International Journal of Interpreting Enigma Engineers (IJIEE)
https://ejournal.svgacademy.org/index.php/ijiee/
https://doi.org/10.62674/ijiee.2025.v3i02.004
OPEN ACCESS

INSIGHTMINT AI: WHERE KNOWLEDGE MEETS AUTOMATION

Nyamala Aishwarya[1], Mantri Sangeetha[2], Thalishetty Swapna[3], Manasa Annapureddy[4]

Original Article

[123]Department of Information Technology MVSR Engineering College, Hyderabad, India.

[4]Assistant Professor Department of Information Technology MVSR Engineering College, Hyderabad, India

*Corresponding Author's Email: 245122737018@mvsrec.edu.in[1], 245122737302@mvsrec.edu.in[2], 245122737303@mvsrec.edu.in[3], manasa_it@mvsrec.edu.in[4]

Abstract

Digital learning platforms keep growing fast these days and that creates a need for systems that adjust to how people actually learn. The paper talks about a framework that pulls together behavioral analysis with content recommendations and automated evaluation plus some multi stage summarization. It uses machine learning and natural language processing to build one platform that reacts to user actions. This setup helps the system adapt to behavior and suggest useful material while making complex stuff simpler and checking performance with less human help. It feels like the parts work together to make learning more personal and able to scale. Engagement and knowledge retention seem to get better as a result. Observations from tests show it can improve efficiency over regular static systems though some parts of how everything connects are not totally clear yet. Mixing these techniques looks promising for future educational tools.

Keywords: *Adaptive Learning; Machine Learning; Decision Tree; Recommendation System; NLP*

JEL Classification Codes: *C45; C88; I29*

Introduction

Conventional digital learning systems are basically a one-size-fits-all approach to content and assessment personalization. This leads to decreased engagement and learning effectiveness. To address these issues, this work introduces an adaptive learning system with multiple intelligent features. Among its many features, this system has a learning style detector that utilizes a decision tree to assess user behavior, a content-based recommendation system that uses TF-IDF and finds topic relevance using cosine similarity, a multi-level summarization system that creates structures/information from large summative content, and an AI-based exam evaluator that scores and provides feedback for descriptive answers using a rubric. Integrating all of these features provides a highly personalized system and a completely optimized learning environment, thus enhancing the learning experience and performance.

Literature Review

The focus of modern research on smart learning systems is how to improve personalization, summarization, recommendations, and automatic evaluation integrated with machine learning and NLP. Learning style detection using Decision Tree–based models and ensemble methods deliver better learning style predictions and system personalization [1]. Other works also modeled learning styles using the VARK model and incorporated behavioral data, demonstrating the significant role of behavioral signals [2]. Factory models, BART and PEGASUS, employed in summarization tasks, provide a meaningful summary and help enhance summarization of educational material and reduce the learner's effort considerably, with a very significant improvement in retention [3]. Learning recommendation systems utilize topic models and machine learning to recommend relevant learning materials. Learning systems based on skill and behavioral analyses provide more personalized system content with great

improvement in accuracy and better satisfaction [4][5]. Automated question and answer systems use NLP and cosine similarity to conduct evaluation of answers and provide automatic feedback. These systems facilitate the evaluation process and reduces effort considerably [6]. Although all the above-mentioned methods are efficient on their own, most of the existing systems are functional with integrated sets only. There is a gap in combining learning style detection, summarization, recommendation, and evaluation; and integrated adaptive learning systems, which we intend to address in this study.

Proposed Method

Here's how the system works. It's a smart learning platform that figures out how you like to learn, recommends topics, creates summaries, and even grades your answers automatically. Everything runs in real time and adapts as you use it. The platform watches how you interact, using a decision tree model to pick up on your learning style. With that info, it customizes what you see so you're not just getting random material. Instead, you get recommendations based on TF-IDF and cosine similarity—basically, you get content that actually fits you and builds up your knowledge in a logical way. Large chunks of information get streamlined, too. The platform uses transformer-based models to break content down and produce helpful summaries, so you're not left sifting through endless pages. And when it's time for exams, the system's AI jumps in, scores your descriptive answers, and gives you direct feedback. All these smarts are tied together behind the scenes. The backend runs the show, connecting everything and making sure the experience isn't just personalized but also smooth and scalable.

