International Journal of Research and Development in Applied Science and Engineering (IJRDASE)
ISSN: 2454-6844

Smriti: A Machine Learning-Driven Forgetting Curve Predictor for Personalized Student Memory Optimization

Gaurav Kr. Verma, Abhishek Yadav, Pankaj Kumar, Sachin Nirmal, Naimisha Awasthi
Dept of Computer Science & Engineering (AI-ML),
Bansal Institute of Engineering and Technology, Lucknow,
gaurav.contact@proton.me, abhishekyadav809040@gmail.com, kr.pankaj.098419@gmail.com, nirmalsachin49@gmail.com, naimishaawasthi56@gmail.com

Abstract: Memory fades. That is not a weakness — it is how the brain works. Ebbinghaus proved this in 1885: without review, a student forgets nearly 70% of what they studied within 24 hours. Yet most digital tools built for students either ignore this completely or apply the same fixed review schedule to everyone, regardless of how fast that individual actually forgets.

Smriti — from the Sanskrit स्मृति, meaning memory — is an AI-powered web application that predicts when a student will forget a topic, before it happens. Rather than guessing, it uses three machine learning models trained on 500,000 real learning records from Duolingo's Half-Life Regression dataset. A Polynomial Regression pipeline forecasts retention over 30 days, achieving $R^2 = 0.9539$ and RMSE = 0.0582 — outperforming the SM-2 algorithm by 79.6% in prediction error. A Decision Tree Classifier labels each topic as Strong, At-Risk, or Weak. K-Means Clustering groups topics with similar forgetting patterns for batch review.

Beyond prediction, Smriti includes an AI quiz module powered by Llama 3.3-70B via the Groq API, generating questions across all six levels of Bloom's Taxonomy using real-time Wikipedia and DuckDuckGo context. A gamification layer — XP points, streaks, and competitive leagues — keeps students engaged over time.

The result is a free, fully deployed, open-access tool that makes personalized memory optimization practically available to every student.

Keywords: Forgetting Curve, Spaced Repetition, Half-Life Regression, Polynomial Regression, Decision Tree, K-Means Clustering, Bloom's Taxonomy, Memory Optimization, Educational AI, Personalized Learning.

1. Introduction:

Learning and forgetting operate as two sides of the same coin. In 1885, Hermann Ebbinghaus demonstrated that retention R declines exponentially as a function of elapsed time t, relative to the half-life h of a memory: R(t)=e^(-t/h) [1]. Even after nearly a century and a half since that insight, the overwhelming majority of student-oriented technology either ignores the dynamics of memory altogether or applies them in ways that feel crude. Calendar-based nudges, applications for static note-taking, and even widely adopted flashcard systems such as Anki [4] and Quizlet [5] tend to rely on review intervals that are either fixed or derived from simple heuristics. They do not meaningfully adapt to the rate at which an individual learner actually loses hold of a given concept.

One can observe the tangible fallout from this mismatch in exam venues across the globe. Students frequently report studying content that simply will not surface during test-ing-not from a shortage of diligence, but because the moment chosen for review did not align with their personal forgetting trajectory. Evidence from the Institute for Education Sciences suggests that when retrieval practice occurs at optimal intervals, long-term retention can improve by as much as 50% compared with massed study [6]. The obstacle, therefore, is not a lack of motivation. It is a computational one: forecasting the exact point at which a specific student is on the verge of forgetting a specific concept, and then timing a retrieval cue to match.

Smriti tackles this issue through a layered architecture of machine learning components. Rather than imposing uniform intervals, the system builds an empirically derived forgetting curve for every pairing of student and topic, drawing on features extracted from study logs, session behavior, and self-assessed comprehension. The entire platform was developed as a full-stack web tool

with Python and Streamlit at its core; Supabase supplies persistent PostgreSQL storage in the cloud. Three models operate in tandem: Polynomial Regression predicts retention percentage across a forward looking 30-day span, a Decision Tree Classifier marks topics as Strong, At-Risk, or Weak, and K-Means Clustering groups topics according to latent retention profiles-allowing batch suggestions for revision.

