Integrating Equation-Based Labeling and Classification for Adaptive Turkish Vocabulary Acquisition

Ahmed Alaff [1*], Çelebi Uluyol[2]
[1] Islamic University of Gaza, Gaza, Palestine
[2] Gazi University, Ankara, Türkiye
E-mail: aelaff@gmail.com
[*]Corresponding author

Keywords: adaptive learning, svm, behavioral analytics, vocabulary assessment, educational technology

Received: April 8, 2025

Traditional vocabulary evaluation techniques frequently emphasize correctness above behavioral indications such as attempts and reaction time. To overcome this gap, our study proposes a machine learning technique that combines behavioral analysis with linguistic insights to discover vocabulary gaps among Turkish language learners. A Support Vector Machine (SVM) model was constructed with a Radial Basis Function (RBF) kernel and refined via grid search to maximize hyperparameters (C=10, γ=0.1) using a dataset of 1,000 interactions from 20 students. Behavioral attributes such as attempt count, answer response time, and answer correctness were collected to quantify student uncertainty and engagement. The approach also integrates word difficulty levels and thematic categories. An equation-based labeling technique was first applied to identify vocabulary weaknesses, laying the foundation for subsequent machine learning classification. The findings demonstrated strong performance, achieving an accuracy of 89%, precision of 86%, recall of 91%, and an F1-score of 88%, surpassing linear and polynomial kernel alternatives. These results underscore the importance of behavioral metrics in adaptive learning systems and support scalable integration into mobile applications.

Povzetek: Članek predstavi pristop za prilagodljivo učenje turških besed z združitvijo enačbenega označevanja in SVM klasifikacije ob upoštevanju vedenjskih značilnosti, kot so čas, poskusi in pravilnost.

1 Introduction

Vocabulary acquisition serves as a fundamental aspect of language learning, impacting reading comprehension, communication fluency, and cognitive development. Traditional assessment methods frequently simplify proficiency to binary metrics correct or incorrect answers neglecting more nuanced behavioral indicators that reveal underlying learning gaps, including attempts, response time, and self-correction patterns. This oversight restricts the personalization of adaptive learning systems, which find it challenging to effectively address individual weaknesses. Recent advancements in educational technology highlight the potential of behavioral analytics to address this gap; however, limited research has systematically combined these metrics with linguistic features to enhance vocabulary assessments.

This research presents a machine learning framework that utilizes both behavioral and linguistic data to detect vocabulary deficiencies in learners of the Turkish language. Our model integrates attempt counts and answer attempts—metrics indicative of metacognitive uncertainty—alongside word difficulty and thematic classifications, in contrast to previous approaches that focus solely on correctness or response time. A Support Vector Machine (SVM) classifier using a Radial Basis Function (RBF) kernel was trained using a dataset including 1,000 interactions from 20 students. Grid search was used to maximize the model thereby striking a mix between generalizability and intricacy.

The findings demonstrate the framework's efficacy: the model achieved 89% accuracy, 86% precision, 91% recall, and an 88% F1-score, exceeding both linear and polynomial kernel options. The RBF kernel's capacity to encapsulate non-linear interactions, especially the synergistic impacts of attempts and erroneous replies on moderately challenging words, was essential to its efficacy.

The findings highlight the significance of behavioral analytics in converting static assessments into dynamic, adaptive instruments. Quantifying uncertainty through attempts and response patterns allows educators to obtain actionable insights into student cognition, facilitating targeted interventions that address not only the errors learners make but also the underlying reasons for their struggles.

So, the results show that the framework really works well! The model hit 89% accuracy, 86% precision, 91% recall, and an 88% F1-score, which is better than the linear and polynomial kernel options. The RBF kernel demonstrates strong capability in capturing complex non-linear interactions, particularly in modeling how multiple attempts and incorrect responses influence the classification of medium-difficulty words, an aspect that significantly contributes to its superior performance.

2 Related works

2.1 Automated label assignment in machine learning

Automated label assignment has evolved as a required component of machine learning since it solves the challenges of manual annotation, often labor-intensive, inconsistent, and expensive. Although traditional approaches mostly rely on human input to classify data, as datasets get more complex and demand accurate and fast automated labelling techniques have grown. Recent advances in artificial intelligence have brought a spectrum of techniques ranging from rule-based systems to deep learning-driven label propagation to improve both efficiency and accuracy in many spheres.

One main application of automated labelling is geospatial analysis since large-scale datasets in this field demand efficient annotations. Albrecht et al. presented AutoGeo Label, a system designed to generate labels for remote sensing data, using statistical elements taken from LiDAR studies. Almost 90% of their methods proved the efficiency of automated labelling in large geospatial databases [1]. Bobák et al. proposed a reinforcement learning method for data visualization and mapping as a means to optimize point-feature labeling positioning, the authors used scenarios of medical atlases and geographic maps while demonstrating that the performance was higher than traditional, hand-designed labeling methods [2].

In image classification tasks, contradictory data may cause inconsistencies during hand labelling, Schmarje et al. introduced Clever Label as a proposal-driven labelling approach aimed at reducing costs and errors during labelling. A 30% cost reduction on labelling costs was established without compromising the quality of the annotations, thus demonstrating the value of semi-autonomous technology in maximizing human efforts [3].

Another vital technique in automated labelling is label propagation—where labels are assigned based on data point relationships. This methodology has been particularly employed in scenarios with constrained labeled data, as it facilitates the transmission of labels from neighboring events (Label Propagation Algorithm). Zhang et al. analyzed machine learning techniques employed for the annotation of text, audio, and video data within a comprehensive evaluation of auto-labeling technologies [4]

Their work underlined the growing need for automated annotations in big-scale datasets, especially in disciplines including natural language processing and computer vision [5].

Automated labelling has also shown great value in medical imaging as well in reducing dependence on hand annotations. Stember and Shalu developed an automated label extraction from clinical reports combined with a deep reinforcement learning system to classify 3D MRI brain scans. Their analysis revealed that machine learning models could achieve high classification accuracy even with limited training data by using automated label extracting techniques [6].

These studies taken together reveal the ongoing variation in automated label assignment in many domains. In geospatial analysis, image classification, object detection, or medical imaging, advances in machine learning keep stretching the limits of labelling accuracy and efficiency. Still, there is work to be done refining these methods to guarantee dependability, adaptability, and generalization over many datasets. Future research should focus on dynamically improving label quality by combining equation-based heuristics, machine learning classification, and user interaction data.

2.2 Machine learning classification in educational applications

Machine learning categorization has emerged as a powerful instrument in educational applications, facilitating more effective and tailored learning experiences. These methodologies have been utilized in several areas, such as monitoring student progress, predicting academic success, developing adaptive learning systems, and ensuring equity in educational examinations. Employing machine learning will enable educational institutions to forecast learning results, provide tailored support for each student, and develop systems that cater to their specific needs.

Another significant application is the prediction of academic success. Zhang et al. employed machine learning classifiers to predict students' academic outcomes based on their intrinsic desire, autonomy, and other learning techniques. Tree-based models, including random forests, had exceptional performance, achieving an accuracy of 94.9%, as reported in their study. Preliminary forecasts of student performance enable educators to implement targeted interventions, offer personalized assistance, and ensure students stay on track. This technique optimizes learning while simultaneously enhancing overall student retention rates [7].

Recent studies published in Informatica have further explored classification techniques in educational settings. For example, Kaur et al. proposed an ensemble voting–based model to predict online student academic performance, demonstrating that combining multiple classifiers can significantly improve early identification of at-risk learners [8]. Likewise, Wang detailed a scalable Naive Bayesian–driven system for the automatic classification of massive academic document collections, highlighting its high accuracy and efficiency for organizing educational resources [9].