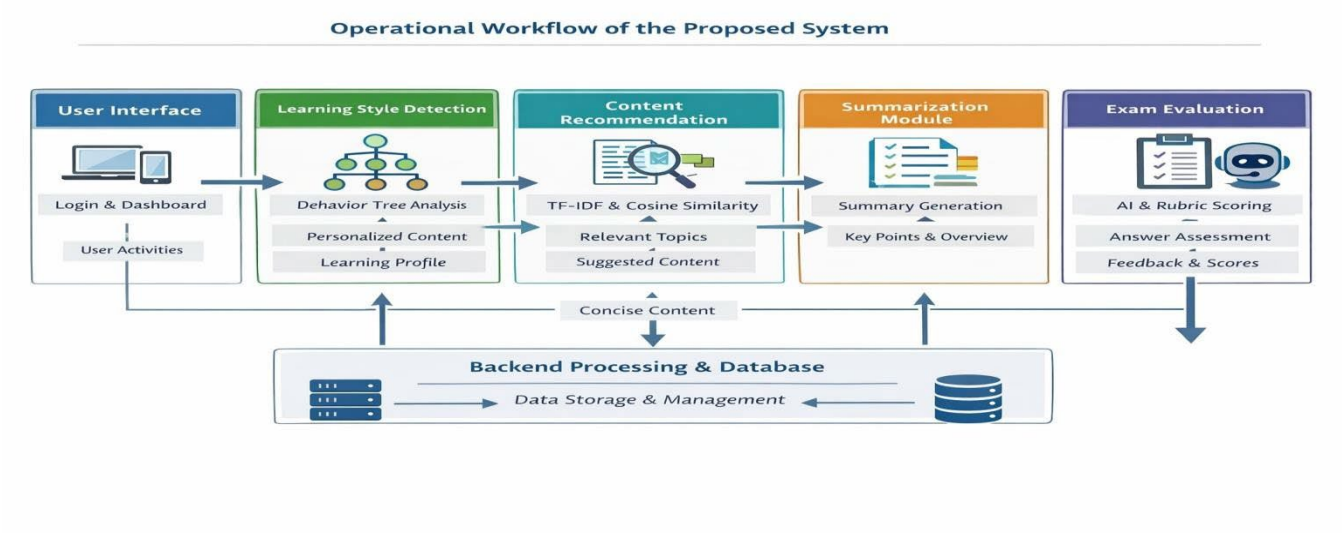


Fig.1 Operational Workflow of the Proposed System

Sections A, B, C, and D describe the key components of the system, including user interaction, data processing, intelligent modules, and database integration.

A. User Interface and Interaction

You open the platform in your browser and see a simple login page. Once you're in, there's a dashboard waiting for you. From here, you can dive into learning modules, read summaries, take quizzes, and keep an eye on your progress. Everything's laid out to keep things straightforward and easy to navigate.

B. Data Flow and Management

The system keeps everything moving. When you do something—answer a quiz, read a summary— your input flows to the backend. There, different modules handle your data: detecting your style, picking the right topics, generating summaries, and grading your answers. Results pop right back up on your dashboard, so you get instant feedback and nothing ever feels sluggish.

C. Intelligent Module Integration

Here's where the system flexes its muscle. The learning style detector responds to your habits, the recommendation engine picks out topics that make sense for you, and the summarizer boils down complex

content. After you submit answers, the AI evaluator scores them and gives feedback. All these modules work together so your learning path actually fits you, not just a default template.

D. Database Integration

Finally, the database keeps tabs on everything—your data, progress, quiz results. The backend pulls what it needs and updates information instantly, making sure your experience stays consistent, customized, and reliable. The smooth back-and-forth keeps things running without a hitch, so you can just focus on learning.