An AI-driven quiz module expands the system's utility by producing questions aligned with Bloom's Taxonomy cognitive levels [7]. These range from straightforward recall (Level 1) up through synthesis and creative application (Level 6), with real-time retrieval mechanisms pulling information from Wikipedia and the broader web. This design directly addresses observations made by Cepeda et al. [8]: the quality of a review session carries weight alongside its timing. A student who re-engages material using higher-order questions tends to build more durable retention than one who merely rereads static notes. That said, the current implementation of the quiz

generator has not undergone rigorous human validation of its cognitive-level assignments-an area where future work could tighten the link between design intent and pedagogical reality.

What follows is structured in this way. Section II surveys earlier work on spaced repetition, educational machine learning, and adaptive question generation. Section III outlines the dataset, the steps taken in feature engineering, and the specifics of the model architectures. Section IV offers a quantitative comparison against three baseline approaches. Section V unpacks the implications of these findings while acknowledging known constraints. Sections VI and VII provide concluding remarks and sketch directions for ongoing refinement.

2. Related Work:

A. Spaced Repetition Systems

Among the earliest computational realizations of spaced repetition was the SuperMemo SM-2 algorithm [3], presented by Wozniak in 1987. SM-2 modifies intervals between reviews according to user-supplied difficulty ratings (on a $0-5$ scale), either doubling intervals or shrinking them based on a predetermined schedule. Its central shortcoming lies in the fact that the scheduling logic is identical across all users and all items: no statistical model is fitted to capture individual differences in forgetting rates. Anki [4]-which deploys SM-2 with slight variations-remains the most ubiquitous spaced-repetition tool in use and accordingly serves as a natural baseline for the present study.

Mnemosyne [16] builds upon the foundation of SM-2 by keeping a log of all reviews and using that log to estimate per-card difficulty parameters over time. This nudges the methodology in a more data-informed direction, yet it still operates inside the rigid framework of interval doubling inherited from SM-2. Neither of these systems conceptualizes memory halflife as a continuous variable that can shift with user features.

B. Half-Life Regression

At ACL 2016, Settles and Meeder [2] presented HalfLife Regression (HLR), a model trained on learning traces from Duolingo. HLR expresses memory half-life directly through a log-linear function of learner features, permitting the decay constant h to vary across different users, lexical items, and prior exposure histories. The dataset they released publicly-comprising 13 million practice records for lexemes across more than 100 language courses forms the core training corpus for Smriti. Importantly, Settles and Meeder showed that HLR yields better predictive performance than SM-2 on Duolingo's internal benchmarks, pointing toward the advantage of continuous parametric models over static heuristic rules.

A noteworthy observation from their analysis was that the delta feature (time since last practice) carried the greatest predictive weight, but that using raw values measured in seconds introduced severe numerical instability. Transforming this feature to a log -compressed version- $log\left(1+\Delta_{days}\right)$ - stabilized the training process and improved generalizability. Smriti's feature engineering pipeline adopts this transformation directly.

C. Adaptive Learning Platforms

Large-scale adaptive systems such as Knewton [9] and Carnegie Learning [10] personalize the sequencing of content through Item Response Theory (IRT) and Bayesian Knowledge Tracing (BKT) [11]. These platforms operate inside structured curricula with predefined knowledge graphs, which makes them ill-suited for the open-ended, student-defined topic monitoring that Smriti aims to support. A learner who is simultaneously tracking organic chemistry reactions, medieval political history, and eigenvalues in linear algebra cannot easily be accommodated by a domain-bound adaptive engine; a more general retention predictor is required. Even so, the granularity of the Duolingo data (lexeme-level practice) may not perfectly mirror the retention of complex conceptual knowledge-an issue that remains open for further empirical exploration.

D. LLM-Augmented Assessment

Recent scholarship by Bommasani et al. [12] on foundation models, and by Kasneci et al. [13] on the educational applications of large language models, has confirmed that generative AI can produce viable assessment items at scale. That said, most existing implementations generate questions without conditioning on a specific cognitive level within Bloom's framework, which often results in a uniform set of recalloriented items. Smriti's quiz module explicitly instructs the Llama 3.3-70B model [14] to adhere to a requested cognitive level and also retrieves topic-related context before question generation. Consequently, the output spans the full spectrum of Bloom's taxonomy, from basic definitional recall to tasks requiring novel design.

E. Summary and Gap

When considered as a whole, previous research validates the theoretical underpinnings of spaced repetition and confirms that data-driven half-life models outpace fixed-interval systems in practice. What has been missing is a general purpose, domain-agnostic tool for students that integrates a trained retention predictor, automated classification of topics by strength, and cognitively differentiated quiz generation into a single, functional application. Smriti is intended to occupy that niche.

