Classification of Algebric Students Solutions Using Transformer Based Embeddings

Chunduri Yoshitha
Dept. of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4aie23003@bl.en.students.amrita.edu

G Vishwa Prakashini
Dept. of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4aie23007@bl.en.students.amrita.edu

Neha S
Dept. of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4aie23022@bl.en.students.amrita.edu

Roshni M Balakrishnan
Dept. of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
m_roshni@blr.amrita.edu

Peeta Basa Pati*
Dept. of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
pb_pati@blr.amrita.edu

***Abstract*—This project aims to develop a machine learning based classification system for automatically grading algebraic student solutions into three categories: "Incorrect", "Partially Correct", and "Correct". The system leverages embeddings from transformer-based language models, such as GPT-2, BART, MathBERT, and RoBERTa, to capture the semantic features of student responses. Pretrained and fine-tuned embeddings of the four transformer models are extracted from each model and used with traditional multi-class classifiers. The models are evaluated using cross-validation, and the best ones are improved further using feature selection. The results show that fine-tuned MathBERT performs well, with RoBERTa close behind. Feature selections further improve the results, and the best accuracy of 81.60% is achieved. This demonstrates the system's strong potential for use in real educational settings.**

***Index Terms*—Algebraic Solutions, Machine Learning, Mathematics, GPT-2, BART, MathBERT, RoBERTa, Embeddings, Automatic Evaluation**

I. Introduction

Evaluating student-written solutions to algebraic problems, particularly quadratic equations, is a fundamental yet complex task in mathematics education. These problems often involve multi-step reasoning, calculation, and varied solution paths, therefore making manual grading labour-intensive, subjective, and sometimes inconsistent.

This issue is especially problematic in digital and large-scale learning environments, where timely and fair assessment is critical. It also highlights the urgent need for automated systems that can assess students' work reliably and efficiently because inaccurate or delayed feedback can hinder learning and burden educators.

To solve this problem, we propose a machine learning based classification system that categorizes algebraic student responses into three meaningful classes: Incorrect, Partially Correct, and Correct. Our approach leverages advanced natural language processing models such as GPT-2, BART, MathBERT, and RoBERTa to extract pre-trained and fine-tuned embeddings, which are then used with traditional classifiers to predict the correctness level of each solution.

Fine-tuning the language models on our domain specific data significantly enhances their understanding of algebraic reasoning. We applied feature selection techniques on top performing models like MathBERT and RoBERTa to further improve performance.

A key strength of this system lies in its support for teachers. By automatically distinguishing between fully incorrect and partially correct answers, it enables educators to give more focused attention to students who need conceptual guidance, while those making minor errors can be supported with targeted feedback. This not only saves time but also fosters more personalized learning experiences.

The novelty of this work lies in the integration of fine-tuned mathematical language models with traditional machine learning classifiers, enhanced through systematic feature selection techniques. These methods are applied specifically to the domain of algebraic assessment. Our best-performing model, using feature selection on fine-tuned MathBERT embeddings with the machine learning classifier, achieves an accuracy of approximately 81.60%.

This work contributes to UN Sustainable Development Goal 4 (Quality Education) by promoting accessible, scalable, and intelligent tools for student evaluation. The remainder of the paper is structured as follows: Section 2 covers related work, Section 3 explains the dataset and methodology, Section 4 presents experiments and results, Section 5 discusses interpretability and analysis, and Section 6 concluded with future directions.

II. Literature Survey

In [1], Shen et al. developed MathBERT, a pre-trained language model designed specifically for mathematics education. By training on a large corpus of mathematical texts, MathBERT outperformed general-purpose models in tasks like knowledge component prediction and auto-grading. This supports our decision to explore MathBERT embeddings for accurately representing and classifying algebraic solutions.

