AI-Based Personalized Learning System for Skill Development

Mohammed Poonawala[1], Gargi Nakil[2], Abhilash Dash[3], Jeet Bhalerao[4], and Dr. Suvarna Pawar[5]
[1,2,3,4] *Student, MIT ADT University, Pune*
[5]*Professor and Program Head, MIT ADT University, Pune*

***Abstract*—The diversity in individual learning preferences has long posed a challenge in the field of education. Conventional study techniques often fail to accommodate the varying modalities through which students absorb information most effectively. This paper presents the design and methodology of an AI-based learning system that identifies a user's ideal learning modality using the VARK framework—Visual, Auditory, Reading/Writing, and Kinaesthetic. By tracking user performance and adapting learning delivery through machine learning models like K-Means clustering and Support Vector Machines, the system dynamically refines its recommendations. Additionally, it incorporates proven study techniques and gamification to enhance user engagement and retention.**

***Index Terms*—K-means clustering, Learning modalities, SVM, VARK Theory.**

I. INTRODUCTION

The education sector is evolving rapidly, fuelled by advances in technology and a growing recognition of the need for personalized learning experiences. Traditional educational models as seen in classrooms adopt a ‘one-for-all’ approach, with content delivered uniformly to a diverse student population. However, this method does not account for the fact that learners absorb and retain information in different ways. As a result, many students struggle to keep with the material, not due to a lack of capability, but due to a mismatch between their learning preferences and the instructional methods used.

To address this issue, as theorised by many educational psychologists, we advocate the recognition of different learning modalitites. Among the most widely accepted models is the VARK theory, proposed by Neil Fleming, which classifies learners into four primary categories based on their sensory preferences: Visual, Auditory, Reading/Writing, and Kinaesthetic.

In parallel, the rise of artificial intelligence (AI) and data-driven decision-making in education presents new opportunities to operationalize theories like VARK on a personalized scale. Adaptive learning systems, powered by machine learning algorithms, have the potential to continuously monitor student performance and adjust instructional strategies in real time. These systems not only identify the most effective learning modalities for each student but also evolve with the learner over time.

This research paper introduces an AI-based adaptive learning system that leverages the VARK framework to recommend optimal study methods for users based on their demonstrated performance. The system uses an initial probabilistic approach when data is scarce and transitions to a more sophisticated machine learning pipeline as more data is collected. It incorporates K-Means clustering to detect learning type patterns and Support Vector Machines (SVM) for accurate classification. Moreover, the system is designed to accommodate multimodal learners and integrates gamified elements and cognitive strategies to boost engagement and knowledge retention.

II. METHODOLOGY

A. Concept

The VARK theory of learning, proposed by New Zealand educationalist Neil Fleming after working upon the VAK theory by Walter Burke Barbe, is the most popular theory of learning styles in humans. This theory categorizes human learning into four sensory modalities:

1. Visual: Watching or looking at something to learn.
2. Aural\Auditory: Listening to learn.
3. Reading and Writing: Reading and writing about a topic to learn about it.
4. Kinaesthetic: Performing and engaging in activities to learn.

Apart from this, Fleming also described the concept of ‘multimodality’, i.e., the concept of people having an affinity to more than one of these sensory modalities to learn.

While there have been a few researches criticizing and doubting the efficiency of this theory, it still remains the most popular and widely known theory describing human learning patterns, and ever since its birth, has inspired numerous other works.

We, having been inspired by this work ourselves, decided to develop an AI-based learning system that classifies users based upon their affinity for each learning method, and recommends what kind of learning they should engage in at a specific stage.

B. Working

The program we have developed works in two stages: The initial stage, when no data on the user is available. The main part will come into effect when enough data is collected based on how much the user scores upon learning with a certain method.

Initially, the system will have no data on the user, and all affinities to each learning modality will be set equally. After enough data has been collected, the main classification algorithm takes over and classifies the study method based on that.

1. Working of the initial stage

Affinities to each learning modality can be in the range of 10 to 90. The higher the affinity score is, the better a person is at learning with that modality.