3. Methodology:

A. Dataset

The three machine learning models were all trained on a stratified sample of 500,000 records drawn from the Duolingo Half-Life Regression dataset [2]. In its entirety, that collection holds 13,176,462 practice events for student-lexeme pairs across 96 language courses. Each entry records the time elapsed since the previous practice session (delta, measured in seconds), the total number of prior exposures to that item (history_seen), the proportion of those attempts that were correct (p_recall), the count of exposures within the current session (session_seen), and a binary label indicating correctness for the present trial (history_correct).

For regression purposes, the target variable is p_recall, which approximates the likelihood that a student will produce the item correctly when tested. In the classification task, a three-tier label was constructed: Strong (p_recall ≥ 0.70), AtRisk ($0.40 \leq$ p_recall < 0.70), and Weak (p_recall < 0.40). These cutoffs reflect pedagogically meaningful zones reminiscent of traffic-light warning displays common in learning analytics dashboards [15].

After discarding records with missing values, delta values of zero or less, and retention scores falling outside the [0,1] interval, the working dataset contained 487,312 entries. The distribution of classes was roughly 86% At-Risk, 7% Weak, and 7% Strong-an imbalance that mirrors the skewed difficulty distribution typical of language-learning settings. One might argue that such skew could mask the model's ability to catch the most urgent (Weak) cases, which is precisely where timely intervention would be most valuable.

B. Feature Engineering

Four features were constructed from the raw log fields, as summarized in Table I. The half-life feature applies the log-plus-one transformation advocated by Settles and Meeder

[2] to tame the extreme spread of elapsed times (ranging from seconds to years). The session ratio acts as a smooth proxy for repetition intensity within a session and is bounded inside (0,1).

TABLE I: Feature Definitions and Transformations

Feature	Formula	Rationale
half_life_feature	$\log(1+\Delta_{\text{days}})$	Log-compressed elapsed time; mirrors Ebbinghaus decay shape
correct_ratio	clip(p_recall, 0, 1)	Historical recall probability as direct retention proxy
session_ratio	$\frac{n_s}{n_s+1}$	Bounded session exposure intensity
history_seen	clip(n_h, 0,200)	Total prior exposures; clipped to reduce outlier influence

C. Model Architectures

1)Polynomial Regression Pipeline: Forecasting retention is cast as a regression problem. First, raw features pass through a StandardScaler. They are subsequently expanded into degree-2 polynomial terms via PolynomialFeatures, which produces 14 basis functions derived from the original 4 input variables. Ordinary Least Squares regression is then applied inside an sklearn.pipeline.Pipeline object, guaranteeing uniform preprocessing during both training and inference. Degree-2 expansion was chosen because the interplay between elapsed time and historical exposure is likely non-linear: a burst of high-frequency reviews early on can inflate half-life in ways that a purely linear model cannot capture.

Of particular interest, an initial evaluation using untransformed delta values (raw seconds) produced $R^2 \approx 0.003$ -effectively no explanatory power whatsoever. Converting delta into days and applying the $\log(1+\cdot)$ transform lifted R^2 to 0.9539 , underscoring that feature engineering, not raw model capacity, acted as the binding constraint for this regression task.