In [2], Zhang et al. proposed an in context meta learning approach for automatic short math answer grading. Utilizing MathBERT, a BERT variant tailored for mathematical content, their model demonstrated improved generalization to unseen questions. This demonstrates the strength of domain-specific models like MathBERT, aligning with our findings where fine-tuned MathBERT outperformed other embeddings.

In [3], Haller et al. provided a comprehensive survey on automated short answer grading (ASAG) with deep learning, tracing the evolution from traditional word embeddings to transformer-based models. The study emphasized that combining learned representations with hand-engineered features yields superior performance in grading tasks. This insight is valuable for developing hybrid models that leverage both semantic understanding and specific feature engineering for classifying mathematical solutions.

In [4], Romero and Ventura offered an updated survey on educational data mining and learning analytics, discussing the application of data-driven techniques in educational settings. The paper highlighted the significance of automated assessment tools in providing timely feedback and enhancing learning outcomes. Their work supports the integration of machine learning models, like those using MathBERT and RoBERTa embeddings, into educational platforms for efficient assessment processes.

In [5], Rao et al. presented a system for the automatic assessment of quadratic equation solutions utilizing MathBERT and RoBERTa embeddings. Their approach involved fine-tuning these models to classify student responses into categories such as correct, partially correct, and incorrect. The study demonstrated that MathBERT embeddings outperformed RoBERTa in terms of accuracy, highlighting the effectiveness of domain-specific language models in educational assessments.

In [6], Injeti et al. conducted a comparative study on the classification of students' algebraic responses using MathBERT embeddings. The research focused on evaluating the performance of various machine learning classifiers when combined with MathBERT representations. Results indicated that integrating MathBERT with classifiers like Support Vector Machines (SVM) and Multi-Layer Perceptron (MLP) enhanced the accuracy of classifying algebraic solutions, emphasizing the synergy between advanced embeddings and traditional classifiers.

In [7], Balakrishnan et al. investigated the use of a fine-tuned T5 transformer model for auto-grading quadratic equation problems. The research highlighted the model's capability to understand and evaluate mathematical expressions, providing accurate grading outcomes. This work underscores the potential of transformer-based models in handling complex educational tasks, such as assessing algebraic solutions.

In [8], Narmada et al. explored the evaluation of uniXcoder embeddings for automated grading across varied code perspectives. Although centred on programming assignments, the study's insights into embedding effectiveness and grading accuracy are pertinent to mathematical solution assessment. The findings suggest that selecting appropriate embeddings tailored to the content type is crucial for improving automated grading systems.

In [9], Gaddipati et al. evaluate the effectiveness of pre-trained models, including GPT-2, BERT, and ELMo, in automatic short answer grading tasks. Using cosine similarity as a feature, the models were assessed on the Mohler dataset. While ELMo outperformed others, GPT-2 demonstrated potential in semantic understanding for grading purposes.

In [10], Savelka et al. investigate the capabilities of GPT models, including GPT-2, in completing assessments from introductory and intermediate Python programming courses. The findings reveal that while GPT models can achieve a significant portion of the overall score, they struggle with complex reasoning tasks, highlighting both their potential and limitations in educational settings.

Recent advances in natural language processing and educational AI have led to the development of domain-specific models like MathBERT, which has demonstrated superior performance over general-purpose models in tasks such as knowledge component prediction and auto-grading [1,2]. Studies have shown that combining transformer-based embeddings with traditional classifiers, such as SVM and MLP, significantly enhances the accuracy of classifying algebraic solutions [6]. Surveys by Haller et al. and Romero and Ventura emphasize the growing importance of integrating deep learning and educational data mining to support automated assessment and timely feedback in learning environments [3,4]. Several works specifically explore fine-tuned MathBERT and RoBERTa embeddings for classifying quadratic equation responses, with MathBERT consistently outperforming due to its mathematical context understanding [5,6]. Other models like T5 and uniXcoder, also show promise in auto-grading tasks, demonstrating the adaptability of transformer architecture to varied educational content [7,8]. Research into GPT-2 reveals its potential in short answer grading and programming assessment, it faces challenges in complex reasoning scenarios [9,10]. Collectively, these studies highlight the effectiveness of tailored embeddings in improving grading accuracy and educational support systems.