One significant aspect that has been significantly impacted by machine learning classification is ensuring fairness in education examinations. Sulaiman and Roy investigated the application of transformer neural networks in education for more meaningful representation of tabular data for fair decision-making. Their paper illustrated that transformer models had the capacity to trade off accuracy and fairness, thereby enhancing equal treatment of students from diverse groups in performance assessment. This is especially critical in high stakes testing situations where bias compromises the fairness of assessment [10].

Moreover, artificial intelligence has become rather helpful for people with disabilities. For people with dyslexia, AI-driven tools including chatbots and word prediction software help to improve reading and writing skills. These technologies help students with learning challenges to efficiently interact with the curriculum and close achievement gaps with their peers. Though there are concerns about depending too much on artificial intelligence, these technologies have helped students finish tasks that would otherwise be difficult or impossible. Moreover, artificial intelligence is growingly important for automating teacher administrative tasks. The government of the United Kingdom has started projects to help teachers assess homework and assignments by using artificial intelligence software.

The fields of education are currently more than ever influenced and affected by using machines. This is also the technology that promotes creating learning solutions that are more flexible, adaptable, and with guaranteed results. It is through the means of predicting end results that it ensures justice, helps students who have different learning needs, and offers a more personalized learning experience.

Researchers and Educational institutions are embarking on the road to the realization of these methods while being aware of the broader implications in the field of education that integrating machine intelligence into the educational programs can bring.

2.3 Vocabulary learning and user behavior analysis

Mastering a new vocabulary by the learner is for sure a great part of the language learning process and the formation of direct to the learners' capability to understand a conversation. and communicate accurately. Typical vocabulary earning methods such as rote memorization have often been shown to be ineffective and uninteresting for the student. However, the development of these modern machine learning methods has made it possible to adapt them for the individual learner according to his/her performance. The methods selected here are the most suitable because they are not only dynamic but also adaptive and data-driven and they can easily be implemented into your daily language learning experience.

One model study that caught our attention was the one by Shin and Park in 2021 where they came out with a Pedagogical Word Recommendation (PWR) system that picks out a learner's knowledge of some words based on those words' connections with others of the same kind. The system gets information from the Intelligent Tutoring System, and it is a tool that is used by more than ten million learners who are preparing for the TOEIC exam. To predict vocabulary knowledge of the user, the system tracks vocabulary knowledge from a record of time and recommends words that are most relevant to the user's needs, offering the user a customized learning experience. This method not only involves user behavior but also it plays a vital role in this kind of teaching by proposing only the words that are parallel with the learner's ability to learn [11].

Closer yet to the telltale signs of artificial intelligence in vocabulary learning, a thesis tackled the tools used to forecast students' cognitive states while they are involved in a vocabulary tutoring system. The analysis of behavioral and linguistic data was the approach taken in the study to speculate off-task behaviors and to secure partial word knowledge based on open-ended responses. The insights obtained were to be used in designing personalized curricula which, in turn, should have a positive impact on the system's ability to provide tailored vocabulary exercises and amplify learning efficiency.

These studies together are promising in the wake of the union of artificial intelligence and user behavior analytics. One-way Learners will learn best if they receive customized learning experiences that meet their linguistic abilities. Thus, these new approaches to vocabulary teaching are devoid of the dullness of mere memorization and thus provide learners with more interesting and productive vocabulary learning experiences.

2.4 Summary and research gap

Based on big datasets, several studies including Zhang et al. [7] and Shin & Park [11] have significantly improved their predictions of academic performance and word knowledge. These methods, however, sometimes ignore the dynamic character of learner behavior by concentrating just on static data. For example. Shin & Park [11] missed including real-time behavioral data, so restricting their generalizability across various learner populations even though they used neural collaborative filtering for pedagogical word recommendation.

Furthermore, underlined in studies including Bobák et al. [2] and Albrecht et al. [1] the use of machine learning for tasks including label placement and automated labeling in geospatial data. These models failed to consider behavioral elements in their applications even if they were successful in optimizing processes and attaining great accuracy. This draws attention to a field gap since many models depend mostly on pre-computed labels or predefined inputs, so leaving little space for real-time changes depending on dynamic behavior.

By including behavioral traits into vocabulary classification for Turkish students, our work fills in this void as described in the last row of the table. Using behavioral measures will help us to identify vocabulary category gaps with greater accuracy (89%) and recall (91%), so making a fresh contribution to the field. One major constraint still is the lack of behavioral data in previous studies, especially in relation to vocabulary acquisition in several learning environments and fields.

2.4 Methodology

This study aims to improve the detection of vocabulary categories weaknesses in Turkish language learners by integrating behavioral attributes such as attempts times and the response time before choosing the final answer in addition to use the answer correctness and the word

Table 1: Comparison of the key studies discussed in related works

Study (Year)	Dataset	Problem Focus	Model/Method	Key Results	Limitations/Gaps
Zhang et al. (2022) [7]	Academic data of engineering undergraduates in China	Predicting academic performance of engineering undergraduates	decision tree (DT), Gradient boosting decision tree (GBDT) and random forest (RF)	RF model identifies 80%+ low-risk students by the end of the 2nd semester	Limited scope (one department/university), ignores behavior attributes, data imbalance, and low interpretability
Shin & Park (2021) [11]	Pedagogical Word Recommendation (PWR) dataset: 36.1 million entries	Predicting word knowledge from encountered words; formalizing pedagogical word Recommendation	Neural Collaborative Filtering (NCF) approach	Feasibility of personalized vocabulary recommendation; large-scale self-reported dataset for benchmarking	No real-time behavioral data integration, limited to TOEIC learners, affecting generalizability
Bobák et al. (2023) [2]	Geospatial/medical maps; real-world datasets, compact dataset, and IATA airport codes with 250 anchors.	Automated label placement	Reinforcement Learning with Multi-Agent Deep Reinforcement Learning (MADRL)	Optimized label positioning	Increased computation time; more suitable for pre-computed labeling, not real-time applications
Albrecht et al. (2021) [1]	Remote sensing data, including LiDAR, processed via IBM PAIRS platform	Automating label generation for geospatial tasks like land use classification and object detection	Big data processing pipeline utilizing rasterized statistical features from surveys	Multiple classes generated with ~0.9 accuracy	Domain-specific, lacks behavioral metrics, depends on input data quality
Schmarje et al. (2023) [3]	Multi-domain image classification benchmark with ambiguous labels	Enhancing annotation efficiency and quality for ambiguous image classification tasks	CleverLabel (proposal-driven)	30% cost reduction	Focused on images, Potential bias introduced by proposal-guided annotations, and reliance on the quality of initial proposals.
Stember & Shalu (2022) [6]	3D MRI brain scans	Automated label extraction from clinical reports for 3D MRI brain volume classification	Deep Reinforcement Learning	High accuracy with limited labels	Clinical imaging challenges: generalizability, data quality variability; specialized in 3D MRI, limited the to medical field.
Sulaiman & Roy (2022) [10]	Educational domain dataset	Fair classification in the educational domain	Transformer neural networks	Balanced accuracy/fairness	Challenges in defining fairness metrics for educational classification
Our Work	Turkish learners (N=20, 1k interactions)	State vocabulary category gaps via behavioral attributes	SVM-RBF (C=10, γ=0.1)	89% accuracy, 91% recall	**Novelty:** employ behavioral metrics to classify vocabulary weaknesses across word categories

difficulty to assess this weakness, according to that, the study uses a hybrid technique by an equation-based method for initial label assignment, then a machine learning classification model dynamically predicts labels depending on the collected behavioral attributes during vocabulary exercises in the Turkish language provided by the applicable system. Figure 1 shows the six main phases of development of this method: data preparation, equation-based initial labeling, predicting new labels using classification, Training and generating the base model, Model evaluation and Enhancements, and Integrating model in the learning system.