All modalities are initially set to an affinity of 50 out of 100, reflecting a neutral assumption in the absence of historical data. The score calculation for each learning modality to find the best choice at a certain stage is given by the following equation:

$(CAS + Gap * 4) * RF$, where:

- *CAS* refers to the Current Affinity Score for a particular learning modality.
- *Gap* represents the number of lessons since a particular modality was last used, incentivizing diverse mode exposure.
- *RF* is a random factor that ranges between 0.96 and 1.00 to bring a little more variability in the learning mode used.

The idea behind this equation is as follows:

- The *CAS* parameter is required, as the learning modality that a person has the highest affinity with should be prioritized in the selection. *Note: Since the initial stage is mainly used because there isn't enough data to identify a user's affinity to a certain modality, this parameter is just a rough guess, which is required as despite there not being enough data, we must ensure that a user learns as much as they can from the learning mode that best suits them.*
- The *Gap* parameter increases the priority for using a certain learning mode for a user based upon the number of 'turns', i.e., the number of lessons it has been since it was used. This is in order to balance all the learning modalities, as people are, more often than not, multimodal learners who need a very specific balance of each of these learning styles to efficiently learn.

The multiplication by 4 on the Gap is the factor by which we determined the importance of balancing between different modalities. This factor can be changed depending upon how much importance others give to said balance, but for the sake of simplicity, we have included it as a constant and not a variable in the above equation.

- The *RF* is to bring about a little more unpredictability in the choice of learning mode. This is important for 2 reasons: At the very beginning, using this equation without the random factor pretty much guarantees what learning mode each student will get in the first four cycles. This will be common for all students and makes the program lose the feel of being customized for each user, and a little unpredictability helps in increasing the attention in a certain field in human beings. While the unpredictability here isn't major, it does contribute a little.

After calculating the scores for each modality, the one with the highest score is chosen as the learning option for the current lesson.

The effectiveness of the learning modes on a person, i.e., the user's affinity towards each learning mode, is calculated via tests. A test is conducted after each lesson, and the score determines how well the current learning mode is for them. The equation for changing the learning mode is as follows:

$NAS = min(90, max(10, CAS + (Score - IdealScore) * CAS(MaxScore * 5)))$, where:

- *NAS* is the New Affinity Score
- *CAS* is the Current Affinity Score
- *Score* represents the marks obtained in the current test
- *IdealScore* represents the ideal marks a student should obtain
- *MaxScore* is the maximum marks obtainable in the test

The idea behind this equation is as follows:

- The affinity score cannot fall below 10 or rise above 90, in order to maintain a healthy balance between all the learning modalities.

- The equation that changes the value of the current affinity score has been crafted in such a manner that even the most extreme scores won't let the score change too drastically. Of course, depending on the need, this equation can be changed to a different one.

The results of each test, along with the learning mode used in the lesson preceding it and its timestamp, are all stored within a database. Once the database has an enough number of entries, we move on from this stage to the main algorithm.

2. Working of the main algorithm

The system transitions from the initial probabilistic model to a machine learning pipeline once sufficient interaction data is collected. This pipeline comprises a continuous affinity update system, unsupervised clustering for learner grouping, and supervised classification for learner type identification. The detailed workflow is as follows:

1. Affinity Score Update: Each time a user completes a lesson and its corresponding test, the system updates the affinity scores for the modality used. The formula employed is: $NewAffinity = (OldAffinity * 0.8) + (TestScore/MaxScore) * 0.2$, where:
   a. *OldAffinity* = current affinity score for the modality.
   b. *TestScore* = marks obtained by the user.
   c. *MaxScore* = maximum achievable marks.

This weighted moving average ensures that recent performance impacts affinity, but historical data retains dominance (80% weight) to avoid overfitting to outliers.

2. Normalization: After each update, the affinity scores across all modalities are normalized so that $modalitiesAffinity = 1.0$. This normalization keeps the system scale-invariant and ensures a balanced input to machine learning models.
3. K-Means Clustering: The normalized affinity vector: $[VideoAffinity, TextAffinity, InteractiveAffinity]$ is passed to a K-Means clustering model (where k = 3) to form clusters corresponding to Visual learners, Reading/Writing learners, Kinaesthetic (interactive) learners. The clustering algorithm uses Euclidean distance as its metric and is retrained periodically (e.g., weekly) to adapt to emerging user patterns.
4. SVM Classification: Once the clustering assigns the user to a tentative group, the SVM classifier takes the same normalized vector and refines the classification. The SVM is configured as follows:
   a. Kernel: Radial Basis Function (RBF),
   b. C Parameter: 1.0 (to balance bias-variance trade-off),
   c. Gamma: 'scale' (auto-adapts to the input feature space).