III. Methodology

Classifying student-written solutions to quadratic equations based on their accuracy has significant implications for both mathematical understanding and educational evaluation. To address this challenge, this study proposes a structured approach for the automatic classification of algebraic solutions. RQ1: Do fine-tuned embeddings perform better in classification than pretrained embeddings? RQ2: Does feature selection improve the performance of top-performing models? A multi-phase pipeline was designed. The complete workflow is shown in Fig. 1, capturing all stages from raw data preparation to final classification performance evaluation.

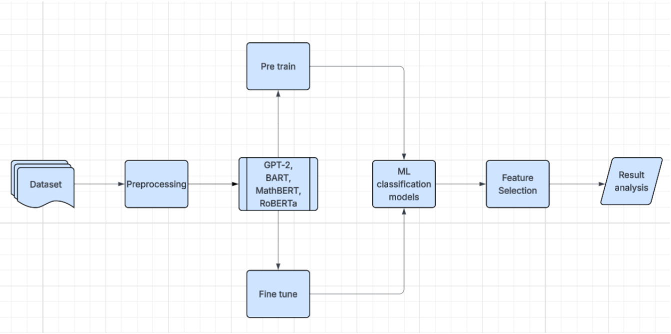

Fig. 1. Flow of Work.

A. Dataset Description

The dataset used consists of 1,860 student-written solutions to problems involving quadratic equations. Each response was labeled into one of three categories: Correct, Partially Correct, or Incorrect. An initial analysis showed significant class imbalance, with a higher number of incorrect solutions compared to correct or partially correct ones. This imbalance is visualized in Fig. 2. Recognizing this issue was important, as it can bias learning algorithms towards the majority class and distort evaluation metrics. Techniques like stratified splitting during cross-validation were used to mitigate this effect in later stages. Therefore, SMOTE has been applied for ML model classification.

B. Data Pre-processing

Data preprocessing was performed to clean and prepare the dataset for subsequent feature extraction and model training. The process began with the removal of duplicate entries to eliminate redundancy and prevent bias in the learning process. Following this, the dataset was thoroughly examined for any null or missing values, and appropriate measures such as imputation or removal were applied to maintain the completeness and consistency of the data. Additionally, text normalization was carried out to ensure uniformity across all responses. This involved standardizing whitespace, character encoding, and special symbols, while carefully preserving the mathematical expressions and notation essential for understanding student solutions. These preprocessing steps laid a strong foundation for generating high-quality embeddings and applying machine learning algorithms effectively.

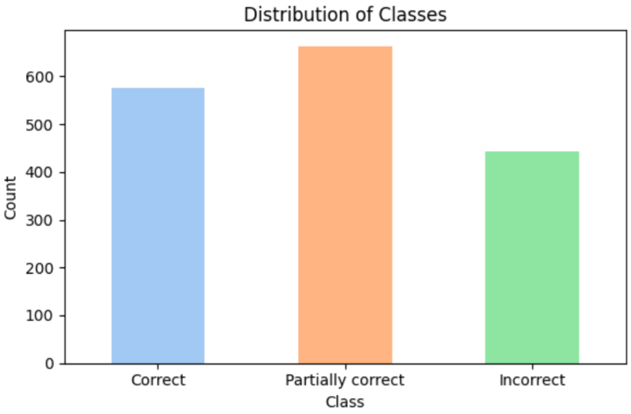

Fig. 2. Class Distribution.

C. Feature Extraction

To convert the textual mathematical solutions into numerical representations suitable for machine learning, contextual embeddings were generated using four transformer-based models: MathBERT, RoBERTa, BART, and GPT-2. For each of these models, both pretrained and fine-tuned embeddings were extracted. The fine-tuning process involved adapting the models to the specific classification task using the labeled dataset, allowing them to better capture the semantic intricacies of mathematical language and student expression. Each student response was transformed into a dense feature vector using the respective model.