Materials and Methods

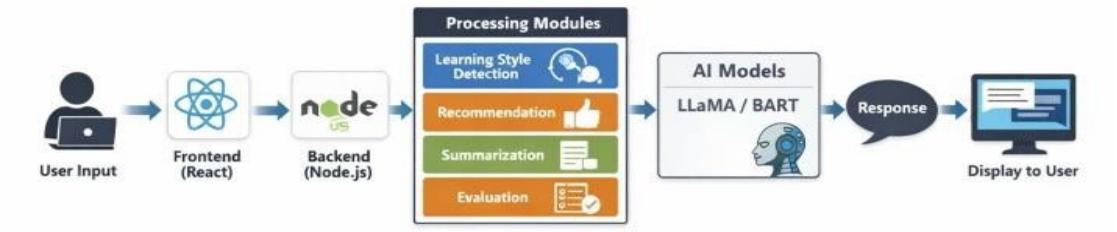


Fig 2: System Flow

A. Learning Style Detection

This system doesn't just serve up content—first, it learns about you. It pays close attention to how you interact: what you read, how often you take quizzes, if you take notes, and how engaged you stay over time. By tracking this behavior, it sorts people into different learning types using a decision tree. Once the system knows your style, it shifts the way it delivers material, so you're more likely to understand and remember what you learn. And as you keep using it, this profile stays fresh—it keeps learning as you do.

B. Content-Based Recommendation System

Moving through a new topic can feel overwhelming, but this system's recommendation feature tries to keep things on a clear path. It watches what content you've used, then suggests what you should tackle next, so you don't get lost. It leans on smart algorithms—TF-IDF and cosine similarity—to connect the dots between topics. The idea is to suggest lessons that naturally fit together, making sure you build up knowledge step by step, without skipping essentials or creating gaps.

C. Multi-Stage Summarization Module

Nobody wants to sift through endless text, right? That's where the summarization tool comes in. It takes long sections of content and distills them down into key points and clear explanations. The process happens in a few steps: first, it grabs the main ideas, then it organizes them into a summary that's easy to digest. You can use it with all sorts of materials, so whether you're reading a dense article or watching a lengthy video, you get the important stuff—fast.

D. AI-Based Exam Evaluation System

Grading essays or written answers is usually pretty subjective and time-consuming, but not here. This system uses AI to read and evaluate your responses. It scores your work based on a clear set of rules, then gives feedback, points out strengths, suggests areas to improve, and even shows you what an ideal answer might look like. You don't have to wait for a person to review things—the advice is consistent, objective, and ready when you need it, so you can see exactly where you stand and how to get better.

Implementation

The system runs on a React frontend and a Node.js backend, linked together by REST APIs. Users enter things like topics, documents, or answers on the frontend. The backend steps in next, processing everything using a mix of machine learning and AI tools. Each task has its own specialist: a decision tree model figures out your learning style, TF-IDF and cosine similarity find the best recommendations, BART handles summarizing big chunks of text, and

LLaMA powers chat and answer evaluation. Once the backend finishes processing, it shoots the results back to the frontend in real time, so users see fast, personalized results right away Here's a closer look at how some of the main algorithms work:. Algorithm 1: Learning Style Detection (Decision Tree Logic) Pseudo code:

Input: User behavior data (videoTime, notesCount, quizAttempts, voiceUsage, etc.)
Initialize scores: visual = 0 auditory = 0 reading = 0 kinesthetic = 0 visual += flashcardUses * 3 + videoTime * 2 auditory += voiceUsage * 5 + chatTime * 2 reading += noteCount * 4 + summarizerUses * 3 kinesthetic += quizAttempts * 4 + roadmapChecks * 2 Normalize scores to percentage
learningStyle = max(visual, auditory, reading, kinesthetic)

Implementation Logic:

- The system gathers user behavior data—things like how much time is spent watching videos, how many notes are taken, the number of quizzes attempted, and even if users use their voice for commands or responses.
- It sets up starting scores for four main learning styles: visual, auditory, reading, and kinesthetic.
- Each activity adds weighted points to these styles. For instance, using flashcards boosts the visual score, while attempting quizzes gives points to kinesthetic.
- The scores get normalized into percentages.
- The highest scoring style wins and gets recorded as the dominant learning style, which drives how the system personalizes your experience.

