LLM-Based Automatic Grading of Open-Ended Questions Using Human Curated Rubrics

Quah Jun Chuan[1], **Liyana Shuib**[1]
[1]*Faculty of Computer Science and Information Technology*
Universiti Malaya, Kuala Lumpur, 50603, Wilayah Persekutuan Kuala Lumpur, Malaysia
22004851@siswa.um.edu.my, liyanashuib@um.edu.my

***Abstract* — Manual grading of open-ended questions is a time-consuming and labour-intensive task. It is also prone to subjectivity and bias, even when graded using pre-defined rubrics, which often leads to inconsistent grading results. This study aims to develop a grading model that evaluates open-ended questions based on human-curated rubrics using a Large Language Model (LLM). The ASAP-SAS dataset from the Kaggle competition was selected to assess the model's grading capabilities. GPT-4o Mini was chosen as the LLM and was tested with various prompt engineering techniques, including zero-shot, chain-of-thought (CoT), and Reflexion. The model's performance was evaluated using five metrics: Accuracy, Precision, Recall, F1 Score, and Quadratic Weighted Kappa (QWK). The Chain-of-Thought prompting outperformed the others with scores of 0.763 (Accuracy), 0.764 (Precision), 0.766 (Recall), 0.763 (F1), and 0.871 (QWK). A Streamlit web application was deployed, integrating the LLM with the best-performing prompting technique for automated grading.**

***Keywords* — Automated Grading, Large Language Model (LLM), Open-Ended Questions, Human-Curated Rubrics, Prompt Engineering.**

I. Introduction

Evaluation of open-ended questions is a critical aspect of educational assessment as it allows educators to measure students' understanding, reasoning, and critical thinking skills. However, manual grading of these responses is often tiring and prone to inconsistencies due to subjective judgment.

Thanks to the developments in Artificial Intelligence (AI), automated grading systems have gained significant attention due to their ability to reduce the workload on educators while ensuring consistent assessment standards. Initially, these systems were primarily designed to evaluate objective-type questions (e.g., multiple-choice or true/false questions). However, recent studies have expanded their capabilities to include the assessment of open-ended questions, which traditionally required subjective human judgment (Gnanaprakasam and Lourdusamy, 2024). This shift is driven by the need to manage the growing volume of educational assessments and to provide timely feedback to learners.

Since 2020, Large Language Models (LLMs) have revolutionized the field of Natural Language Processing (NLP) by surpassing traditional models through their capacity to generate and comprehend human-like text. Unlike rule-based or statistical NLP models, LLMs utilize deep learning architectures trained on massive datasets to perform complex language-related tasks (Brown et al., 2020). This advancement makes LLMs ideal in handling the complexity of grading open-ended responses, where contextual understanding and reasoning are essential for accurate grading. Their ability to understand nuances in language and provide human-like reasoning allows them to offer more precise and consistent evaluations compared to traditional grading methods (Pinto et al., 2023).

Central to the fair evaluation of open-ended responses is the use of Human-Curated Rubrics, which provide structured scoring guidelines with clear evaluation criteria to ensure consistency and fairness in assessment (Stevens and Levi, 2012). Open-ended questions graded with well-defined rubrics allows a comprehensive evaluation of a student's depth of knowledge. However, manual grading using these rubrics can be inefficient and inconsistent, especially in large-scale educational settings.

Therefore, integrating LLMs with human-curated rubrics offers an innovative solution by combining the model's contextual understanding with standardized scoring criteria. This integration enables automated grading systems to deliver more accurate and scalable assessments while maintaining alignment with predefined grading standards (Yamamoto et al., 2017).

II. Literature Review

A. LLM-Based Automated Grading Without Prompt Engineering

Automated grading of short-answer questions using Large Language Models (LLMs) has gained significant attention due to its potential to reduce educators' workloads and provide scalable assessment solutions. This section reviews studies evaluating LLMs for automated short-answer grading without applying prompt engineering.