MathBERT, BART, RoBERTa, and GPT-2 produced embeddings of size 768. Among these, MathBERT and RoBERTa were identified as the top-performing models. Their embeddings were used to create a feature matrix of shape 1860×768, representing 1,860 student responses each encoded into a 768-dimensional vector. These high-dimensional, context-rich embeddings served as the input features for the machine learning classifiers in the next stage of the pipeline.

D. ML Classifiers

To evaluate classification performance across different embeddings, nine supervised ML algorithms were employed as shown in Table I.

Each model underwent hyperparameter tuning using RandomizedSearchCV, and performance was evaluated using 10-fold cross-validation. This allowed consistent and fair comparison across all embeddings and classifiers, as well as ensured generalization across different subsets of data. Standard classification metrics such as accuracy, and F1-score were computed for each fold. Additionally, standard deviation of accuracy across folds was recorded to assess model stability.

E. Feature Selection

To improve model performance and reduce overfitting, feature selection was applied to the fine-tuned embeddings of

TABLE I
MODELS AND HYPERPARAMETERS

Model	Hyperparameters
SVM	{C: 1, kernel: 'rbf', gamma: 'scale', probability: True, class_weight: 'balanced', random_state: 42}
KNN	{n_neighbors: 7, metric: 'minkowski', p: 2}
Decision Tree	{max_depth: 10, min_samples_split: 5, class_weight: 'balanced', random_state: 42}
Naïve Bayes	{var_smoothing: 1e-9}
AdaBoost	{n_estimators: 100, learning_rate: 0.1, random_state: 42}
XGBoost	{learning_rate: 0.05, max_depth: 8, n_estimators: 150, subsample: 0.8, colsample_bytree: 0.8, use_label_encoder: False, eval_metric: 'mlogloss', random_state: 42}
MLP	{hidden_layer_sizes: (100, 50), activation: 'relu', alpha: 0.0001, max_iter: 300, random_state: 42}
RF	{n_estimators: 200, max_depth: 15, min_samples_split: 5, class_weight: 'balanced', random_state: 42}
Logistic Regression	{C: 1.0, penalty: 'l2', solver: 'liblinear', class_weight: 'balanced', max_iter: 1000, random_state: 42}

the top two models—MathBERT and RoBERTa. Each model initially produced embeddings of size 768 for 1,860 student responses. A tree-based selection approach using SelectFromModel with a Random Forest classifier was used to retain only the most relevant features.

This method identified features with importance above the average threshold, leading to a refined feature space that maintained the original 1860 × 768 shape but with enhanced focus on informative dimensions. Feature selection was integrated into a 10-fold stratified cross-validation setup, where SMOTE was first applied to balance classes before selecting features.

IV. RESULT AND ANALYSIS

This section is the most important part of the study conducted in concluding important results. All the graphical representations are present in this section.

The bar chart in Fig. 3,compares the accuracy of various machine learning models using pretrained and finetuned GPT-2. Finetuned GPT-2 consistently improves accuracy across models, with XGBoost increasing from 58% to 64%, SVM from 55% to 61%, and MLP from 57% to 63%. Smaller gains are observed in Naive Bayes from 50% to 52% and Decision Tree from 53% to 56%. Error bars represent standard deviation, indicating variability in accuracy measurements and reinforcing the reliability of the results.

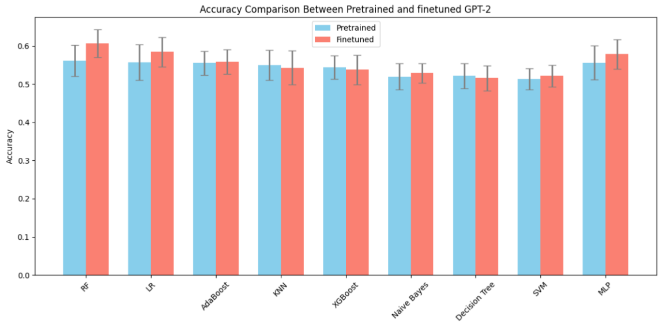

Fig. 3. Accuracy comparison between pretrained and finetuned GPT-2.