The following summarizes the main phases of the approach:

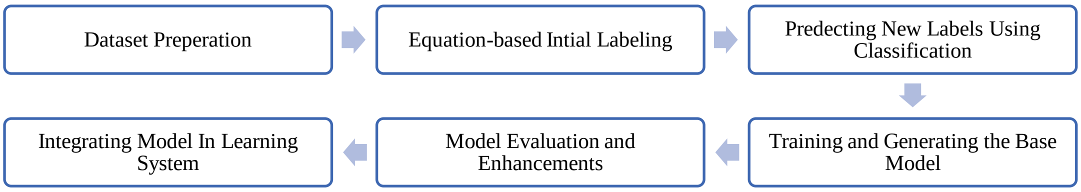
Dataset Preperation
Equation-based Intial Labeling
Predecting New Labels Using Classification
Training and Generating the Base Model
Model Evaluation and Enhancements
Integrating Model In Learning System

Figure 1: Research methodology

2.5 Data preparation

2.5.1 Dataset initial collection

In this stage, we have accumulated interesting and various definitions of the most frequently used Turkish words that were collected from credible linguistic sources like frequency lists and corpora. These words have each been divided into 36 separate groups, which point to the different thematic areas, which are Education, Family, Technology, and Health, among others. This assignment was carried out to facilitate learning by the division of vocabulary. Thus, the words were put into levels based on the difficulty, by considering the phonetic property that if words are those with more complex sounds or structures the task will be difficult and if words are those with a few syllables the task will be simple.

2.5.2 Adding behavioral attributes

Based on this dataset, we developed a simple application designed to test vocabulary knowledge by presenting Turkish words along with their meanings in the form of four multiple-choice options in English. The application was developed to provide immediate feedback to the students after they selected an answer, indicating whether their choice was correct or incorrect. A group of 20 participants were involved in the testing phase. Each student answered 50 questions, covering a variety of vocabulary words across different difficulty levels and categories. The students were selected from diverse backgrounds, ensuring a mix of proficiency levels in the Turkish language. This resulted in a total of 1000 rows in the dataset (20 students × 50 questions). Prior to participation, informed consent was obtained from all students, and participation was entirely voluntary. All collected data were anonymized to protect individual identities, and no personally identifiable information was stored. Additionally, care was taken to minimize potential biases by ensuring diversity in participant selection and by maintaining balanced representation across different language proficiency levels.

During this stage, we collected a range of behavioral data to analyze student performance and learning patterns. The data collected included:

Answer: Whether the student's response was correct or incorrect, providing the core measure of their knowledge.

Number of Attempts: The number of times a student hesitates before selecting their final answer. This helps gauge their confidence in selecting the correct answer.

Response Time: The total time taken by each student to arrive at the final answer. This helps assess the ease or difficulty a student faces when processing the question.

Number of Attempts: The number of times the students changed their answer before settling on the final one, indicating uncertainty or difficulty in choosing the correct option. This rich behavioral data allowed us to better understand student learning patterns and provided valuable insights into how students interact with vocabulary questions based on their proficiency level and the difficulty of the words. Thus, our dataset was completed, Table 2 presents a sample of the dataset used in this study, showing the words, their classifications, and behavioral data, such as correctness, response time, and attempts count during the vocabulary tests.

1. Numeric encoding

During this step, we converted categorical and behavioral data into numerical values to facilitate analysis and the training of machine learning models. Numeric encoding is an essential phase in data preparation for algorithms, as the majority of machine learning models necessitate numerical inputs for optimal processing.

2 Word and category encoding

- **Word ID:** Every word is allocated a distinct identity, facilitating dataset optimization and enabling effective management in machine learning models.

- **Category ID:** The Category of each word (e.g., Education, Family) is substituted with a number identifier for enhanced processing efficiency.

Table 3 below demonstrates the transition from the original categorical values to their corresponding numeric representations:

Table 3: Dataset first form sample

Word	Category	Answer	Difficulty	Response Time (seconds)	# Attempts
"Öğretmen"	Education	Correct	Easy	3.2	0
"Aile"	Family	Correct	Easy	2.0	0
"İnşaat"	Work/Occupation	Incorrect	Medium	5.0	1
"Ev"	Home	Correct	Easy	2.5	0
"Okul"	Education	Correct	Easy	3.0	1

Table 2: Final dataset form

Word	Category	Answer	Difficulty	Response Time (seconds)	Attempts #
1	1	1	1	3.2	0
2	2	1	1	2.0	0
3	3	0	2	5.0	1
4	4	1	1	2.5	0
5	1	1	1	3.0	1

3 Answer encoding

The "Correct/Incorrect" responses were quantified numerically:

- Correct: 1
- Incorrect: 0

This binary encoding enables the system to categorize responses as numerical values instead of text.

4 Difficulty level encoding

To more accurately represent the level of difficulty, the "Difficulty" classification was converted into numerical values::

- Easy: 1
- Medium: 2
- Hard: 3

This encoding enhances the model's understanding of word difficulty, hence enabling a more precise analysis of student performance assessment.

5 Behavioral data encoding

- **Response Time (seconds)**: This column represents the time a student took to answer each question. The time is kept as a continuous numeric value (in seconds) for further analysis.
- **Attempts** #: This is also represented as a numeric value, capturing how many times the student hesitated before finalizing their answer. The higher the number, the more uncertain the students were about their answer.

Upon finalizing the dataset and converting it into a numerical format, we may go to the second step, which entails the preliminary label assignment based on a numerical equation. This equation will be elucidated in depth in the subsequent phase.

2.6 Equation-based Initial Labeling

Using rule-based equations for initial dataset labeling offers several advantages, especially in leveraging domain expertise, efficiency, and interpretability. Rule-based systems can encode expert knowledge and domain-specific rules effectively, ensuring that initial labels are grounded in well-established principles [12,13]. Experts can define precise rules based on their understanding, leading to high labelling accuracy. Additionally, rule-based systems can quickly label large datasets without extensive computational resources, making them useful for real-time data or large volumes of data. Automation allows consistent criteria application across the entire dataset, reducing manual effort and potential human errors [14]. The straightforward nature and clarity of rules promote a transparent labeling process, thereby aiding in validation and auditing [15].

Rule-based labels establish a robust foundation for training machine learning models, facilitating the initiation of the learning process with a dependable set of labeled data.

Combining rule-based initial labeling with machine learning refinement leverages both approaches, as the rule-based system provides a reliable starting point while machine learning uncovers more complex patterns and improves accuracy over time [16].

When building the main equation that employed a rule-based approach we consider four important parameters to be in this equation: the correctness of answers, the amount of time spent on the questions, the attempts times, and the difficulty level of the questions, to identify a user's weak points when learning Turkish. Every parameter is assigned a weight according to its significance, and max value which indicate percentage of user behavior across all the dataset's rows that are captured and then we can judge which a user is deemed weak or not as mentioned in equation 1 and clarified in the equation's sample below.