Meyer et al. (2024) introduced the ASAG2024 benchmark, a combined dataset of seven commonly used short-answer grading datasets to evaluate automated grading models. Models such as Llama3-8B, GPT-3.5-turbo, and GPT-4o were assessed without employing prompt engineering. GPT-

4o achieved the best performance with a weighted RMSE of 0.27; however, its grading error was still more than double that of human graders.

Schneider et al. (2024) evaluated ChatGPT (GPT-3.5) for grading short textual answers across two academic exams. The study aimed to assess the model's utility as a supportive tool for educators rather than a fully autonomous grading system. Findings indicated a low correlation between human and LLM-generated grades, reflecting poor alignment in grading standards. The model struggled to provide feedback and was sensitive to minor variations in input resulting in inconsistent grading.

Gobrecht et al. (2024) developed an automated open-ended question grading system using a fine-tuned German BERT (GBERT) model, trained on university exam data from IU International University of Applied Sciences. The model achieved a mean absolute error (MAE) of 1.32 and a Pearson correlation of 0.78 on unseen questions. Compared to human graders, it showed a 44% lower median absolute error and higher consistency (0.59 vs. 0.49). However, the study identified potential bias issues due to imbalanced dataset. To be specific, the model's struggle with grading high-point questions and a lack of explainability in grading decisions.

The GradeOpt framework by Chu et al. (2024) is a multi-agent automatic short answer grading (ASAG) framework that which incorporates three specialized agents: Rubric Parser for rubric generation, Answer Analyzer for analyzing answer text and Feedback Generator for providing constructive feedback. For evaluation, it achieved the highest average performance with an Accuracy of 0.85 and a Quadratic Weighted Kappa (QWK) of 0.73 which outperformed all baseline LLM models such as RoBERTa, SBERT, GPT-4o, and APO.. Despite its strong performance, a key limitation of the system is its reliance on well-structured rubrics where poorly defined rubrics may reduce grading accuracy.

Similarly, Kortemeyer (2023) evaluates how well GPT-4 performs in automated open-ended question grading. The research uses the SciEntsBank dataset (science questions for grades 3–6) and the Beetle dataset (basic electricity and electronics questions). GPT-4 was tested under three scenarios: using reference answers (2-way and 3-way classification) and without reference answers. GPT-4 scored highest on SciEntsBank's 2-way task (F1 = 0.744) and best in Beetle's no-reference scenario (F1 = 0.651). However, it struggled with identifying contradictory answers in complex cases which may lead to arise of bias issue.

Overall, the reviewed studies demonstrate that LLMs, particularly GPT-4 achieve a moderate grading performance. However, a consistent limitation across all models was their susceptibility to bias due to imbalanced and underrepresented data. This issue was evident in models struggling with grading high-point or complex questions and maintaining consistency in evaluations. These findings emphasize the need for effective prompt engineering to guide LLMs in producing more accurate and reliable grading outcomes.

B. LLM-Based Automated Grading With Prompt Engineering

Prompt engineering has emerged as a technique for improving the performance of LLMs. By fine-tuning the prompts, LLM can be guided to produce more accurate and contextually appropriate evaluations of open-ended student responses. This section reviews studies that applied various prompt engineering strategies to improve grading performance across different datasets and educational contexts.

Xie et al. (2024) introduced an innovative approach to automated grading by leveraging OpenAI's GPT model with various prompting strategies including one-shot prompting, self-reflection prompting, and batch prompting on OS and Mohler datasets. Result suggests that batch prompting significantly improved grading performance, achieving a Mean Absolute Error (MAE) of 3.59, Root Mean Square Error (RMSE) of 4.64, and a Pearson Correlation of 0.79. The incorporation of a grouping and re-grouping strategy further enhanced the detection of grading anomalies. However, the study also identified limitations, such as domain specificity of datasets and high token costs.

Yoon (2023) developed an automated short answer grading (ASAG) system combining OpenAI GPT-3.5 for one-shot prompting with Sentence-BERT (SBERT) for text similarity scoring. This model was tested on the ASAP-SAS dataset. The integration of one-shot prompting with SBERT achieved an accuracy of 0.68 and a quadratic weighted kappa (QWK) of 0.71, surpassing the baseline SBERT model but still falling short of the fine-tuned BERT model (Accuracy: 0.77). Limitations noted in this study included the lack of justification key evaluation and occasional inconsistencies in scoring higher-grade responses.