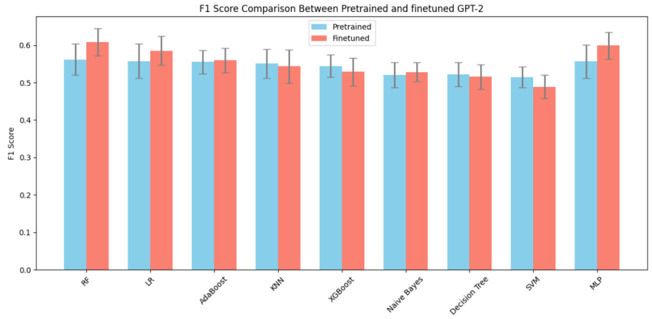

Fig. 4. F1 score comparison between pretrained and finetuned GPT-2.

The bar chart in Fig. 4,compares the F1 scores of various machine learning models using pretrained and finetuned GPT-2. Finetuned GPT-2 consistently improves performance across models, with XGBoost increasing from 0.48 to 0.56, SVM from 0.42 to 0.50, and MLP from 0.44 to 0.52. Smaller gains are observed in Naive Bayes from 0.30 to 0.32 and Decision Tree from 0.35 to 0.38. Error bars indicate variability in F1 scores, reinforcing the reliability of the results. The bar chart in Fig.5, illustrates the accuracy of various machine learning models using pretrained and finetuned BART. Finetuned BART generally improves performance across models, with SVM increasing from 0.55 to 0.60, while models like RF, LR, AdaBoost, KNN, XGBoost, and MLP remain around 0.55 for both pretrained and finetuned versions. Smaller gains are observed in Naive Bayes and Decision Tree, both increasing slightly from 0.50 to 0.50. The bar chart in Fig. 6, compares the F1 scores of various machine learning models using pretrained and finetuned BART. Finetuned BART generally improves performance across models, with SVM increasing from 0.55 to 0.60, while models like RF, LR, AdaBoost, KNN, XGBoost, and MLP remain around 0.55 for both pretrained

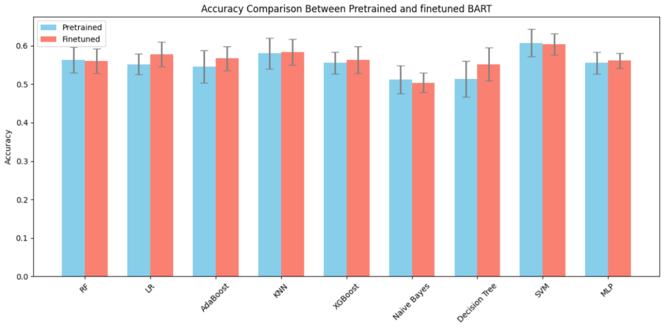

Fig. 5. Accuracy score comparison between pretrained and finetuned BART.

and finetuned versions. Smaller gains are observed in Naive Bayes and Decision Tree, both increasing slightly from 0.50 to 0.50.

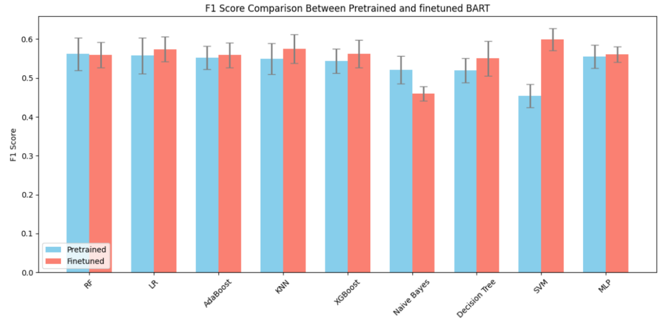

Fig. 6. F1 score comparison between pretrained and finetuned BART.