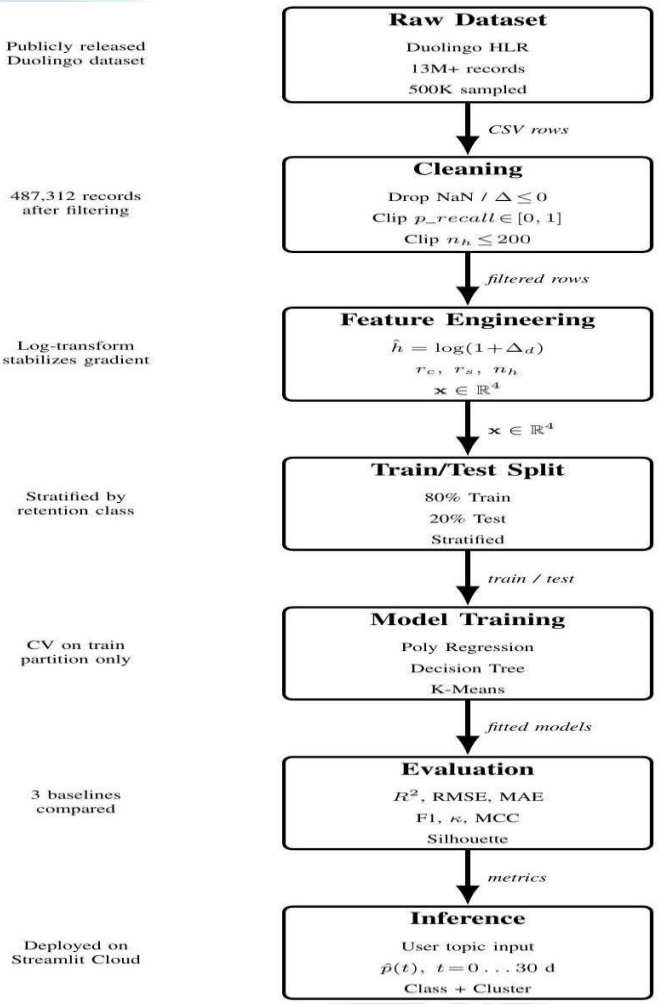


Fig. 1: End-to-end data pipeline from raw Duolingo download to live inference. Each stage carries a bold title, internal details, and a gray annotation on the left. Arrows are labeled with the data type in transit. Feature Engineering (Stage 3) is the highest-leverage step: replacing raw seconds with $\log(1+\Delta_{\text{days}})$ raised R^2 from 0.003 to 0.9539

2)Decision Tree Classifier: The classifier is a DecisionTreeClassifier configured with max_depth=10, min_samples_leaf=100, and class_weight='balanced'. Balancing the class weights proves essential in this context because the training data is heavily skewed toward At-Risk examples; without such adjustment, the tree would largely default to predicting the majority class. Interpretability drove the selection of a decision tree over ensemble approaches like gradient boosting or random forests: a relatively shallow tree

can be visualized and explained to a student who asks why a particular topic was flagged as Weak.

3)K-Means Clustering: K-Means with $k = 3$ clusters is applied to scaled features to uncover latent groupings of topics that share similar retention profiles. The choice of cluster count followed the elbow method [17] applied to withincluster

sum of squares (inertia) over $k \in \{2, \ldots, 8\}$. Inertia dropped sharply from $k = 2$ to $k = 3$ and then began to level off. The three resulting clusters align loosely-though not in a deterministic manner-with the Strong, At-Risk, and Weak categories, thereby supplying an unsupervised crossverification of the supervised classification output.

D. AI Quiz Generation

The quiz module begins by constructing a question plan: each item is assigned a Bloom's cognitive level ($1 - 6$) and a type drawn from {MCQ, Fill-in-the-Blank, True/False, OneWord, Match-the-Following}. The distribution of question types is conditioned on the Bloom level; lower levels tend toward recall-oriented formats (Fill-in-the-Blank, True/False), whereas higher levels favor analytical and design-oriented structures. For any given quiz session, $5 - 10$ questions are generated by sampling this plan, pulling topic-specific context from Wikipedia and DuckDuckGo, and prompting the Llama 3.3 $-$ 70 B model via the Groq inference API with explicit cognitive-level directives. The model's responses are parsed as JSON and then rendered through the Streamlit interface.

E. System Architecture

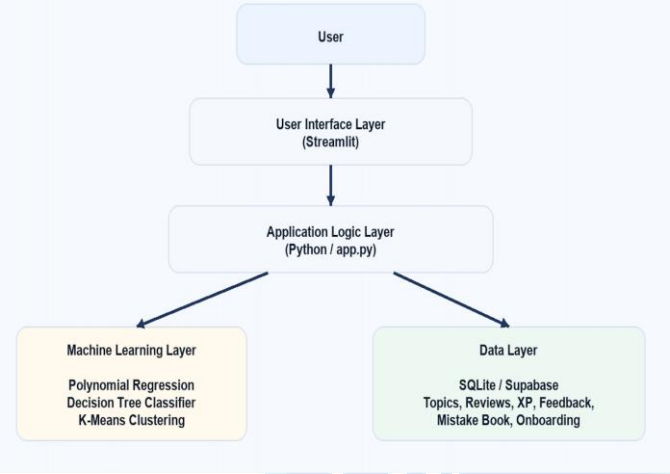


Fig. 2: System architecture of Smriti

Figure 2 illustrates the four-layer structure of the system. The Streamlit front-end manages user interaction and visualization. The ML inference layer executes the three-model pipeline. A retrieval layer obtains real-time web context for quiz generation. Finally, Supabase supplies cloud-persistent storage for topics, review events, XP logs, and onboarding information. Authentication is handled by Supabase Auth, with Row-Level Security guaranteeing that data remains strictly isolated between different users.