Cohn et al. (2024) investigated the use of OpenAI's GPT-4 model with Chain-of-Thought (CoT) prompting and active learning for grading open-ended responses in middle school Earth Science. the best performance was achieved when CoT prompting was combined with active learning, reaching an F1 Score of 1.00 and a QWK of 0.90. However, the study acknowledged potential drawbacks, such as overfitting in simpler subtasks and ethical concerns, including bias and hallucinations in grading.

The study by Golchin et al. (2024) evaluated GPT-3.5 and GPT-4 for automated grading in MOOCs across three courses: Introductory Astronomy, Astrobiology, and History and Philosophy of Astronomy, using Zero-shot Chain-of-Thought (Zero-shot-CoT) prompting. GPT-4, combined with instructor-provided answers and rubrics, outperformed GPT-3.5 and peer grading, aligning closely with instructor grades (e.g., 8.65 in Introductory Astronomy vs. instructor 8.20, peer 7.55). However, it struggled with subjective content in the History and Philosophy course, highlighting the need for improvements in handling nuanced tasks.

Additionally, Henkel et al. (2024) assessed GPT-3.5 and GPT-4 in grading Science and History responses from 1,710 students (ages 5–16) on the Carousel learning platform. GPT-

4 with few-shot prompting achieved a Cohen's Kappa score of 0.70, near human agreement (0.75), but struggled with ambiguous responses. This study underscores GPT-4's potential for reducing grading workloads in low-stakes assessments while emphasizing the need for better handling of complex answers.

In short, these studies demonstrate that LLMs with prompt engineering consistently outperform models without it. This highlights the importance of applying this technique when using LLMs for specialized tasks. Despite these improvements, challenges remain. Models still face grading inconsistencies, particularly with subjective or higher-complexity responses and are prone to overfitting on simpler tasks. These findings emphasize that while prompt engineering significantly boosts LLM grading performance, further research is needed to enhance consistency and mitigate overfitting risks.

III. Problem Statement & Objective

As mentioned in most paper in literature review, manual grading of open-ended questions is a time-consuming and labor-intensive task, requiring significant effort from educators to assess responses accurately and fairly. This process becomes increasingly challenging with larger class sizes, leading to delays in providing feedback to students (Wiser et al., 2016).

Moreover, LLM-based grading systems have emerged as a solution but often struggle with biases due to imbalanced or unrepresentative datasets. These biases can result in unfair scoring and misrepresentation of a student's actual performance, raising concerns about the reliability of automated grading models (Gobrecht et al., 2024).

Additionally, maintaining grading consistency in LLM models remains a significant challenge although prompt engineering techniques is applied. The complexity of aligning automated grading with human-designed rubrics often leads to variability in scores, reducing trust in the model reliability (Yoon, 2023).

To address the issues mentioned above, there are three objectives is aimed to be achieve for this project.

1. To develop a grading model that evaluate open-ended questions based on human-curated rubrics using LLM.
2. To evaluate the performance of the LLM-based grading model across different prompting techniques.
3. To deploy a web application that can automate the grading of open-ended questions based on uploaded rubrics.

IV. Methodology

This project adopts the CRoss Industry Standard Process for Data Mining (CRISP-DM) methodology, a widely recognized and industry-accepted project lifecycle for structuring data science projects. This methodology consists of six key phases as illustrated in Figure 1. Each phase will be further elaborated in the following sections.