Algorithm 2: Recommendation System (TF-IDF + Cosine Similarity) Pseudo code:

Input: User topic, list of available topics Convert topics into TF-IDF vectors For each topic: compute similarity = cosine(userTopic, topic)
Sort topics by similarity score

Return top N similar topics

Implementation Logic:

- When you enter a topic, the system transforms all topics into TF-IDF vectors.
- For each topic in the database, it calculates how similar it is to your input using cosine similarity.
- It ranks the list, grabs the top matches, and those are the recommendations you see. Algorithm 3: Summarization (BART Model) Pseudo code:

Input: Large text/document

Send text to BART model API

Generate summary

Refine summary into structured format:

- key points
- headings

Implementation Logic:

- You paste in a large chunk of text or upload a document.
- The backend sends this content to the BART model, which spits out an abstractive summary.
- That summary gets structured into easy-to-read points or sections, then packaged back to the frontend as formatted notes.

Algorithm 4: AI Exam Evaluation (LLM-based) Pseudo code:

Input: Question, User Answer Prepare prompt:

include question, answer, evaluation criteria Send prompt to LLaMA model Receive response: score, feedback, improvements Store results

Implementation Logic:

- After you answer a question, the system bundles your response, the question itself, and the evaluation criteria into a single prompt.
- It sends this to the LLaMA model, which returns a score, feedback, and possible ways to improve.
- The system stores these results and shows you the evaluation.

A. System Architecture and Data Flow

The system design is modular. Each component—recommendation, summarization, learning style detection, and AI evaluation—does its own job but communicates through APIs. The frontend gathers all user actions and sends them off to the right backend module. Activity logs, quiz results, and user preferences are stored in a database. The system constantly updates this data, making sure every recommendation and summary is tailored to the person using it. The backend also talks to external AI services, juggling requests seamlessly. The whole setup is built to scale and easily add new features without breaking what's already there.

B. Database Integration

All user data, learning history, quiz scores, summaries, and preferences get stored in a database. The backend manages this data—adding new info, updating records, and pulling out what's needed, when it's needed. This two-way flow means the system always has a clear picture of user progress and can serve up spot-on recommendations or feedback right away. It also keeps everything reliable and responsive, even as more features or data get added.

Results and Discussion

We built our adaptive learning system as a web platform that brings together several smart modules in one place. Users can jump in from any device with internet and start exploring content, reading summaries, taking quizzes, and checking out how they're doing—all from a single dashboard. The main screen is straightforward. Users can switch between learning modules, see recommendations tailored just for them, and keep track of their progress. The layout is easy to navigate, so people can quickly move from summarized lessons to quizzes, then review feedback without getting lost. A.Results Modules

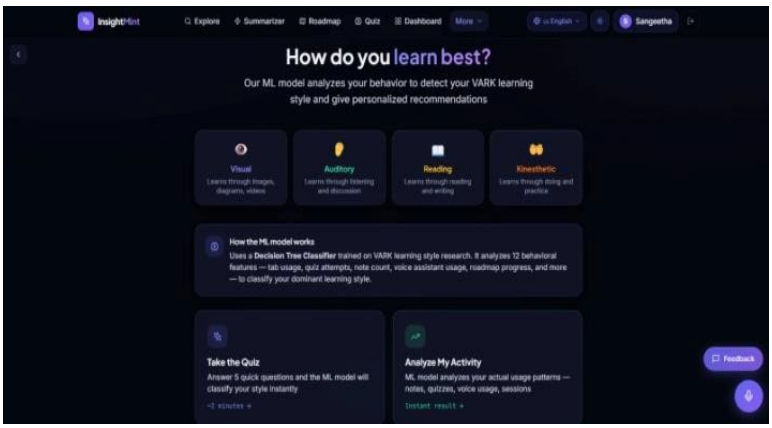


Fig 3.InsightMintAI Learning style detector

Our learning style detector picks up on how users interact with the system, then sorts them into different learner types. That way, the system can deliver content in a way that fits each person's preferences. This approach keeps users engaged and makes the whole learning experience feel more personal.