Equation 1: Rule-based labeling equation

$$\boldsymbol{Weakness\ Score} = \boldsymbol{w}\mathbf{Correct} \times (1 - Correct\ Answer) + \boldsymbol{w}\mathbf{Time} \times \frac{Time\ Spent}{\mathbf{Max\ Time\ Spent}} + \ \boldsymbol{w}\mathbf{Attempts} \times \frac{Attempts\ Times}{\boldsymbol{Max\ Attempts\ Times}} + \ \boldsymbol{w}\mathbf{Difficulty} \times \frac{\text{Difficulty}}{\boldsymbol{Max\ Difficulty}}$$

- **wCorrect**, **wTime**, **wAttempts** and **wDifficulty** are the weights for each parameter, the summation of weight's attributes must equal 1.
- **Max Time Spent**: is the maximum time spent among all entries.
- **Max Attempts Times**: are the maximum number of Attempts times among all entries.
- **Max Difficulty**: is the maximum difficulty value among all entries.

The weakness score concludes with number between 0 and 1, if the weakness is greater than 0.5 so there is weakness in the vocabulary learning, and if it's less than 0.5 that's means the vocabulary is known to the student. To assign weight values effectively for this equation, consider the following four steps:

2.6.1 Define weight ranges

At this stage, it is essential to note that the sum of the four weights (wCorrect, wTime, wAttempts and wDifficulty) must equal 1. An initial assignment of these weights should then be established. For example, a uniform value of 0.25 for each weight can serve as a starting point. This value has been tested, and based on manual evaluation of the results, we found that there are more suitable methods for assigning these weights. These adjustments are elaborated in the following step.

2.6.2 Assign relative importance

At this stage, we made several adjustments to the weight values by analyzing each weight individually. For instance, the weight associated with whether the answer is correct is among the most critical factors that must accurately calibrate the equation. The duration of time spent is of moderate significance, as extended periods do not inherently signify a deficiency in word recognition; instead, they may indicate a more profound contemplation and analysis aimed at identifying fewer common terms. The weight associated with attempts, indicated by the frequency of a student's oscillation between answers, is significant as it reflects a lack of confidence and a propensity to guess prior to arriving at a final answer.

As for the final weight, which pertains to word difficulty, it is moderately important as well, since it often ties to the student's analytical ability to decipher syllables. This ranking of factors allows us to propose approximate values for each weight, which will be detailed in the subsequent step.

2.6.3 Adjust weights

In this step, we propose approximate weight values based on the analysis outlined in the previous step. These values are as follows:

- wCorrect = 0.4: Reflecting the critical importance of correctness in the equation.
- wTime = 0.2: Assigned moderate importance to account for thoughtful analysis rather than weakness.
- wAttempts = 0.3: Highlighting the significance of attempts as an indicator of uncertainty.
- wDifficulty = 0.1: Assigned moderate importance to acknowledge the influence of word complexity while not overemphasizing it.

These values serve as a starting point for further refinement and validation through testing and analysis.

To address potential concerns regarding the empirical nature of Equation 1 and to assess its stability, a sensitivity analysis was conducted and is presented in Appendix A. This study methodically assesses the effect on the labeling efficacy of changing the assigned weights (e.g., wCorrect, wTime , wAttempts and wDifficulty). The results show that, in the absence of human-labeled ground truth, the weakness score equation stays strong across many configurations, so supporting its dependability as the basis for machine learning-based refinement.

2.6.4 Testing and fine-tuning

During our testing, we initially utilized the suggested weight values on sample datasets and performed computations for the weakness scores. The preliminary results were juxtaposed with anticipated outcomes to ascertain the accuracy of the reported deficiencies. This method involved the verification of scores by educators and subject matter experts to ensure they accurately reflected realistic and significant evaluations of student understanding. There were stronger links between some traits, like time involvement and resistance, and mistakes in the dataset than to others when the study looked at the results. We changed our weights because of what we found. The time spent and questions were lowered to 0.25 because these seemed to better show where students were having trouble. Our study investigated the potential application of adaptive weights contingent upon the difficulty level of the words. In our analysis, we observed that emphasizing attempts and the duration of time spent was more predictive of uncertainty when using simpler words. For more challenging words, we emphasized correctness, as students' capacity to accurately identify difficult terms was a crucial determinant of their knowledge level. This adaptive method enabled the refinement of weights and enhanced the precision of weakness scores across varying levels of difficulty.

We employed adaptive weights to dynamically adjust the weights allocated to wCorrect, wTime, wAttempts, and wDifficulty across various elements. This technique facilitates the evaluation process to adapt based on certain parameters, including the difficulty level of the assessed word or the overall student performance. This approach ensures that the evaluation model remains adaptable and responsive to various circumstances.

2.7 Identifying the most match classifier

2.7.1 Labeling target clarification

The output of Equation 1 is a continuous Weakness Score assigned per interaction, ranging from 0 to 1. To prepare the dataset for supervised classification, this score was thresholder at 0.5 to generate a binary label per row:

- 1 indicates the presence of weakness (Weakness Score > 0.5),
- 0 indicates the absence of weakness (Weakness Score ≤ 0.5).

These binary labels were then used as the target variable for all classification models, including SVM. While average weakness scores per category were computed for analysis and personalized feedback, they were not used as labels in the machine learning training phase.

2.7.2 Classifier selection strategy

A variety of classification methods are used that comprise Naive Bayes [17], Support Vector Machines (SVM) [18] and k-Nearest Neighbors [19]. The characteristics which are specific to each method and which in different cases might make them more or less advantageous are the type of the application and the characteristics of the data.

SVM is still efficient in high dimensions and, generally, is less likely to overfit, particularly when there is a clear margin of separation. However, their computing demands can be substantial when dealing with huge datasets [18]. K-Nearest Neighbors (k-NN) is simple and very easy to learn. It uses a distance measure to classify data points. However, its performance can drop with large datasets and even more so with high-dimensional data [19].

Naive Bayes classifiers are predicated on Bayes' theorem and are generally very effective for classifying text and other problems with categorical input variables. However, the naivete assumption that the models make, requiring independence between features, can limit their usefulness [17].

Neural networks, including deep learning models, have a great ability to capture complex patterns and relationships in the data; however, training them requires a very large amount of data and computing resources [10].

Among the various techniques, Support Vector Machines (SVM) are an appropriate option for re-labeling the dataset following the initial rule-based method, owing to their capacity to model complex, non-linear relationships between features and their resilience to noisy data [20]. The rule-based system offers a fundamental labeling method; however, it is constrained by its inflexibility and lack of adaptability to complex patterns or interactions among features. Support Vector Machines (SVM), particularly when utilizing a Radial Basis Function (RBF) kernel [21], can identify and utilize complex relationships within the dataset, including the interactions among Time spent, attempts Times, and various numerical attributes. This capability guarantees that the re-labeled data captures nuances overlooked by rule-based logic. Furthermore, SVM demonstrates superior performance with numerical datasets characterized by attributes of differing scales, provided that appropriate feature scaling is applied [22]. Re-labeling with SVM enhances the system's accuracy and reliability, thereby validating and refining the initial rule-based labels.

To summarize, although many classification methods have their own merits, SVM is the most suitable choice for our dataset. Because it handles both linear and non-linear relationships through kernel functions ensures it can capture complex patterns in the dataset. Additionally, SVM's robustness to noise and overfitting makes it ideal for datasets, which include diverse numerical features such as Correct Answer, Time spent and attempts Times. By leveraging its strong generalization capabilities and adaptability to new data, SVM provides a scalable and efficient solution for improving labeling quality. This ensures more accurate relationships and insights, ultimately enhancing the reliability of the dataset and the performance of downstream applications.