TABLE II. Mean accuracy and mean F1 scores shown by different ML classifiers in 10 folds for Pre-Trained MathBERT.

Models	Accuracy		F1-Score	
	Mean	STD	Mean	STD
SVM	0.6226	0.0307	0.6196	0.0306
MLP	0.5732	0.0321	0.5730	0.0311
Random Forest	0.5738	0.0298	0.5720	0.0305
XGBoost	0.5702	0.0320	0.5690	0.0330
KNN	0.5631	0.0526	0.5490	0.0552
Logistic Regression	0.5464	0.0251	0.5453	0.0255
AdaBoost	0.5423	0.0200	0.5434	0.0207
Naive Bayes	0.5173	0.0408	0.5166	0.0409
Decision Tree	0.5107	0.0309	0.5061	0.0311

From Table II, and III, it is clear that fine-tuning MathBERT significantly improves the accuracy and F1 scores of all machine learning classifiers compared to the pre-trained version. The most notable improvements are seen in Naive Bayes and AdaBoost, with accuracy increasing from 0.5173 to 0.8149 and 0.5423 to 0.7940, respectively. SVM and XGBoost also show substantial gains, with SVM improving from 0.6226 to 0.7577 and XGBoost from 0.5702 to 0.6970 in accuracy. The standard deviation remains relatively stable across models, indicating consistent performance improvements without introducing excessive variability. While fine-tuning benefits all classifiers, models like Decision Tree and Logistic Regression still perform lower than others, suggesting that more complex models like AdaBoost and Naive Bayes leverage fine-tuning more effectively.

TABLE III. Mean accuracy and mean F1 scores shown by different ML classifiers in 10 folds for Fine-Tuned MathBERT.

Models	Accuracy		F1-Score	
	Mean	STD	Mean	STD
Naive Bayes	0.8149	0.0209	0.8144	0.0209
AdaBoost	0.7940	0.0292	0.7922	0.0286
KNN	0.7661	0.0274	0.7621	0.0280
SVM	0.7577	0.0355	0.7557	0.0347
Random Forest	0.7006	0.0318	0.7000	0.0313
XGBoost	0.6970	0.0332	0.6965	0.0332
MLP	0.6923	0.0323	0.6916	0.0324
Logistic Regression	0.6881	0.0355	0.6874	0.0358
Decision Tree	0.6649	0.0339	0.6629	0.0342

TABLE IV. Mean accuracy and mean F1 scores shown by different ML classifiers in 10 folds for Pre-Trained RoBERTa.

Models	Accuracy		F1-Score	
	Mean	STD	Mean	STD
SVM	0.5917	0.0339	0.5888	0.0353
XGBoost	0.5536	0.0388	0.5537	0.0399
MLP	0.5524	0.0270	0.5521	0.0266
Random Forest	0.5488	0.0357	0.5483	0.0363
Logistic Regression	0.5298	0.0262	0.5286	0.0268
KNN	0.5137	0.0499	0.5002	0.0539
Decision Tree	0.4970	0.0317	0.4955	0.0319
AdaBoost	0.4762	0.0316	0.4663	0.0403
Naive Bayes	0.4304	0.0431	0.4272	0.0503

In Tables Iv, and V it is seen that fine-tuning RoBERTa significantly enhances the accuracy and F1 scores of all machine learning classifiers compared to the pre-trained version. The most notable improvements are observed in Naive Bayes and AdaBoost, with accuracy increasing from 0.4304 to 0.7875 and 0.4762 to 0.7655, respectively. SVM also shows substantial gains, improving from 0.5917 to 0.7601 in accuracy, while KNN sees a notable increase from 0.5137 to 0.7250. The standard deviation remains relatively stable across models, indicating consistent performance improvements without excessive variability. While fine-tuning benefits all classifiers, models like Decision Tree and Logistic Regression still perform lower than others, suggesting that more complex models like Naive Bayes and AdaBoost leverage fine-tuning more effectively.