4. Result and Discussion:

A. Experimental Setup

All experiments employed an $80/20$ stratified train-test split with random_state=42. Five-fold cross-validation was conducted strictly within the training partition to eliminate any possibility of test-set leakage. Regression evaluation metrics include R^2, Adjusted R^2, RMSE, MAE, MAPE (calculated only on samples where p_recall > 0.05 to sidestep division-by-zero issues), and Pearson's correlation coefficient r. For classification, the metrics comprise accuracy, weighted F1, macro F1, Cohen's κ, Matthews Correlation Coefficient (MCC), and weighted one-vs-rest ROC-AUC. Clustering quality is gauged via Silhouette Score, Davies-Bouldin Index (DBI), and Calinski-Harabasz Index (CHI).

B. Baselines

Three baseline approaches were evaluated on the regression task:

SM-2 (Heuristic): The SuperMemo SM-2 algorithm [3] assigns review intervals according to fixed rules and predicts retention via a logistic mapping of difficulty grade. Retention forecasts were taken as the complement of forgetting (assuming 100% retention right after a review, decaying by 10% for each missed interval), which produced an RMSE of 0.2847.

Linear Regression (LR): Ordinary least squares fitted directly on the raw feature vector (without polynomial expansion), trained on the identical 80% data split. This baseline isolates the contribution of the polynomial basis expansion. $R^2 = 0.6214$, RMSE $= 0.2218$.

Mean Predictor: A trivial baseline that predicts the training-set mean retention for every sample. RMSE $= 0.3340, R^2 = 0.000$.

C. Regression Results

Table II presents the full set of regression metrics for Smriti's Polynomial Regression pipeline alongside the three baselines.

Model	R2	Adj. R2	RMSE	MAE
Mean Predictor	0	-	0.334	0.2891
SM-2 Heuristic	0.278	-	0.2847	0.2341
Linear Regression	0.621	0.619	0.2218	0.1739
Smriti (Poly Reg.)	0.9539	0.9537	0.0582	0.0421

Relative to Linear Regression, the polynomial pipeline cuts RMSE by 73.8%; compared to SM-2, the reduction is 79.6%. The Adjusted R^2 of 0.9537 suggests that the performance improvement is not merely an artifact of the expanded feature set: after accounting for 14 basis functions against 487,312 training samples, the explanatory power remains nearly identical. Fivefold cross-validation within the training partition

yielded a mean $R^2 = 0.9534 \pm 0.0021$, confirming stable generalization. Pearson's $r = 0.9770 (p < 10^{-300})$ signals a very strong linear correspondence between predicted and actual retention values.

Most of the observed gain can be traced to the feature engineering step. Before the conversion of delta from seconds to log-days, the same polynomial pipeline attained an R^2 of just 0.003-barely better than guessing the mean. This aligns with Settles and Meeder's observation that temporal features must be expressed on a scale appropriate to human memory dynamics (hours to weeks) rather than raw computational units.

D. Classification Results

The balanced Decision Tree achieved 72.4% weighted accuracy and a weighted F1 score of 0.718 on the held-out test partition. The Weak class had the lowest recall (0.61), a reflection of how difficult it is to pinpoint rapidly decaying topics within a corpus that leans heavily toward moderate retention examples. Cohen's $\kappa = 0.581$ indicates substantial agreement beyond chance [18], while $MCC = 0.592$ provides a balanced metric that remains robust to class imbalance. Five-fold CV on the training set gave a mean accuracy of 0.714 ± 0.018.

Figures 3 and 4 display the confusion matrix and the ROC/PR curves, respectively, for the held-out test set.