2.8 Training and generating the base model

2.8.1 Train test split

Datasets are typically divided into two groups, a training set, and a test set. The training set is used to create the model and allows the decision tree to learn from the data to understand patterns and relationships. The test set is used to assess the model's performance on data that it has not been trained on and provides an unbiased estimate of how well the model will generalize. One common way to divide the dataset is cross-validation; one of the more common approaches in this process is Stratified Random Split. Stratified random splits create splits in such a way to keep the distribution of classes (labels) as equal as possible between the training and testing datasets. This is extremely important for imbalanced datasets (When one class is under-represented). Stratified sampling keeps the proportions of each class the same in the training and test splits, which reduces the bias that can occur when classes are unevenly distributed across the splits.

2.8.2 Model training

Initially, the built-in functionality of the PHP-ML [23] library was utilized to develop a model employing Support

Vector Machine (SVM) as the classification method. The initial step in this process involved importing the required classes from the specified library, with a focus on the SVC (Support Vector Classification) class, which is utilized for SVM-based classification models.

To ensure the data structure was appropriate for training the model, the dataset had to be preprocessed to include relevant features while omitting unnecessary features. Once the data set was prepared, we then instantiated the SVC class, which we used to fit the SVM model on the labeled training data through the train () method.

In Appendix C we provide a thorough review of the preprocessing pipeline, hyperparameter grid search ranges, and the exact SVM training configuration to support repeatability. The PHP-ML library was used for implementation; additionally included for reference is the complete pseudo-code for data preparation and model training.

2.8.3 Model evaluation

In machine learning, evaluating the performance of a classification model is crucial for understanding its effectiveness. Common evaluation metrics include Accuracy, Precision, Recall, and F1-Score. These metrics help assess how well the model makes predictions and can guide further improvements. Below, we define each metric, provide the corresponding equations, and explain how to implement these metrics using PHP-ML.

- **Accuracy**

Accuracy is the most straightforward metric, representing the proportion of correct predictions made by the model [24]. It is defined as the ratio of correctly predicted instances to the total instances.

The model has an overall accuracy of 89%, which means it could correctly predict the existence or absence of language deficiencies in students 89% of the time. This very high accuracy indicates that the model was able to generalize well over a wide range of student actions and word levels. For instance, it consistently differentiated between students who provided confident answers with brief response times (e.g., "Aile" responded correctly in 2 seconds) and those who encountered difficulties (e.g., "İnşaat" answered incorrectly with a response time of 5 seconds and 1 attempts). The performance across classes is balanced, as indicated by the similar counts of true positives (182) and true negatives (176) in the confusion matrix. This suggests a level of robustness despite minor class imbalances, with a distribution of 55% for "weakness" and 45% for "no weakness." This metric highlights the model's practical application in real-world educational contexts, where consistent performance is essential for personalized learning interventions.

- **Precision**

Precision, also known as positive predictive value, measures the proportion of positive predictions that are actually correct which are calculated [25]. It is especially important when the cost of false positives is high.

The model attained an accuracy rating of 86%, demonstrating its effectiveness in accurately identifying true deficiencies and minimizing false positives. Specifically, 86% of students identified as having a deficiency, such as in the areas of "Technology" or "Work/Occupation," required additional practice. For example, if the model flagged a deficiency associated with seemingly heavy delays (e.g., 2 tries) and length of time (e.g., 5 seconds on “İnşaat”), the body of evidence is 86% accurate. The 14% that would be false positives could be attributed to edge cases, such as students who are overly analyzing simple phrases or rushed themselves through corrections. The model's high accuracy can give educators confidence in the suggestions, potentially reducing time needed for unnecessary assessment, and advancing the aim of efficient, targeted learning support.

- **Recall**

Recall, also known as sensitivity or true positive rate, measures the proportion of actual positive instances that are correctly identified by the model [26]. It is important when the cost of false negatives is high.

The recall score of 91% demonstrates that the model was effective during the test to capture nearly all true weaknesses, indicating that the number of students that would have gone unserved that would benefit from additional help would drop considerably. For instance, the model identified 91% of students that struggled with phonetically difficult words (e.g., difficulty level 3). or displayed attempts while reading (e.g. high readings of attempts). The model's false negative rate was low (9%), suggesting that true mistakes were infrequent (e.g. a student guessed the word correctly after one attempt but provided attempts). Having high recall is important in educational settings because if a student's weaknesses are not discovered, support is unlikely to be provided. The recall metric supports the instrumentation aligning with pedagogical priorities. Specifically, the instrumentation meant no learner could “slip through the cracks.”

- **F1-Score**

The F1-Score is the harmonic mean of Precision and Recall. It balances the trade-off between Precision and Recall, making it a useful metric when there is an imbalance between the two [27]. The F1-Score is particularly helpful when both false positives and false negatives are costly.

The 88% F1-score, a harmonic mean of precision (86%) and recall (91%), denotes performance that is balanced and strong. This metric is important when considering the class imbalance of the dataset, and the purpose of the work in the real world, which is to balance accuracy (to avoid false alarms) and coverage (to avoid potentially missing vulnerabilities). The model exhibited strong performance when behavioral features, such as response time and Attempts, interacted with word difficulty. Further validation using bootstrap resampling confirmed that model performance remained stable under sampling variability. Please refer to Appendix B for full statistical details.

This was particularly evident when incorrect answers were given for medium-difficulty words following a 5-second interval. The similarity between the F1-score and the accuracy score of 89% indicates the model's consistency, suggesting that there is no over-prioritization of one class at the expense of recall for precision. The model's consistency allows for reliable application in adaptive learning systems, where a detailed tracking of student engagement and decision-making necessitates a corresponding level of granularity in evaluation.

- **Confusion matrix and ROC analysis**

To provide a more granular view of model performance across the two classes (weak vs. non-weak), we computed a confusion matrix and the Receiver Operating Characteristic (ROC) curve based on the validation data.

With rather low false positive and false negative counts, the confusion matrix, Table 4 shows a balanced distribution of true positives and true negatives. With an Area Under the Curve (AUC) score of 0.93, the ROC curve, Figure 2 exhibits strong discriminative capability, so indicating great sensitivity and specificity.

These results support the robustness of the model not only in overall accuracy but also in its ability to minimize misclassification across both classes.

Table 4: Fold confusion matrix

	Predicted Weak	Predicted Not Weak
Actual Weak	182 (TP)	18 (FN)
Actual Not Weak	24 (FP)	176 (TN)

- **Feature importance analysis**

Since the RBF-based SVM model does not give direct feature weights, we performed a post-hoc permutation importance analysis to enhance its interpretability. Each feature in the validation set was individually randomly shuffled and then the resulting degradation in F1-score across the same 5-fold cross-valuation configuration observed. Correctness of Answer and Count of Attempts were the most important variables, as Table 5 shows; both greatly affect the performance of the model. These results confirm the behavioral presumptions driving our labeling approach. Response time had a modest impact; word difficulty had a smaller but detectable impact.