The SVM's role is to enhance precision and handle overlaps between clusters, especially for users with multimodal tendencies.

5. Cross-Validation & Rule Check: As an additional safety net, the system performs a rule-based validation. If the highest single affinity (e.g., video = 0.7) matches the SVM classification, the learner type is confirmed. If there's a mismatch (e.g., video affinity is highest but SVM predicts Reading/Writing), the result is logged, and the SVM classification is favoured, but flagged for potential review in future iterations.

i. Model Training & Data Handling

- Dataset: The system uses an initial training set of anonymized user records (minimum 200 entries) and continues incremental learning as new data comes in.
- Retraining: The K-Means and SVM models are retrained every 500 new data points or every 2 weeks, whichever comes first.
- Storage: Every affinity update, cluster assignment, and learner type classification are time-stamped and logged in the central database for longitudinal tracking.

ii. Technical Considerations

The K-Means model may be replaced by DBSCAN or Gaussian Mixture Models (GMM) in the future to better handle non-spherical clusters.

The SVM classifier may evolve to a multi-layer perceptron (MLP) if richer feature sets (e.g., time-to-complete, engagement metrics) are integrated.

C. Other developments

1. Optimizing learning

In addition to identifying the right modality, our system incorporates techniques shown to enhance memory encoding and retrieval. It is not enough just to recognize how a person learns; what is also important is to understand how to make their studies efficient too. Recognizing the methods in which a person can study efficiently will help them in grasping any knowledge quicker and retaining it longer, no matter what learning modality they have an affinity towards. Dunlosky et. al. identified ten different studying methods. Some of those that we can incorporate into our system include:

- Summarization: Summarizing what has been learnt so far is probably the simplest method of studying we can aid in. It might just be revisiting old modules, or taking the aid of an AI chatbot to help summarize what was learnt.
- Rereading: Rereading is probably the easiest method of studying to implement, especially for learners who fall into the Read\Write category. This method would require no additional module, apart from one that actively encourages a user to visit the previous modules.
- Practice testing: Practice testing involves asking the user questions and grading them. While our system does entirely run on tests, what's more important in this scenario is to inform the user where they have fallen short and help them in improving upon those areas.
- Distributed practice: What distributed practice essentially means is that we must ask learners to properly schedule their studying times. We could award rewards, such as badges and titles, for students who log in as per schedule (not just log in daily).

Apart from the studying methods, there are some other ways in which learning can be optimized:

- Drawing attention: Ensuring that a user's attention is maximized towards their lessons helps in a huge way to boost their learning speed. There are many ways to draw a user's attention, but the most notable ones are:
  - Loud noises: This method would work best when dealing with auditory and kinaesthetic learners. Loud noises draw a person's attention to their source. Hence, if we were to play loud noises periodically for our auditory learners, it would ensure that their attention is constantly drawn back towards learning.
  - Progress bar: Showing a progress bar helps retain attention. A progress bar, which shows how much progress a user has made towards completing a lesson, keeps the user engaged and encourages them to quickly complete a given module.
- Punishments: Punishments are another way to make learning quicker. In order to avoid punishments, people tend to work harder towards a goal. While unpleasant, punishments have proven to be very effective as a learning method. We can incorporate punishment in our system by forcing a user to sit through extra lectures for a low score.
- Rewards: Rewards are also a very wonderful way to help users learn. Rewarding a user for doing well not only boosts confidence but also encourages them to return more frequently. Some major ways a reward can be implemented are:
  - Badges: We can reward users with digital badges for any major achievements, such as completing a module in a short time or achieving a streak of perfect scores. We can also continuously draw a user's attention towards these badges by implementing a progress bar, indicating how close they are to achieving it.
  - Skins: We can let users buy skins for their in-app avatars using in-app coins, awarded periodically based upon their performance.
  - Escape Punishment: A rare reward that lets a user escape punishment, if they were to ever face one in the future, can be incorporated as the rarest and the highest level of reward that users can set their goal to achieve.
  - Lucky wheel: People tend to lose interest in rewards when they are predictable. Hence, to introduce some variability, we can add a lucky wheel to randomize the rewards a user earns on each stage.