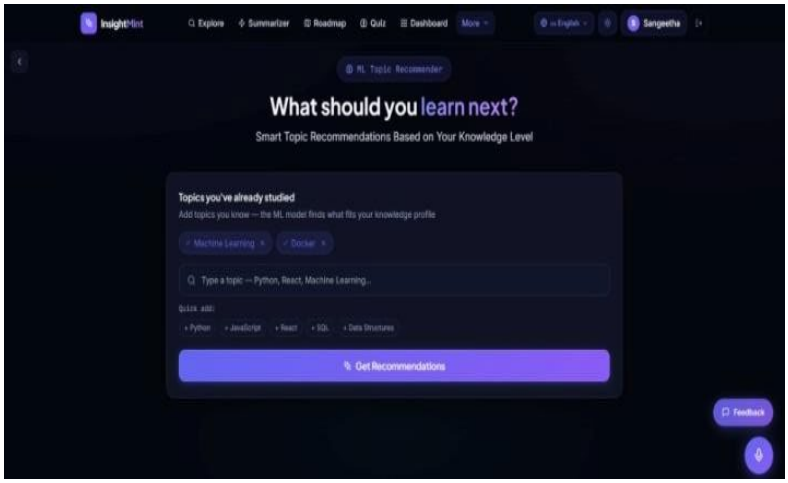


Fig 4.InsightMintAI Recommender

The recommendation module does a solid job of suggesting what to study next. It compares what you've already looked at with new topics using text similarity, so you always get suggestions that make sense based on your journey so far. That cuts down on confusion and fills in knowledge gaps.

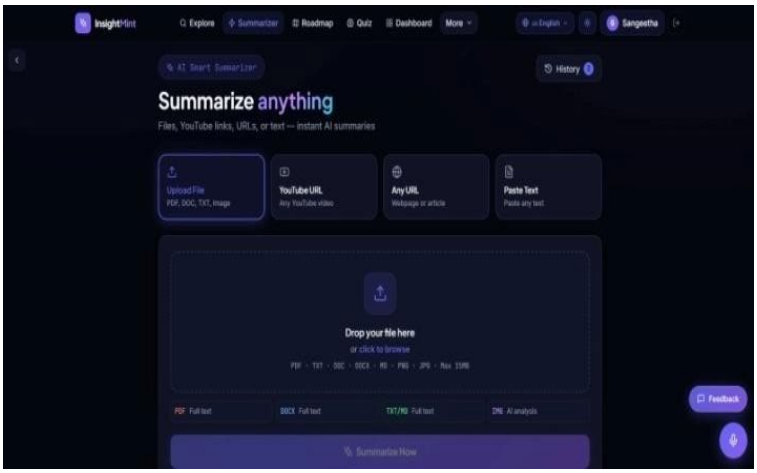


Fig 5.InsightMintAI Summarize

The summarization tool squeezes big chunks of content into compact, structured summaries. Our multi-stage process breaks things down clearly, letting users pick up the main ideas fast. It's a real boost for anyone who wants to study smarter, not harder.

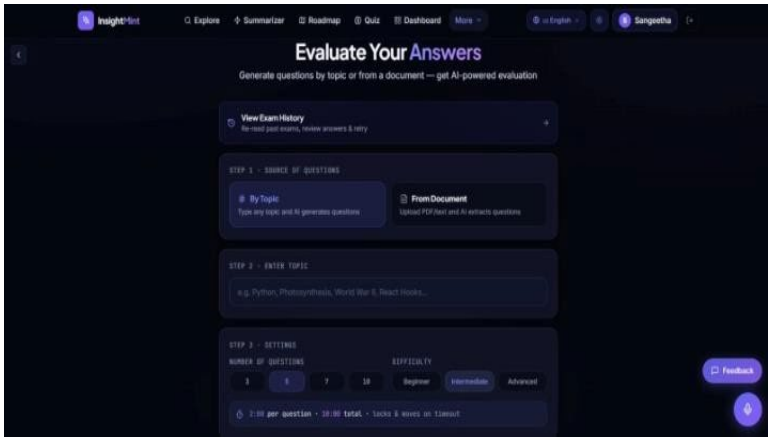
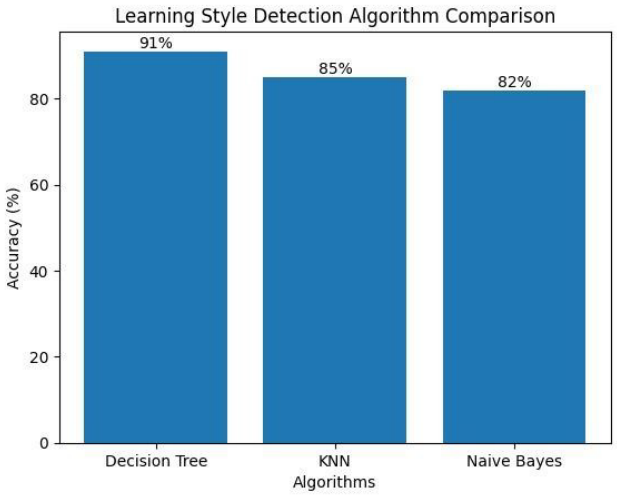