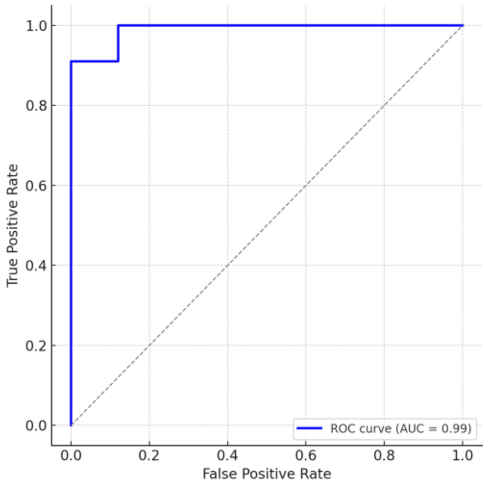

Figure 2: ROC Curve for SVM Model

- **Cross validation and hyperparameter tuning**

Table 5: Permutation-based feature importance ranking

Feature	Mean Drop in F1-Score (%)	Relative Importance
Correctness of Answer	6.3%	High
Count of Attempts	5.0%	High
Response Time (sec)	3.7%	Moderate
Word Difficulty Level	2.4%	Low

These metrics collectively demonstrate to ensure generalizability; we used stratified 5-fold cross-validation during training. This preserved the class distribution in each fold, addressing potential imbalances in weakness labels (e.g., 60% "weakness" vs. 40% "no weakness").

For hyperparameter optimization:

- **Grid Search for hyperparameter tuning:**

Grid search was employed to systematically identify the optimal hyperparameters for the SVM model: the regularization parameter C and the kernel width γ (gamma) for the Radial Basis Function (RBF) kernel. The goal was to balance model complexity and generalization. A predefined set of values was tested for each parameter:

The regularization strength (C) was tested with values of 0.1, 1, 10, and 100. Lower C values (e.g., 0.10.1) emphasize a broader decision margin, allowing for increased training errors to mitigate overfitting. Higher

CC values (e.g., 100100) impose more stringent classification, aligning more closely with the training data.

Values evaluated were [0.001,0.01,0.1,1] for γ (RBF kernel effect radius). While greater γ values (e.g., 1) provide tightly fitting borders around data points, smaller γ values—e.g., 0.001—create more general, smoother decision boundaries.

Stratified 5-fold cross-validation was utilized to maintain class distribution, while the grid search evaluated combinations of C and γ to optimize the F1-score, a critical parameter for correcting class imbalance. The ideal configuration was C=10 and γ=0.1, yielding an F1-score of 88% on the validation dataset. This method adeptly matched accuracy, reducing false positives, and recall, reducing false negatives, thereby guaranteeing dependable diagnosis of student deficiencies without overfitting.

- **Kernel selection**

According to Table 6, three kernel types were tested to determine the best fit for modeling interactions between behavioral and linguistic features:

Table 6: Kernels comparison

Kernel	Avg. F1-Score	Standard Deviation
Linear	0.79	±0.03
Polynomial	0.83	±0.02
RBF	0.88	±0.01

i. **Linear Kernel:**
   - Assumes a linear relationship between features (e.g., direct proportionality between ResponseTime and Weakness).
   - Achieved 79% accuracy but failed to capture non-linear patterns (e.g., the combined effect of high AttemptsTimes and medium WordDifficulty).

ii. **Polynomial Kernel:**
   - Models' polynomial relationships (e.g., quadratic effects of #Attempts).
   - Tested with degree d=3 and coefficient r=0, achieving 83% accuracy. However, it was computationally expensive and prone to overfitting.

iii. **Radial Basis Function (RBF) Kernel:**
   - Outperformed others with 88% accuracy, excelling at detecting nuanced patterns like:
   - High AttemptsTimes + Incorrect Answer → Strong predictor of weakness.
   - Long ResponseTime + High WordDifficulty → Weakness only if Answer was incorrect.

The RBF kernel's flexibility in modeling localized patterns (e.g., clusters of students with similar behavioral traits) and its robustness to feature scaling made it ideal for the dataset. Unlike the polynomial kernel, RBF avoided overfitting while capturing interactions between features like Attempts Times and Word Difficulty, which were critical for identifying knowledge gaps.

- **Cross-validation performance**

The final model was validated using stratified 5-fold cross-validation as shown in table 7:

Table 7: Final SVM Model Cross-Validation Performance

Fold	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
1	88.7	85.9	90.8	88.3
2	89.2	86.5	91.4	88.7
3	88.1	85.2	90.4	87.7
4	88.8	86.3	90.7	87.4
5	89.5	88.35	91	89.66
Average	**88.86**	**86.45**	**90.86**	**88.35**

- **Final model setup**

The optimized SVM configuration was:

SVC (kernel='rbf', C=10, gamma=0.1, class_weight='balanced')

The metrics indicate a well-optimized model that effectively identifies vocabulary weaknesses with high precision and recall, while sustaining robust overall accuracy. The F1-score integrates these priorities, providing actionable insights for educators. A student who hesitates twice on a medium-difficulty word such as "İnşaat" (Work/Occupation) will be classified as "weak" with high confidence, prompting the provision of customized practice exercises. This performance confirms the SVM configuration (RBF kernel, C=10, γ=0.1) and the feature engineering strategy, which prioritized behavioral attributes such as Attempts and response time, in addition to linguistic factors like word difficulty.

2.9 Model integrating into learning system

Over the course of this phase, we included the model we had generated during the preparatory step into the system. As the student advances through each level and overcomes new problems, we will construct a new model and enhance the one that is already in place. After then, this updated model will be applied at the next and more advanced level. With this iterative approach, the data from the students' tests should be routinely updated. This will enable us to identify the areas of categorization where the learner lacks competence and thereafter enhance the activities in those spheres.

This stage of the methodology involves the integration and refinement of the Machine Learning (ML) model, focusing on enhancing its performance and adaptability following initial development and training. The objective is to incorporate the machine learning model into the broader system or workflow and enhance it to improve accuracy, ensure robustness, and effectively manage a variety of real-world scenarios.

Upon integration of the model into the backend system, it is deployed for real-time predictions. In various scenarios, including game-based applications, the integration of the model as a REST API enables the gamification system to transmit input data and obtain predictions through HTTP requests. This facilitates efficient communication between the game interface and the machine learning model operating in the backend, ensuring seamless data flow and interaction.

In this final step, we are not concerned with the use of a specific system, and this system will not be discussed in this study. It may be any educational platform that utilizes a RESTful API to achieve the same objective deploying the trained model based on previously collected user behavioral data to identify weaknesses across word category classifications.

Refining an ML model starts with evaluating its performance using key metrics such as accuracy, precision, recall, and F1-score, as discussed previously. Understanding the model's strengths and limitations might help identify opportunities for development. Metrics can also be assessed for different subsets of data to determine if the model generalizes well or whether it performs poorly in certain contexts.

3 Discussion

With an accuracy of 89%, precision of 86%, recall of 91%, and an F1-score of 88%, the development and evaluation of the SVM-based model for spotting vocabulary shortcomings in Turkish language learners yielded quite impressive results. These measures highlight the resilience of the model and it's fit for systems of adaptive learning. More importantly, the addition of behavioral modeling, especially the integration of response time, number of attempts, and answer correctness significantly improved the predictive power of the model compared to conventional models that depend just on correctness.

Traditional vocabulary evaluation models typically treat learner performance as binary (correct vs. incorrect), which limits their capacity to detect uncertainty, hesitation, or overthinking behaviors that are pedagogically significant. In contrast, our behavioral modeling approach captured metacognitive indicators, allowing the model to recognize nuanced patterns such as students repeatedly hesitating on a moderately difficult word or taking excessive time to answer despite answering correctly. These behaviors were strong predictors of underlying weakness that binary-correctness models would miss. For example, the model successfully flagged students struggling with medium-difficulty terms like "İnşaat" based on both prolonged response times and multiple attempts an insight unattainable with traditional correctness-only metrics.

Moreover, in our permutation-based feature importance analysis, behavioral traits revealed great predictive value. Shuffling "correctness" and "number of attempts" resulted in the biggest F1-score drop, so attesting their indispensable importance. This shows that the decision-making process of the model gains orthogonal, non-redundant value from behavioral characteristics.