2. Accommodating the 4th Learner Type

While the current system classifies users into three primary learner types—Visual, Reading/Writing, and Kinaesthetic—future iterations of the algorithm will introduce a 4th learner type: Auditory learners. We plan to use audio engagement metrics (e.g., playback duration, frequency of rewinds) to strengthen the auditory affinity score over time. To incorporate this:

- Affinity Extension: The affinity vector will be expanded to include Auditory Affinity, resulting in$[VideoAffinity,\ TextAffinity,\ InteractiveAffinity,\ AuditoryAffinity]$. Affinity updates will follow the same weighted formula, but now with normalization over 4 modalities.
- Content Mapping: A dedicated auditory learning track (e.g., podcasts, voice notes, narrated tutorials) will be introduced, and each lesson will be tagged accordingly. New content pipelines will ensure parity across all 4 modalities.
- Clustering & Model Update: The K-Means model will be retrained k = 4 to reflect the new

cluster. The SVM classifier's input dimension increases to 4 features, with hyperparameters fine-tuned through cross-validation.

- Hybrid Learners: The updated system will better capture hybrid learner profiles (e.g., Visual-Auditory mix) by analysing pairwise affinity patterns. If significant hybrid clusters emerge, a soft clustering approach (like GMM) will be explored for smoother classification.
- Backward Compatibility: Existing user data will be retrofitted by initializing auditory affinity to a neutral baseline (e.g., 0.25) until enough auditory interaction is gathered.

III. CONCLUSION

This research introduces a novel AI-driven adaptive learning system that classifies learners based on their interaction with different modalities and continuously refines its suggestions through performance data. By leveraging theories like VARK and applying AI and ML techniques, the system addresses the need for individualized education, improves engagement, and enhances learning outcomes. Moreover, the integration of cognitive strategies, gamified incentives, and support for multimodal learning ensures both effectiveness and user motivation. As future iterations expand to include auditory learners and more nuanced hybrid profiles, the system promises to become a comprehensive tool for personalized education in the digital age.

IV. ACKNOWLEDGMENT

We would like to thank our university, MIT ADT University, for giving us the opportunity to do such a wonderful project. We would like to thank all the faculty and staff of the university for guiding us in all aspects of our project. Our special thanks go to our guide, Dr. Suvarna Pawar, who believed in us and guided us in every step of the project, and to Prof. Trupti Kudale, who followed through with us at each stage of our project.
We would also like to thank everyone we have likely forgotten to mention. Truly, the success of this project would not have been possible without all of you.

REFERENCES

[1] Dunlosky, J., Rawson, K. A., Marsh, E. J., Nathan, M. J., & Willingham, D. T. (2013). Improving students' learning with effective learning techniques: Promising directions from cognitive and educational psychology. Psychological Science in the Public Interest, 14(1), 4–58. https://doi.org/10.1177/1529100612453266

[2] Donoghue, G. M., & Hattie, J. A. C. (2013). A meta-analysis of ten learning techniques. International Guide to Student Achievement, 274–277.

[3] Prince, M. J., & Felder, R. M. (2006). Inductive teaching and learning methods: Definitions, comparisons, and research bases. Journal of Engineering Education, 95(2), 123–138. https://doi.org/10.1002/j.2168-9830.2006.tb00884.x

[4] Madhu, S., & Bhattachryya, D. (2020). Learning styles preferences among the students. International Journal of Education and Information Studies, 10(1), 39–43.

[5] Yin, C., Wang, Y., Li, B., & Gao, T. (2017). The effects of reward and punishment on the performance of ping-pong ball bouncing. Advances in Physical Education, 7(3), 256–262. https://doi.org/10.4236/ape.2017.73021.