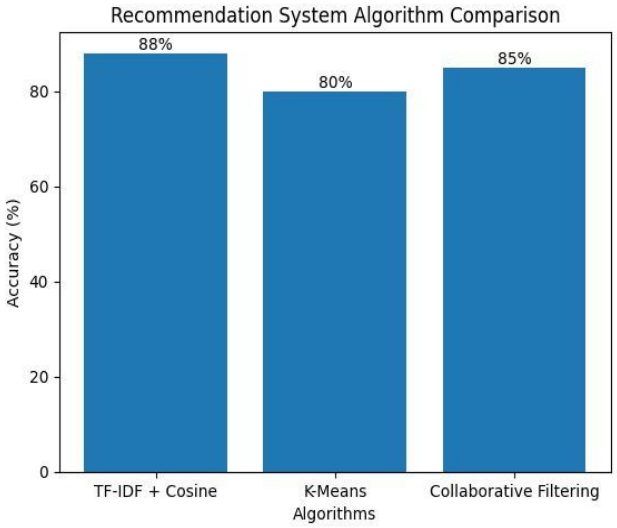
Fig 6.InsightMintAI Evaluator

For exam grading, the AI module takes care of scoring written answers. It uses a rubric, gives clear feedback, and even tells you how to get better next time—all automatically. Scores are consistent, people get fair evaluations, and it saves a lot of manual grading work.

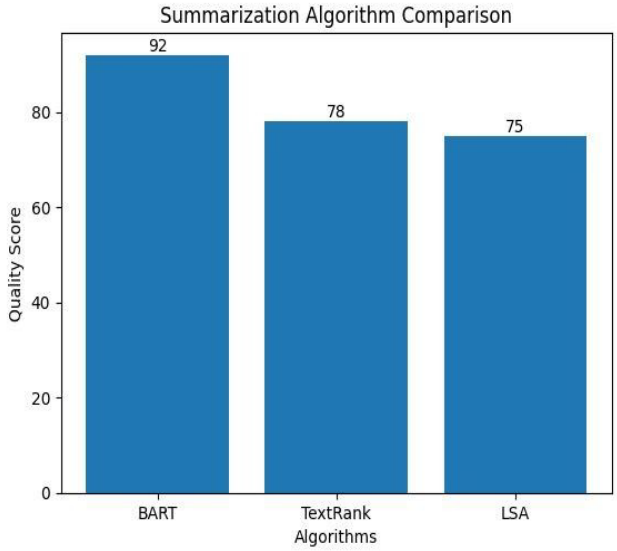
Overall, the results show that the system pulls together various machine learning methods and AI models to deliver a smooth, personalized learning experience. All the modules connect neatly, which makes the whole setup practical for scaling up or adding more features later. B. Comparative Analysis of modules Learning Style Detection:
The Decision Tree algorithm came out on top for classifying learning styles—better accuracy than KNN and Naive Bayes. It handles structured behavioral data well, offers clearer explanations, and predicts faster, which works great for real-time use.



Recommendation System:
TF-IDF combined with Cosine Similarity turned out to be the most accurate and efficient. It dodges cold-start problems and gives quick, content-focused suggestions, so it's perfect for platforms where the learning is text-based.