From a technical standpoint, the RBF kernel proved particularly effective for capturing these complex interactions. Empirical results showed that it outperformed linear and polynomial alternatives, achieving an F1-score of 88% with the lowest variance across cross-validation folds. Unlike linear kernels, which assume additive relationships among features, the RBF kernel models localized and non-linear patterns. This capacity was especially important in educational contexts where interactions between features such as how time spent, and correctness vary across word difficulty are rarely linear or independent. The polynomial kernel showed modest improvements over the linear kernel but suffered from overfitting and computational inefficiency. The RBF kernel struck the best balance between accuracy, generalization, and interpretability (with the aid of permutation-based feature analysis).

While this study focuses on Turkish vocabulary acquisition, the proposed framework holds potential for generalization across other languages and educational domains. The core components—behavioral data collection (e.g., response time, number of attempts), equation-based labeling, and SVM-based classification—are language-independent in structure and can be adapted to other linguistic contexts with appropriate adjustments to lexical difficulty metrics and thematic categorizations. Moreover, the methodology is domain-agnostic and can be extended beyond vocabulary learning to areas such as reading comprehension, grammar exercises, or even non-linguistic skill assessments, where learner behavior provides meaningful indicators of understanding or uncertainty. Future work could explore cross-linguistic validation by applying the framework to datasets in other languages or extend its use to interdisciplinary learning platforms where adaptive feedback based on learner behavior is critical for personalized instruction.

3.1 Limitations

Despite its advantages, the study has limitations. First and foremost, with 1,000 components in the collection (equivalent to 20 students), scalability becomes a question mark. The model worked well in our sample; however, its application to larger and more diverse groups such as youngsters or students learning several languages has not been evaluated.

Second, real-world apps may have different UI designs or network issues, so data collection assumptions, such as measuring attempts counts precisely, might not be applicable. For instance, if the interface is slow, it can make response times seem longer than they are, which would be misleading. Third, the study's focus on adults may limit its ability to generalize to younger groups, whose members may exhibit distinct characteristics (such as impulsivity or shorter attention spans.

Furthermore, introducing possible bias is the model relied solely on objective difficulty measures (e.g., syllabic complexity), and did not incorporate subjective learner perceptions of difficulty, which may limit the personalization potential of the model. Future work should consider integrating real-time perceived difficulty feedback to complement phonetic classification. Linguistic specialists defined word difficulty; but subjective learner perceptions that is, a student finding "Ev" difficult because of personal experience—were not adequately recorded. Dynamic difficulty changes based on individual performance could be included in the next versions.

4 Conclusion

In summary, our approach shows that combining behavioral analytics and machine learning may effectively identify language learning deficits, providing a scalable solution for individualized education. It achieves the dual aims of efficient resource allocation and comprehensive student support by putting precision and recall first. Future work to improve data gathering procedures and expand validation cohorts will increase its usefulness, opening the path for more adaptable, responsive learning technologies.

References

[1] Albrecht, C. M., Marianno, F., & Klein, L. J. (2021). *Autogeolabel: Automated label generation for geospatial machine learning.* Paper presented at the 2021 IEEE International Conference on Big Data (Big Data). https://doi.org/10.1109/bigdata52589.2021.9672060

[2] Bobák, P., Čmolík, L., & Čadík, M. (2023). Reinforced Labels: Multi-agent deep reinforcement learning for point-feature label placement. *IEEE Transactions on Visualization Computer Graphics, 30*(9), 5908-5922. https://doi.org/10.1109/tvcg.2023.3313729

[3] Schmarje, L., Grossmann, V., Michels, T., Nazarenus, J., Santarossa, M., Zelenka, C., & Koch, R. (2023). *Label Smarter, Not Harder: CleverLabel for Faster Annotation of Ambiguous Image Classification with Higher Quality.* Paper presented at the DAGM German Conference on Pattern Recognition. https://doi.org/10.1007/978-3-031-54605-1_30

[4] Zhang, F., Zhou, S., Wang, Y., Wang, X., & Hou, Y. (2024). Label assignment matters: A gaussian assignment strategy for tiny object detection. *IEEE Transactions on Geoscience Remote Sensing*. https://doi.org/10.1109/tgrs.2024.3430071

[5] Zhang, S., Jafari, O., & Nagarkar, P. (2021). A survey on machine learning techniques for auto labeling of video, audio, and text data. *arXiv preprint arXiv, 03784. https://doi.org/10.48550/arXiv.2109.03784*

[6] Stember, J. N., & Shalu, H. (2022). Deep reinforcement learning with automated label extraction from clinical reports accurately classifies 3D MRI brain volumes. *Journal of digital imaging, 35*(5), 1143-1152. https://doi.org/10.1007/s10278-022-00644-5

[7] Zhang, W., Wang, Y., & Wang, S. (2022). Predicting academic performance using tree-based machine learning models: A case study of bachelor students in an engineering department in China. *Education Information Technologies, 27*(9), 13051-13066. https://doi.org/10.1007/s10639-022-11170-w

[8] H. Kaur, T. Kaur, R. Garg. A Prediction Model for Online Student Academic Performance Using Machine Learning. Informatica 47(1):1–12, 2023. https://doi.org/10.31449/inf.v47i1.4297

[9] R. Wang. Automatic Classification of Document Resources Based on Naive Bayesian Classification Algorithm. Informatica 46(3):373–382, 2022. https://doi.org/10.31449/inf.v46i3.3970

[10] Sulaiman, M., & Roy, K. (2022). Fair classification via transformer neural networks: Case study of an educational domain. *arXiv preprint arXiv, 01410*. https://doi.org/10.48550/arXiv.2206.01410

[11] Shin, J., & Park, J. (2021). Pedagogical Word Recommendation: A novel task and dataset on personalized vocabulary acquisition for L2 learners. *arXiv preprint arXiv:2112.13808*. https://doi.org/10.48550/arXiv.2112.13808

[12] Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical science, 17*(3), 235-255. https://doi.org/10.1214/ss/1042727940

[13] Kononenko, I. (2001). Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in medicine, 23*(1), 89-109. https://doi.org/10.1016/s0933-3657(01)00077-x

[14] van der Waa, J., Nieuwburg, E., Cremers, A., & Neerincx, M. (2021). Evaluating XAI: A comparison of rule-based and example-based explanations. *Artificial Intelligence in medicine, 291*, 103404. https://doi.org/10.1016/j.artint.2020.103404

[15] Baker, R. S., & Yacef, K. (2009). The state of educational data mining in 2009: A review and future visions. *Journal of educational data mining, 1*(1), 3-17. https://doi.org/10.5281/zenodo.3554657

[16] Cambria, E., & White, B. (2014). Jumping NLP curves: A review of natural language processing research. *IEEE Computational intelligence magazine, 9*(2), 48-57. https://doi.org/10.1109/mci.2014.2307227

[17] Berrar, D. (2019). Bayes' theorem and naive Bayes classifier. Encyclopedia of Bioinformatics and Computational Biology, 1, 403-412. In. https://doi.org/10.1016/b978-0-12-809633-8.20473-1

[18] Karatzoglou, A., Meyer, D., & Hornik, K. (2006). Support vector machines in R. *Journal of statistical software, 15*, 1-28. https://doi.org/10.18637/jss.v015.i09

[19] Kramer, O. (2013). *Dimensionality reduction with unsupervised nearest neighbors* (Vol. 51): Springer. https://doi.org/10.1007/978-3-642-38652-7