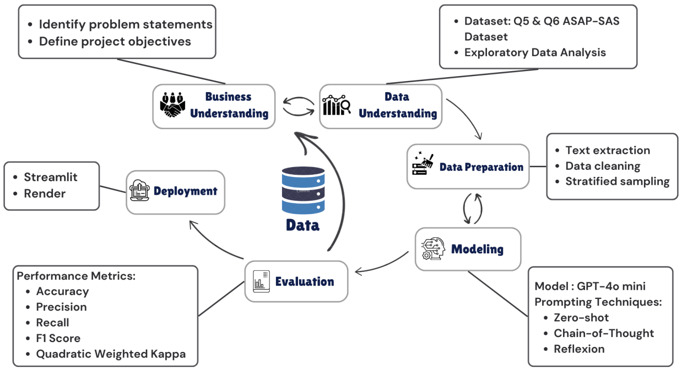


Figure 1: CRISP-DM Lifecycle

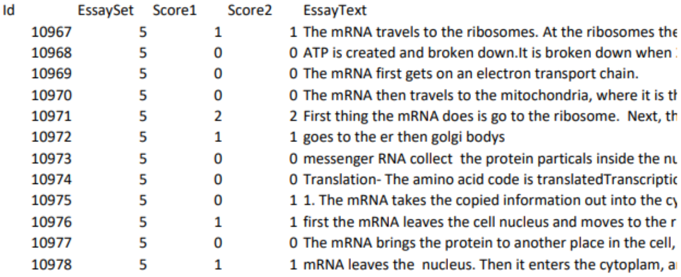
A. Business Understanding

The first step in this project was to clearly identify the problem statements and determine the objectives to ensure that the developed solution aligns with the project goals stated in Section III. Project scope is also defined to outline deliverables to ensure the project follows a structured approach on achieving the desired outcome.

B. Data Understanding

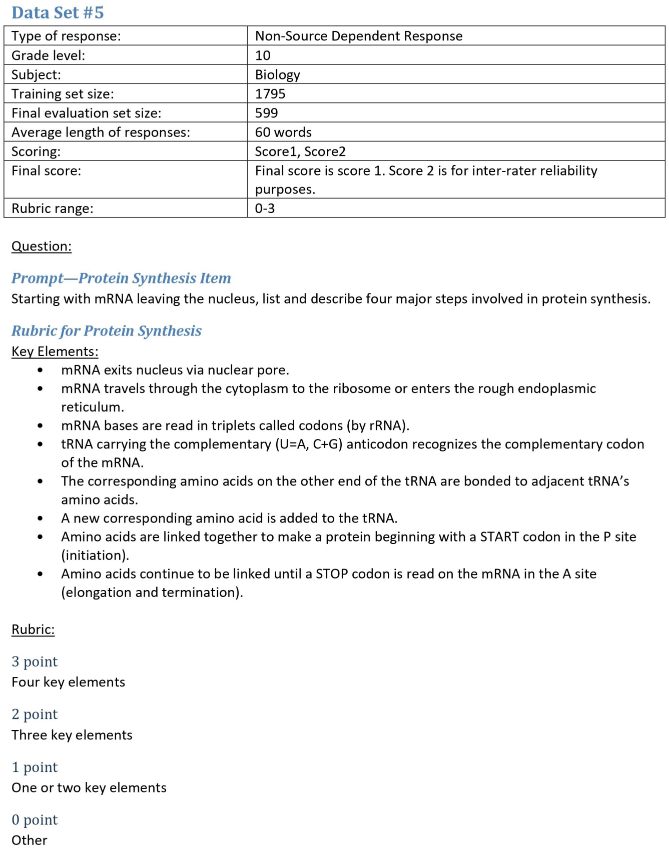
The data understanding phase focuses on data collection and Exploratory Data Analysis (EDA) to ensure the dataset's suitability for this project. The dataset used in this study is the ASAP-SAS (Automated Student Assessment Prize - Short Answer Scoring) dataset, sourced from a Kaggle competition held in 2013. It consists of ten open-ended questions from various domains, including Art, English, Science, and Biology. The dataset includes questions and grading rubrics in Word documents and CSV files containing student responses and corresponding human-graded scores.

For this study, Question 5 and Question 6 were specifically selected due to their well-defined grading rubrics with sample answers, which satisfy the project requirements. The dataset for Question 5 contains 1,795 rows of student responses, while Question 6 contains 1,797 rows. Both datasets consist of five columns: *Id*, *EssaySet*, *Score1*, *Score2*, and *EssayText*. Figures 2 and 3below provide a sample of the CSV file snippet and Word Document for Question 5.