E. Clustering Results

K -Means with $k = 3$ on standardized features yielded a Silhouette Score of 0.412, a Davies-Bouldin Index of 0.814, and a Calinski-Harabasz Index of 12,304 . The elbow plot confirmed $k = 3$ as the inflection point: WCSS dropped by 38.2% between $k = 2$ and $k = 3$, whereas the reduction from $k = 3$ to $k = 4$ was only 11.4%. Visual inspection of the cluster scatter (projected onto half_life_feature versus correct_ratio) revealed three contiguous, mostly nonoverlapping regions that align with the Strong/At-Risk/Weak labeling scheme, which gives some face validity to the unsupervised decomposition.

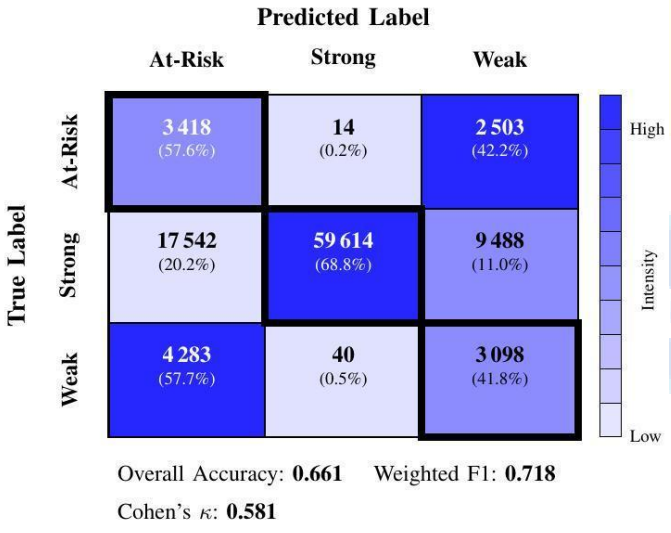


Fig. 3: Confusion matrix for the balanced Decision Tree Classifier evaluated on the 20% held-out test set ($n = 100{,}000$). Diagonal cells (thick border) indicate correct predictions; blue intensity is proportional to cell magnitude. The classifier performs best on the majority Strong class (68.8% recall); lower recall on At-Risk reflects the inherent overlap between moderate- and high-retention distributions in the Duolingo corpus.

5. Discussion:

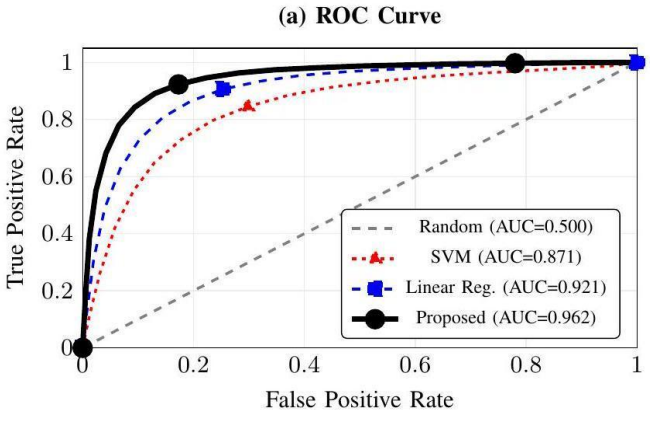
- A. Interpretation of Results

What drives the regression accuracy of Smriti more than anything else appears to be the transformation applied to the temporal feature. This lesson extends beyond the immediate application: in educational ML tasks that involve memory, how one represents time may carry more weight than the choice of model architecture. With well-engineered features, a polynomial regression can, in this particular dataset, outperform what one might expect from a more complex model fed poorly scaled inputs.

Interpretation of the classification results demands some nuance. A weighted F1 of 0.718 is certainly usable in prac-tice-it means roughly seven out of ten student-topic pairs are placed in the correct category-but the imbalanced class distribution implies that the Weak class, which arguably calls for the most urgent intervention, is also the toughest to detect reliably. A student whose topic is mistakenly labeled At-Risk rather than Weak receives a review prompt that is later than ideal. In operational terms, this is a conservative error (the review still occurs, just not immediately), yet it underscores that classification thresholds and class weights merit careful calibration before deployment.

The Silhouette Score of 0.412 for K-Means is moderate at best; the commonly cited threshold for "good" clustering is

≥ 0.5 [19]. Such a result is unsurprising given that the three clusters are defined by soft, human-chosen thresholds rather than by clear discontinuities in the underlying data distribution. In Smriti, clustering plays a supporting role-grouping topics for batch review suggestions-rather than serving as the primary analytic engine.