TABLE V. Mean accuracy and mean F1 scores shown by different ML classifiers in 10 folds for Fine-Tuned RoBERTa.

Models	Accuracy		F1-Score	
	Mean	STD	Mean	STD
Naive Bayes	0.7875	0.0232	0.7849	0.0235
AdaBoost	0.7655	0.0250	0.7603	0.0258
SVM	0.7601	0.0203	0.7552	0.0217
KNN	0.7250	0.0258	0.7167	0.0274
Random Forest	0.6667	0.0270	0.6651	0.0280
XGBoost	0.6601	0.0297	0.6585	0.0301
MLP	0.6595	0.0285	0.6575	0.0291
Logistic Regression	0.6542	0.0256	0.6515	0.0258
Decision Tree	0.6417	0.0363	0.6383	0.0377

Since fine-tuned MathBERT and RoBERTa consistently outperformed their pre-trained counterparts across all classifiers, we proceed with feature selection to further optimize their performance. This step aims to refine the models by identifying the most relevant features, enhancing efficiency and accuracy. Therefore, our findings answer RQ1, confirming that fine-tuned embeddings perform better than pre-trained ones.

TABLE VI. Mean accuracy and mean F1 scores shown by different ML classifiers in 10 folds for Fine-Tuned MathBERT after feature selection.

Models	Accuracy		F1-Score	
	Mean	STD	Mean	STD
Naive Bayes	0.8161	0.0201	0.8139	0.0201
AdaBoost	0.7893	0.0223	0.7862	0.0218
SVM	0.7714	0.0289	0.7692	0.0291
KNN	0.7679	0.0235	0.7637	0.0239
Logistic Regression	0.7107	0.0301	0.7097	0.0297
Random Forest	0.6994	0.0338	0.6987	0.0337
XGBoost	0.6982	0.0366	0.6975	0.0366
MLP	0.6935	0.0296	0.6928	0.0299
Decision Tree	0.6601	0.0402	0.6581	0.0412

TABLE VII. Mean accuracy and mean F1 scores shown by different ML classifiers in 10 folds for Fine-Tuned RoBERTa after feature selection.

Models	Accuracy		F1-Score	
	Mean	STD	Mean	STD
Naive Bayes	0.7881	0.0253	0.7850	0.0254
AdaBoost	0.7708	0.0247	0.7662	0.0263
SVM	0.7631	0.0208	0.7580	0.0222
KNN	0.7304	0.0274	0.7227	0.0288
Logistic Regression	0.6964	0.0251	0.6939	0.0257
Random Forest	0.6738	0.0241	0.6724	0.0246
MLP	0.6690	0.0346	0.6662	0.0354
XGBoost	0.6619	0.0270	0.6604	0.0277
Decision Tree	0.6345	0.0349	0.6320	0.0354

From Tables VI,and VII, we interpret that feature selection further improves the performance of fine-tuned MathBERT and RoBERTa across most classifiers. In MathBERT, SVM increases from 0.7577 to 0.7714 in accuracy, while Naive Bayes remains the best-performing model, improving slightly from 0.8149 to 0.8161. Similarly, in RoBERTa, SVM improves from 0.7601 to 0.7631, and AdaBoost increases from 0.7655 to 0.7708, showing that feature selection enhances model efficiency. While some models like Random Forest and XGBoost show marginal improvements, the overall trend confirms that feature selection refines fine-tuned embeddings, leading to better classification performance.