Id	EssaySet	Score1	Score2	EssayText
10967	5	1	1	The mRNA travels to the ribosomes. At the ribosomes the
10968	5	0	0	ATP is created and broken down.It is broken down when
10969	5	0	0	The mRNA first gets on an electron transport chain.
10970	5	0	0	The mRNA then travels to the mitochondria, where it is th
10971	5	2	2	First thing the mRNA does is go to the ribosome. Next, th
10972	5	1	1	goes to the er then golgi bodys
10973	5	0	0	messenger RNA collect the protein particals inside the nu
10974	5	0	0	Translation- The amino acid code is translatedTranscriptio
10975	5	0	1	1. The mRNA takes the copied information out into the cy
10976	5	1	1	first the mRNA leaves the cell nucleus and moves to the r
10977	5	0	0	The mRNA brings the protein to another place in the cell,
10978	5	1	1	mRNA leaves the nucleus. Then it enters the cytoplam, a

Figure 2: CSV File Snippet for Question 5



Data Set #5

Type of response:	Non-Source Dependent Response
Grade level:	10
Subject:	Biology
Training set size:	1795
Final evaluation set size:	599
Average length of responses:	60 words
Scoring:	Score1, Score2
Final score:	Final score is score 1. Score 2 is for inter-rater reliability purposes.
Rubric range:	0-3

Question:

Prompt—Protein Synthesis Item

Starting with mRNA leaving the nucleus, list and describe four major steps involved in protein synthesis.

Rubric for Protein Synthesis

Key Elements:

- mRNA exits nucleus via nuclear pore.
- mRNA travels through the cytoplasm to the ribosome or enters the rough endoplasmic reticulum.
- mRNA bases are read in triplets called codons (by rRNA).
- tRNA carrying the complementary (U=A, C+G) anticodon recognizes the complementary codon of the mRNA.
- The corresponding amino acids on the other end of the tRNA are bonded to adjacent tRNA's amino acids.
- A new corresponding amino acid is added to the tRNA.
- Amino acids are linked together to make a protein beginning with a START codon in the P site (initiation).
- Amino acids continue to be linked until a STOP codon is read on the mRNA in the A site (elongation and termination).

Rubric:

3 point

Four key elements

2 point

Three key elements

1 point

One or two key elements

0 point

Other

Figure 3: Word Document for Question 5

Notably, the *Score1* and *Score2* columns represent grades assigned by two different human graders for the same student response, providing a basis for evaluating grading consistency.

Exploratory Data Analysis (EDA) was conducted to identify patterns and assess the quality of the data. The consistency between the two human-graded scores was analyzed to detect grading discrepancies. Based on the pie chart in Figure 4, it was observed that more than 96% of the responses had identical scores between the two graders for both questions. However, there is still a small portion of the responses had differing scores. This inconsistency highlights the inherent subjectivity and bias in human grading even when graded using the same grading rubrics.

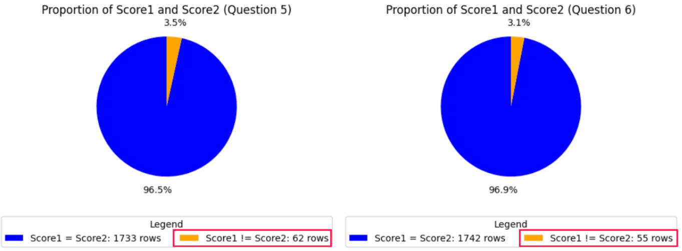


Figure 4: Pie Chart of Proportion Score1 and Score2

Additionally, the distribution of scores was analyzed to identify whether there is a class imbalance issue within the *Score* column. The bar chart in Figure 5 shows that for both questions, the distribution is heavily skewed towards Score 0, while higher scores are significantly underrepresented. This imbalance in the dataset must be addressed to mitigate potential biases during model training and evaluation.