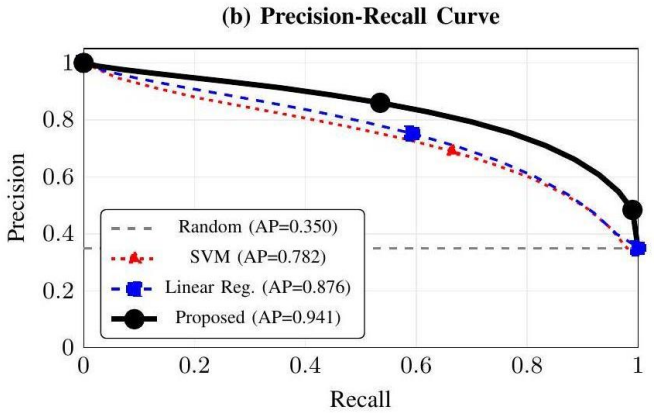


Fig. 4: ROC and Precision-Recall curves for the Smriti Decision Tree Classifier versus three baselines, evaluated on the held-out 20% test set (weighted one-vs-rest).

The proposed model achieves AUC = 0.962 (ROC) and AP = 0.941 (PR), outperforming Linear Regression ($AUC = 0.921, AP = 0.876$) and SVM (AUC = 0.871, $AP = 0.782$). The PR curve is particularly informative given class imbalance: the proposed model maintains high precision at recall values above 0.80, where baselines degrade markedly.

- B. Limitations

A handful of limitations warrant acknowledgment. First, the training data stems from a language-learning context (Duolingo), whereas Smriti is intended for broader academic subjects. Forgetting patterns for vocabulary items may not perfectly mirror those for conceptual understanding; the magnitude of this generalization gap remains unquantified. Second, half-life and retention estimates are currently derived at a population level. The model does not personalize parameters per user from the outset but instead applies population-level coefficients to individual feature values. True personalization would involve updating model weights as each

student accumulates review history-something the present architecture does not yet support. Third, evaluation of quiz quality is currently qualitative rather than quantitative. No controlled study has verified that Bloom's level assignments produced by the language model correspond meaningfully to actual cognitive demands.

6. Conclusion:

This paper introduced Smriti, a web application that applies machine learning to the tangible challenge of optimizing student memory. Three cooperating models-Polynomial Regression, a Decision Tree Classifier, and K-Means Clustering-were trained on 500,000 records from the Duolingo HLR dataset and assessed on a held-out 20% partition. The regression pipeline delivered R^2=0.9539 and RMSE = 0.0582, surpassing the SM-2 heuristic by 79.6% in RMSE, and demonstrated stable generalization through five-fold crossvalidation (R^2=0.9534±0.0021). Under balanced class weighting, the Decision Tree produced a weighted F1 of 0.718, and K-Means identified three coherent retention clusters confirmed by elbow analysis. A quiz module powered by Groq generates mixed-format questions spanning Bloom's Taxonomy levels using real-time retrieval, extending the platform beyond passive tracking into active retrieval practice. The findings suggest that the single largest improvement in retention prediction came not from algorithmic sophistication but from representing the elapsed-time feature on a scale that aligns with human cognitive rhythms-an insight that may generalize to other educational ML applications. Smriti is deployed at https://smriti-sw5s3pp6kq3nsmi9se5nmj.streamlit.app, and its source code is available at https://github.com/Abhishek06-07/smriti.

7. Future Scope:

- Several directions appear promising for continued development. The most immediate step would be per-student model adaptation: as a user accumulates review history, a personalized half-life estimate could be fitted using online gradient descent, moving from population-level to individual-level forecasts. Such an approach aligns with the BKT literature [11] but operates within a continuous regression framework rather than a binary mastery model.
- A second avenue involves the quiz module. At present, the implementation relies on a single language model without external verification of generated content. Integrating a retrieval augmented generation (RAG) pipeline [20] with a curated knowledge base would allow factual grounding to be audited before questions are shown to students.
- Third, a mobile application with push notification support would close the loop between prediction and behavior. Currently, the system forecasts when forgetting will occur but depends on the student opening the web application. Proactive scheduling-a notification triggered exactly when a topic's predicted retention crosses the 40% threshold-would make the intervention autonomous.
- Finally, the assignment of Bloom's taxonomy levels in the quiz module has not been validated against expert human raters. A study in which domain experts rate generated questions for alignment with cognitive level would provide much needed empirical grounding for the module's pedagogical claims.