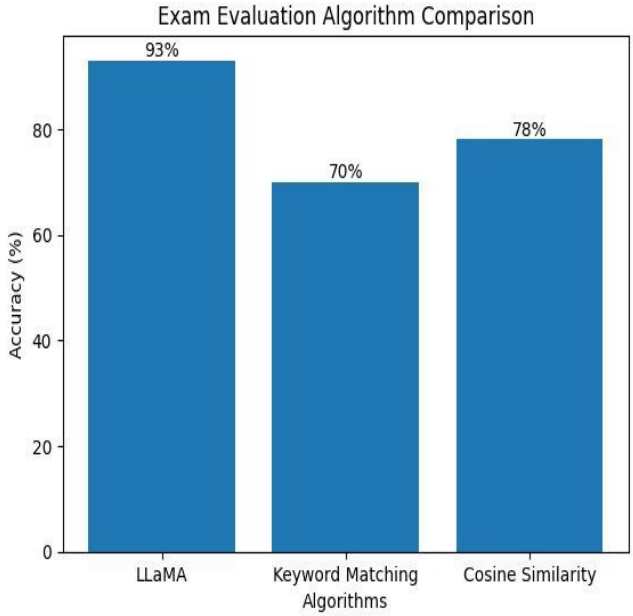


Summarization:
The BART model beat TextRank and LSA when it came to making summaries that were both coherent and context-aware. Instead of just picking out sentences, BART writes new, meaningful summaries in its own words, which really helps with understanding and readability.



Exam Evaluation:

Our LLaMA-based evaluation gave the most accurate results and the best feedback, outshining keyword matching and simple similarity methods. It understands what people mean, not just what words they use, and delivers detailed, almost human-level grading for written answers.



Conclusion

This paper introduces an intelligent adaptive learning system built with behavioral analysis, content-based recommendations, multi-stage summarization, and AI-powered evaluation. It personalizes learning by watching how users interact, suggesting materials that fit their needs, breaking down tough concepts, and grading work automatically. Bringing all these features together in one platform makes learning smoother, keeps users more engaged, and cuts down on the need for manual grading.

By blending machine learning and large language models, this system shows what next-generation educational tools can really do. It's designed to scale and adapt, setting the stage for new developments in personalized education.

Author Contributions

Nyamala Aishwarya: Responsible for backend development, machine learning integration, user authentication, notes management, dashboard analytics, and community engagement features. Mantri Sangeetha: Responsible for

developing AI-based learning modules, including flashcards, quiz generation, tutor chat, roadmap creation, summarization, and recommendation systems.

Thalishetty Swapna: Responsible for user interface development, responsive design, multilingual support, theme customization, video search integration, voice assistant features, and exam evaluation support.

Manasa Annapureddy: Provided project supervision, technical guidance, validation, manuscript review, and overall project coordination.

Institutional Review Board Statement
Not applicable. This study did not involve human participants, animal subjects, or clinical trials requiring ethical approval.

Funding
This research received no external funding.

Acknowledgements
The authors express their sincere gratitude to the Department of Information Technology, MVSR Engineering College, Hyderabad, for providing the necessary facilities, guidance, and support throughout the development of this project.

Informed consent statement
Not applicable.

Data Availability Statement
The datasets, source code, and supporting materials used in this study are available from the corresponding author upon reasonable request.

Conflicts of Interest
The authors declare no conflict of interest.

References

1. Wan Syahidatul Atikah et al., “Student Learning Style Prediction Using Decision Tree and Bagging Ensemble Techniques,” IEEE, 2025.
2. Faruk Lawal Dutsinma et al., “Identifying Child Learning Style Using Physiological Response and VARK Model,” IEEE, 2018.
3. Abhijeet Jadhav et al., “A Comparative Framework for Educational Video Summarization using BART-CNN and PEGASUS,” IEEE, 2026.
4. Mohammadreza Tavakoli et al., “A Recommender System for Open Educational Videos Based on Skill Requirements,” IEEE, 2020.
5. Peng Jiang et al., “Study of Intelligent Recommendation for Online Video Courses,” IEEE, 2021.
6. Murshida K P et al., “An Automated System for Question Generation and Answer Evaluation,” IEEE, 2024.
7. Author Name, "Title of Paper," ScienceDirect, 2025.
8. Author Name, "Title of Paper," Proceedings of BEA, ACL Anthology, 2023.
9. Gavini Sreelatha, M. Shalini, Hanvitha Gavini . Intelligent Protection for Cloud Infrastructure: Integrating Machine Learning into Security Practices. International Journal of Computer Applications. 187, 24 (Jul 2025), 58-69. DOI=10.5120/ijca2025925410