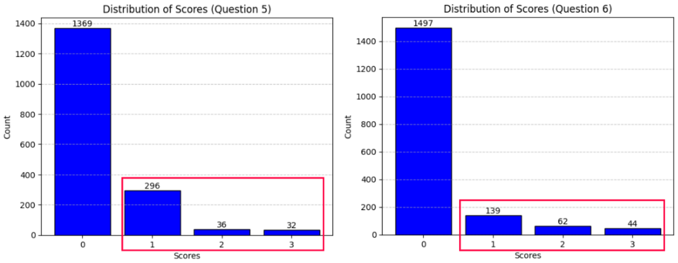


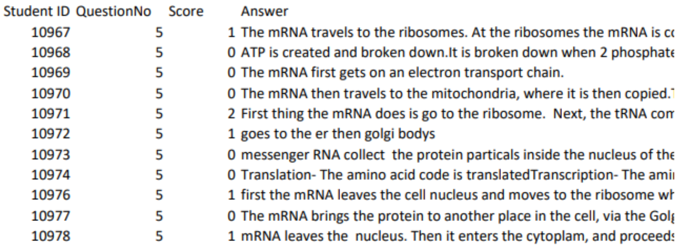
Figure 5: Bar Chart of Distribution of Scores

C. Data Preparation

Data preparation is the process of cleaning and transforming raw data for further modeling and analysis. The effectiveness of this phase plays a crucial role in determining the success of the modeling outcomes (Kerner et al., 2022). This phase involves three key processes which are data extraction, data cleaning, and data sampling.

The initial step focused on extracting relevant information which includes the questions, corresponding key answer elements and grading rubrics from the Word document using a keyword extraction method. The extracted details were then stored in a structured JSON format to facilitate further processing.

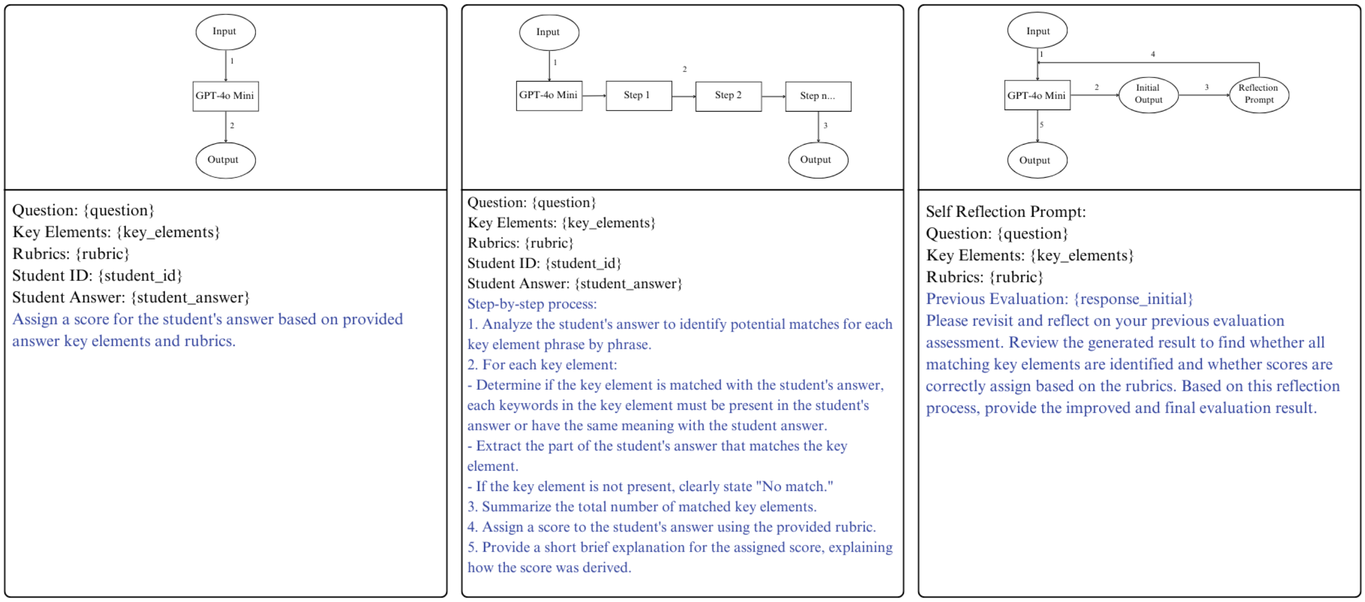
Data cleaning process started by filtering out responses with inconsistent grading scores identified earlier to maintain data integrity. After filtering, the *Score2* column was dropped, and the remaining columns were renamed to improve clarity: *Id* to *Student ID*, *EssaySet* to *QuestionNo*, *Score1* to *Score*, and *EssayText* to *Answer*. The cleaned CSV for Question 5 is shown as below in Figure 6.