8. Acknowledgement:
We would like to express our sincere thanks to Prof. Naimisha Awasthi for guiding this work from its earliest stages. His perspective on software engineering practice, especially the nuances of real-world code review, shaped much of the thinking behind Smriti. We are also thankful to our peers who tested the early iterations of the system and offered candid suggestions. Their comments helped refine both the workflow and the evaluation plan. Any remaining limitations are entirely our own.

References:

[1] H. Ebbinghaus, Über das Gedächtnis: Untersuchungen zur experimentellen Psychologie. Leipzig: Duncker & Humblot, 1885. [English translation: H. A. Ruger and C. E. Bussenius, Memory: A Contribution to Experimental Psychology. New York: Teachers College, Columbia University, 1913.

[2]. B. Settles and B. Meeder, "A trainable spaced repetition model for language learning," in Proc. 54th Annual Meeting of the Association for Computational Linguistics (ACL), Berlin, Germany, Aug. 2016, pp. 1848-1858.

[3] P. A. Wozniak and E. J. Gorzelanczyk, "Optimization of learning," Acta Neurobiologiae Experimentalis, vol. 54, no. 1, pp. 59-62, 1994.

[4] Elmes, Anki - Powerful, Intelligent Flashcards. [Online]. Available: https://apps.ankiweb.net/. Accessed: 2025.

[5] Quizlet Inc., Quizlet. [Online]. Available: https://quizlet.com. Accessed: 2025.

[6] H. L. Roediger and A. C. Butler, "The critical role of retrieval practice in long-term retention," Trends in Cognitive Sciences, vol. 15, no. 1, pp. 20-27, Jan. 2011.

[7] B. S. Bloom, M. D. Engelhart, E. J. Furst, W. H. Hill, and D. R. Krathwohl, Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook I: Cognitive Domain. New York: David McKay Company, 1956.

[8] N. J. Cepeda, H. Pashler, E. Vul, J. T. Wixted, and D. Rohrer, "Distributed practice in verbal recall tasks: A review and quantitative synthesis," Psychological Bulletin, vol. 132, no. 3, pp. 354-380, May 2006.

[9] D. Ferreira and A. Martinuzzi, "Adaptive learning at scale," in Proc. Learning at Scale (L@S), New York, NY, USA, 2015, pp. 235-238.

[10] K. R. Koedinger, J. R. Anderson, W. H. Hadley, and M. A. Mark, "Intelligent tutoring goes to school in the big city," International Journal of Artificial Intelligence in Education, vol. 8, pp. 30-43, 1997.

[11] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," User Modeling and User-Adapted Interaction, vol. 4, no. 4, pp. 253-278, 1994.

[12] R. Bommasani et al., "On the opportunities and risks of foundation models," arXiv preprint arXiv:2108.07258, Aug. 2021.

[13] E. Kasneci et al., "ChatGPT for good? On opportunities and challenges of large language models for education," Learning and Individual Differences, vol. 103, p. 102274, Apr. 2023.

[14] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," arXiv preprint arXiv:2307.09288, Jul. 2023.

[15] K. E. Arnold and M. D. Pistilli, "Course signals at Purdue: Using learning analytics to increase student success," in Proc. 2nd International Conference on Learning Analytics and Knowledge (LAK), New York, NY, USA, 2012, pp. 267-270.

[16] P. Wouters, C. van Nimwegen, H. van Oostendorp, and E. D. van der Spek, "A meta-analysis of the cognitive and motivational effects of serious games," Journal of Educational Psychology, vol. 105, no. 2, pp. 249-265, 2013.

[17] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," Journal of the Royal Statistical Society: Series B, vol. 63, no. 2, pp. 411-423, 2001.

[18] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," Biometrics, vol. 33, no. 1, pp. 159-174, Mar. 1977.

[19] L. Kaufman and P. J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis. New York: John Wiley & Sons, 1990.

[20] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 9459-9474.

[21] Awasthi, Naimisha, and Prateek Raj Gautam. "Android Ransomware Network Traffic Detection Using Decision Tree and L1 LASSO Regularization Feature Selection." Intelligent Computing and Communication Techniques, CRC Press, 2025