a correct response, these models support a range of applications, including personalized question selection, targeted review recommendations, and adaptive pacing—ultimately improving engagement and learning outcomes across diverse student populations.

Performance prediction paired with difficulty estimation forms the backbone of fully automated adaptive learning. Difficulty scores provide item-level priors, while student models capture behavioral patterns to personalize predictions. Together, these components enable systems that construct entire assessment paths on the fly—adjusting scope, granularity, and content coverage to optimize learning trajectories with minimal human input.

As educational platforms scale, the ability to jointly model questions and learners becomes increasingly vital (Dagunduro et al., 2024). These predictive capabilities move us closer to truly individualized learning—where each student receives the right content at the right time—making such modeling a cornerstone of scalable, data-driven personalization.

6 Limitations

Our question difficulty estimation labels are based solely on correctness rates and ignore temporal or student-specific variation; future work may redefine difficulty through joint modeling of student and item characteristics, potentially incorporating response times. Our collaborative filtering models are likely hindered by the high prevalence of low-activity learners—the median questions attempted per student is just 8—which may limit generalization and overall performance. Additionally, SKT does not leverage cross-student learning, unlike collaborative filtering. A hybrid approach may better balance generalization and interpretability. None of our current models incorporate response time features, which could offer valuable signals related to fluency or hesitation. Finally, while UPSC2M is large and diverse, its focus on one high-stakes exam context may limit direct transferability to other educational domains. Despite these limitations, we view our dataset and task formulations as a strong foundation for building more expressive, interpretable, and personalized models of learner behavior.

7 Conclusion

We present UPSC2M, a large-scale dataset of nearly two million question attempts from aspirants preparing for a high-stakes Indian examination, supporting two benchmark tasks: question difficulty estimation and student performance prediction. Our LLM-Guided Feature Regression pipeline yields interpretable features that outperform text embeddings for difficulty prediction, while our lightweight Subject Knowledge Tracking method matches collaborative filtering models in accuracy with greater transparency. Beyond benchmarking, UPSC2M provides a practical foundation for building adaptive educational tools that leverage calibrated difficulty and student modeling to support applications such as personalized sequencing, targeted review, and dynamic assessment (Weiss, 2011). A promising direction for future work is to integrate these signals into generative pipelines to enhance LLM-generated question quality (Raina and Gales, 2022), filter miscalibrated items, and scaffold coherent learning trajectories. Together, the dataset and benchmark tasks offer a robust testbed for advancing scalable, personalized education that connects question design, learner modeling, and real-time adaptivity—ultimately broadening access to high-quality, individualized learning for diverse student populations.

References

Au Blum and James E. Corter. 2014. Estimating question difficulty and user ability in a collaborative question answering community. In *Workshop on Personalized and Adaptive Learning in EDM*.

Christina Boyce-Jacino and Simon DeDeo. 2019. Opacity, obscurity, and the geometry of question-asking. *Cognition*, 193:104026.

Youngduck Choi, Youngnam Lee, Dongmin Shin, Junghyun Cho, Seoyon Park, Seewoo Lee, Jineon Baek, Chan Bae, Byungsoo Kim, and Jaewe Heo. 2020. Ednet: A large-scale hierarchical dataset in education. In *Proceedings of the 21st International Conference on Artificial Intelligence in Education (AIED 2020)*, volume 12164 of *Lecture Notes in Computer Science*, pages 69–73. Springer.

Albert T. Corbett and John R. Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*.

Adebukola Olufunke Dagunduro, Chidinma Favour Chikwe, Olanike Abiola Ajuwon, and Ayo Amen Ediae. 2024. Adaptive learning models for diverse classrooms: Enhancing educational equity. *International Journal of Applied Research in Social Sciences*, 6(9):2228–2240.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*.

Wanyong Feng, Alexander Scarlatos, David Smith, Sam Woodhead, and Andrew S. Lan. 2025. Reasoning and sampling-augmented mcq difficulty prediction via llms. *arXiv preprint arXiv:2503.08551*.

Rudolf Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221–233.

David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. 1992. Using collaborative filtering to weave an information tapestry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 165–170. ACM.

Yoshee Jain, John Hollander, Amber He, Sunny Tang, Liang Zhang, and John Sabatini. 2025. Exploring the potential of large language models for estimating the reading comprehension question difficulty. *arXiv preprint arXiv:2502.17785*.

Frederic M. Lord. 1980. *Applications of Item Response Theory to Practical Testing Problems*. Erlbaum.

Arvind Neelakantan, Kelvin Xu, Yi Tay, Ashwin Paranjape, Yichen Zhang, Alec Radford, David Krueger, Bryan McCann, Sam Shleifer, Barret Zoph, and 1 others. 2022. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*.

Philip I. Pavlik Jr, Hui Cen, and Kenneth R. Koedinger. 2009. Performance factors analysis: A new alternative to knowledge tracing. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education (AIED)*, volume 200 of *Frontiers in Artificial Intelligence and Applications*, pages 531–538. IOS Press.

Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, pages 505–513.

Vatsal Raina and Mark Gales. 2022. Multiple-choice question generation: Towards an automated assessment framework. *arXiv preprint arXiv:2209.11830*.

Pooya Razavi and Sonya J. Powers. 2025. Estimating item difficulty using large language models and tree-based machine learning algorithms. *arXiv preprint arXiv:2504.08804*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of EMNLP*.

Assad Ali Rezigalla, Ali Mohammed Elhassan Seid Ahmed Eleragi, Amar Babikir Elhussein, Jaber Alfaifi, Mushabab A. ALGhamdi, Ahmed Y. Al Ameer, Amar Ibrahim Omer Yahia, Osama A. Mohammed, and Masoud Ishag Elkhalifa Adam. 2024. Item analysis: the impact of distractor efficiency on the difficulty index and discrimination power of multiple-choice items. *BMC Medical Education*, 24(1):445.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM.

John C. Stamper, Kenneth R. Koedinger, Ryan S. J. d. Baker, Alida Skogsholm, Brett Leber, Sandy Demi, Shawnwen Yu, and Duncan Spencer. 2011. Datashop: A data repository and analysis service for the learning science community. In *Proceedings of the 15th International Conference on Artificial Intelligence in Education (AIED)*, page 628, Berlin, Heidelberg. Springer Berlin Heidelberg.

Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009:1–19.

David J. Weiss. 2011. *Adaptive Testing*. Oxford University Press.

Beverly Park Woolf. 2009. *Building Intelligent Interactive Tutors: Student-Centered Strategies for Revolutionizing E-learning*. Morgan Kaufmann, San Francisco.