Student ID	QuestionNo	Score	Answer
10967	5	1	The mRNA travels to the ribosomes. At the ribosomes the mRNA is co
10968	5	0	ATP is created and broken down.It is broken down when 2 phosphate
10969	5	0	The mRNA first gets on an electron transport chain.
10970	5	0	The mRNA then travels to the mitochondria, where it is then copied.T
10971	5	2	First thing the mRNA does is go to the ribosome. Next, the tRNA com
10972	5	1	goes to the er then golgi bodys
10973	5	0	messenger RNA collect the protein particals inside the nucleus of the
10974	5	0	Translation- The amino acid code is translatedTranscription- The ami
10976	5	1	first the mRNA leaves the cell nucleus and moves to the ribosome wh
10977	5	0	The mRNA brings the protein to another place in the cell, via the Gol
10978	5	1	mRNA leaves the nucleus. Then it enters the cytoplam, and proceeds

Figure 6: Cleaned CSV File Snippet for Question 5

To address the significant class imbalance identified during the EDA phase, stratified sampling was implemented. This technique ensured that each score category was equally represented in the sample (Bisht, 2024) for balanced model evaluation. In detail, 32 samples were randomly drawn from each score class for Question 5 and 44 samples for Question 6, resulting in a combined total of 304 responses. This balanced sampling strategy effectively mitigates bias and improving the robustness of the grading model.



Question: {question}
Key Elements: {key_elements}
Rubrics: {rubric}
Student ID: {student_id}
Student Answer: {student_answer}
Assign a score for the student's answer based on provided answer key elements and rubrics.

Question: {question}
Key Elements: {key_elements}
Rubrics: {rubric}
Student ID: {student_id}
Student Answer: {student_answer}
Step-by-step process:
1. Analyze the student's answer to identify potential matches for each key element phrase by phrase.
2. For each key element:
- Determine if the key element is matched with the student's answer, each keywords in the key element must be present in the student's answer or have the same meaning with the student answer.
- Extract the part of the student's answer that matches the key element.
- If the key element is not present, clearly state "No match."
3. Summarize the total number of matched key elements.
4. Assign a score to the student's answer using the provided rubric.
5. Provide a short brief explanation for the assigned score, explaining how the score was derived.

Self Reflection Prompt:
Question: {question}
Key Elements: {key_elements}
Rubrics: {rubric}
Previous Evaluation: {response_initial}
Please revisit and reflect on your previous evaluation assessment. Review the generated result to find whether all matching key elements are identified and whether scores are correctly assign based on the rubrics. Based on this reflection process, provide the improved and final evaluation result.

Figure 6.1: Zero-shot Prompt

Figure 6.2: Chain-of-Thought (CoT) Prompt

Figure 6.3: Reflexion Prompt

Figure 7: Prompting Engineering: Zero-shot, Chain-of-Thought (CoT) and Reflexion.

D. Modeling

The modeling phase of this research involved selecting a suitable Large Language Model (LLM) and applying prompt engineering techniques to optimize the model's performance in grading open-ended questions. The GPT-4o Mini model was chosen due to easily accessible API service and cost-effectiveness compared to other models in the GPT-4 series. Based on experiment carried out by CapeStart (2024), the statistical results also proved that GPT-4 series model outperforms other LLM models such as Gemini Pro and Llama 3.1 70b in various application such as MMLU (Massive Multitask Language Understanding) and MATH (Mathematical Reasoning Dataset).