Feature selection enhances the performance of fine-tuned MathBERT and RoBERTa across most classifiers, with models like SVM and AdaBoost showing noticeable improvements. Naive Bayes remains the best-performing model, maintaining high accuracy while benefiting from refined feature selection. While some models show marginal gains, the overall trend confirms that feature selection optimizes fine-tuned embeddings for better classification. Therefore, this answers RQ2, demonstrating that feature selection improves model performance.

V. Conclusion

This study presents a machine learning-based classification system for automatically grading algebraic student solutions into three categories: Incorrect, Partially Correct, and Correct. By leveraging embeddings from transformer-based language models such as GPT-2, BART, MathBERT, and RoBERTa, the system effectively captures the semantic features of student responses. Fine-tuned MathBERT achieves the highest accuracy of 81.60%, with RoBERTa closely following, demonstrating the effectiveness of fine-tuned embeddings over pre-trained ones. Feature selection further enhances model performance, refining predictions and improving classification accuracy. The system provides a scalable and efficient solution for automated grading, reducing the burden on educators while ensuring timely and fair assessments. Future work can explore domain adaptation techniques to improve generalization across different mathematical topics and expand the dataset to include diverse problem types. Additionally, integrating explainable AI techniques and multimodal approaches, such as incorporating handwritten responses, can enhance transparency and versatility. Further improvements in fine-tuning strategies and feature selection methods will optimize performance for large-scale educational settings, making automated grading more effective and accessible..

References

[1] Shen, Jia & Yamashita, Michiharu & Prihar, Ethan & Heffernan, Neil & Wu, Xintao & Lee, Dongwon. (2021). MathBERT: A Pre-trained Language Model for General NLP Tasks in Mathematics Education. 10.48550/arXiv.2106.07340.

[2] JZhang, Mengxue & Baral, Sami & Heffernan, Neil & Lan, Andrew. (2022). Automatic Short Math Answer Grading via In-context Meta-learning. 10.48550/arXiv.2205.15219.

[3] Haller, S., Aldea, A., Seifert, C., & Strisciuglio, N. (2022). Survey on Automated Short Answer Grading with Deep Learning: from Word Embeddings to Transformers. arXiv preprint arXiv:2204.03503.

[4] Romero, Cristóbal & Ventura, Sebastian. (2020). Educational Data Mining and Learning Analytics: An Updated Survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 10.1002/widm.1355.

[5] S. S. Rao, S. Mishra, S. Akhilesh, R. M. Balakrishnan and P. B. Pati, ”Automatic Assessment of Quadratic Equation Solutions Using MathBERT and RoBERTa Embeddings,” 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, pp. 1-7, doi: 10.1109/ICCCNT61001.2024.10724422.

[6] A. S. Injeti, G. N. Rupsica, G. P. Reddy, R. M. Balakrishnan and P. B. Pati, ”A Machine Learning Based Classification of Students’ Algebraic Responses Using MathBERT Embeddings,” 2024 5th International Conference for Emerging Technology (INCET), Belgaum, India, 2024, pp. 1-6, doi: 10.1109/INCET61516.2024.10593432.

[7] Balakrishnan, Roshni M., et al. ”Fine-tuned T5 for auto-grading of quadratic equation problems.” Procedia Computer Science 235 (2024): 2178-2186.

[8] N. Narmada and P. B. Pati, ”Evaluating uniXcoder Embeddings for Automated Grading: A Study Across Varied Code Perspectives,” 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, pp. 1-5, doi: 10.1109/ICCCNT61001.2024.10725702.

[9] Gaddipati, Sasi & Nair, Deebul & Plöger, Paul. (2020). Comparative Evaluation of Pretrained Transfer Learning Models on Automatic Short Answer Grading. 10.48550/arXiv.2009.01303.

[10] Savelka, Jaromir & Agarwal, Arav & Bogart, Christopher & Song, Yifan & Sakr, Majd. (2023). Can Generative Pre-trained Transformers (GPT) Pass Assessments in Higher Education Programming Courses?. 10.48550/arXiv.2303.09325.