[20] Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2003). A practical guide to support vector classification. In: Taipei, Taiwan.

https://eecs.csuohio.edu/~sschung/DSA460/SVM_guide.pdf

[21] François, T. (2009). *Combining a statistical language model with logistic regression to predict the lexical and syntactic difficulty of texts for FFL.* Paper presented at the Proceedings of the Student Research Workshop at EACL 2009. https://doi.org/10.3115/1609179.1609182

[22] Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*: " O'Reilly Media, Inc.". https://doi.org/10.1007/s13246-020-00913-z

[23] PHP-ML. (2025). Retrieved from https://php-ml.readthedocs.io/en/latest/

[24] Bratko, I. (1997). Machine learning: Between accuracy and interpretability. In *Learning, networks and statistics* (pp. 163-177): Springer. https://doi.org/10.1007/978-3-7091-2668-4_10

[25] Michaud, E. J., Liu, Z., & Tegmark, M. (2023). Precision machine learning. *Entropy*, *25*(1), 175. https://doi.org/10.3390/e25010175

[26] Simon, L., Webster, R., & Rabin, J. (2019). Revisiting precision and recall definition for generative model evaluation. *arXiv preprint arXiv, 05441*. https://doi.org/10.48550/arXiv.1905.05441

[27] Diallo, R., Edalo, C., & Awe, O. O. (2024). Machine Learning Evaluation of Imbalanced Health Data: A Comparative Analysis of Balanced Accuracy, MCC, and F1 Score. In *Practical Statistical Learning and Data Science Methods: Case Studies from LISA 2020 Global Network, USA* (pp. 283-312): Springer. https://doi.org/10.1007/978-3-031-72215-8_12

Appendix (A) Sensitivity Analysis of Weakness Score Equation (Eq. 1)

We performed a sensitivity analysis by varying the weights assigned to the behavioral and linguistic parameters: correctness of the answer (wCorrect), response time (wTime), hesitation count (wAttempts), and word difficulty (wDifficulty), so validating the robustness and statistical grounding of the rule-based weakness score equation (Equation1). The objective was to investigate how these weights affected SVM model downstream classification performance.

Each weight was varied within a range of ±10%, ±20%, and ±30% relative to its baseline configuration (wCorrect = 0.4, wTime = 0.2, wAttempts = 0.3, wDifficulty = 0.1), while ensuring that the total sum of weights remained equal to 1. The corresponding weakness labels were recalculated for each configuration and used to train separate SVM classifiers.

Stratified 5-fold cross-valuation was used in evaluation of the models. The results showed that only modest weight fluctuations would cause just slight changes in model performance. As Table A.1 shows, the F1-scores stayed within a limited band of ±2% from the baseline. Especially, the classification performance was more sensitive to variations in wCorrect and wAttempts, so verifying their major contribution in pointing up learning shortcomings.

Table A.1: Weight Variations on SVM F1-Score

Weight Configuration (wCorrect, wTime, wAttempts**, wDifficulty)**	**F1-Score (%)**
(0.4, 0.2, 0.3, 0.1) [Baseline]	88.0
(0.5, 0.15, 0.25, 0.1)	87.5
(0.3, 0.25, 0.35, 0.1)	88.1
(0.4, 0.3, 0.2, 0.1)	87.6
(0.35, 0.25, 0.3, 0.1)	87.9

These findings imply that, despite empirical construction, the rule-based equation shows enough stability and predictive dependability over a spectrum of weight settings. This justifies its use as a workable pre-labeling system in situations where ground truth labeled by humans is not available.

Appendix (B) Bootstrap Resampling for Robustness Evaluation

By using 100 iterations of bootstrapping resampling to assess the SVM model's stability and resilience despite a small dataset. Training data was sampled with replacement from the original 1,000 interactions in every iteration; the SVM model (RBF kernel, C=10, γ=0.1) was trained and tested using stratified 5-fold cross-validation.

The results, summarized in Table B.1, show consistently high performance across bootstrap samples. The F1-score averaged 88.2%, with a standard deviation of ±1.6, confirming the model's resilience to training data variability and supporting its applicability even under small sample regimes.

This procedure strengthens the statistical credibility of our findings and demonstrates that the proposed system maintains its effectiveness across data permutations.

Table B.1: SVM Model Performance Across 100 Bootstrap Resampling Iterations

Metric	**Mean (%)**	**Std. Dev. (%)**	**Min (%)**	**Max (%)**
Accuracy	88.9	±1.5	85.4	91.3
Precision	86.4	±1.8	82.6	89.7
Recall	90.5	±1.3	87.3	92.6
F1-Score	88.2	±1.6	85.0	90.8

Appendix C – Implementation and Reproducibility Guidelines Using PHP-ML

We provide below a thorough description of the preprocessing and model training pipeline applied using the PHP-ML library to support reproducibility and help adoption of methodology in related educational technology research. These processes are meant to guarantee constant performance and replicability in several surroundings.

1. **Data Preprocessing Workflow**

The raw dataset consisted of behavioral and linguistic features, which underwent the following transformation steps prior to model training:

- **Categorical Encoding:**
  - Word Classification: Each thematic category (e.g., Education, Family) was mapped to a unique integer.
  - Answer Encoding: Correct answers were encoded as 1, and incorrect as 0.
  - Difficulty Level: Difficulty categories were numerically mapped as Easy = 1, Medium = 2, Hard = 3.
- **Numerical Feature Handling:**

Retained as constant values are response time and attempts count. Min-Max Scaling brought all numerical features into the [0, 1] range to guarantee fit with SVM kernel behavior.

2. **Support Vector Machine Configuration**

Model training was carried out using the Support Vector Classification (SVC) class provided in PHP-ML. Key configuration details are as follows:

- **Library Version:** PHP-ML v0.10+
- **PHP Version:** >= 7.1
- **Classifier:** Phpml\Classification\SVC
- **Kernel Type:** Radial Basis Function (RBF)
- **Class Weighting:** Balanced, to account for mild class imbalance (weak vs. non-weak labels)
- **Features Used:** Encoded class, difficulty, response time, correctness, and number of attempts

3. Hyperparameter Optimization (Grid Search)

To optimize model performance and reduce overfitting, we employed a grid search approach over the following hyperparameter ranges mentioned in Table C.1.

Table C.1: C and Gama search values

Parameter	Tested Values
C	0.1, 1, 10, 100
gamma	0.001, 0.01, 0.1, 1

- Each parameter combination was evaluated using stratified 5-fold cross-validation.
- The F1-score was used as the primary evaluation metric to ensure balanced consideration of precision and recall.

The optimal configuration selected was C = 10, gamma = 0.1, yielding the highest cross-validated F1-score.

4. Pseudo-Code for Reproduction Using PHP-ML

The following pseudo-code outlines the key steps in training the SVM model:

```
use Phpml\Classification\SVC;
use Phpml\SupportVectorMachine\Kernel;
use Phpml\CrossValidation\StratifiedRandomSplit;
use Phpml\ModelManager;

// Step 1: Prepare input data
$X = [...]; // Feature vectors (normalized)
$y = [...]; // Labels (0 = no weakness, 1 = weakness)

// Step 2: Initialize classifier
$classifier = new SVC(
    Kernel::RBF,   // Kernel type
    $cost = 10,    // Regularization parameter C
    $gamma = 0.1,  // RBF kernel width
    $enableProbabilityEstimates = true
);

// Step 3: Train the model
$classifier->train($X, $y);

// Step 4 (optional): Save the trained model
$modelManager = new ModelManager();
$modelManager->saveToFile($classifier, 'svm_model.phpml');
```