To ensure consistency in the grading process and reduce randomness, the model's temperature parameter was set to zero in the API call so that LLM would generate responses strictly based on the provided input content and not rely on patterns from pre-trained corpora.

To optimize the model output, prompt engineering was employed to fine-tune the user prompts fed to the model. According to Sanh et al. (2022), prompt engineering involves designing and refining input text to guide generative artificial intelligence (generative AI) in producing desired outputs. Three prompting techniques were implemented in this project, the details of prompt can be viewed in Figure 8:

1. **Zero-shot:** Most basic technique where the model is provided with a task description or query without any prior examples, relying entirely on its understanding of the task to generate output. It also served as a baseline to compare with other advanced prompting techniques.
2. **Chain-of-Thought (CoT):** This technique guide the model to solve complex problems by breaking down tasks into intermediate reasoning steps to generate a more accurate response (Sanh et al., 2022).
3. **Reflexion:** This technique allows the model to self-reflect and iteratively re-evaluate its initial output, identify potential errors or areas for improvement and generate a refined output (Shinn et al., 2023).

While prompt engineering applied on user prompts, a standardized system prompt was designed to guide the model in evaluating student responses by matching them against predefined key elements and assigning scores based on the provided rubric. The output for all techniques was formatted in JSON to maintain uniformity in the results. To further refine the output, text parsing was performed to extract relevant information from the LLM-generated responses, converting unstructured text into a structured format.

E. Evaluation

The evaluation phase focused on assessing the performance of the Large Language Model (LLM) across different prompting techniques by comparing the model's predicted scores with actual human-graded scores as groundtruth. Two types of performance metrics were used to evaluate the result quantitatively: standard classification metrics and agreement-based metrics.

Standard classification metrics include Accuracy, Precision, Recall, and F1-Score. These metrics were used to measure the model's ability to predict the correct score labels for student responses.

1. **Accuracy:** Accuracy measures the proportion of student answers that the model correctly graded out of all pre-

dicted grades.

$$Accuracy = \frac{NumberofCorrectPredictions}{TotalNumberofPredictions} \quad (1)$$

2. **Precision:** Precision calculates the proportion of responses that were correctly identified as a specific score out of all responses the model predicted for that score.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (2)$$

3. **Recall:** Recall evaluates the model's ability to detect all relevant correct scores by evaluating how many actual positives were correctly identified.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (3)$$

4. **F1 Score:** The F1-Score is the harmonic mean of Precision and Recall, providing a balanced metric that considers both false positives and false negatives.

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

A higher value across these metrics indicates that the model is more effective at predicting scores that closely align with human-graded labels indicating higher consistency and reliability in automated grading.

Quadratic Weighted Kappa (QWK) was utilized to evaluate the agreement between the model's predicted scores and the human-assigned scores. QWK is particularly effective for evaluating ordinal data, as it accounts degree of agreement by awarding predictions that are closer to the actual label more than those that are farther off (Vanbelle et al., 2024). It can be calculated using the formula below.

$$\kappa = 1 - \frac{\sum_{i,j} w_{ij} O_{ij}}{\sum_{i,j} w_{ij} E_{ij}} \quad (5)$$

Where:

- O_{ij}: Observed agreement matrix.
- E_{ij}: Expected agreement matrix.
- w_{ij}: Weight matrix, representing the penalty for differences between the actual score i and predicted score j. Larger differences receive higher penalties and are calculated as:

$$w_{ij} = \frac{(i-j)^2}{(N-1)^2} \quad (6)$$

The QWK value ranges from -1 to 1 where 1 indicates perfect agreement and -1 indicates complete disagreement.

F. Deployment

To provide users with an accessible and interactive interface for automated grading, the model is deployed as a functional data product on a web application. Streamlit was selected as the web application framework due to its simplicity and efficiency in transforming Python scripts into interactive web applications with minimal coding overhead.

Through this web application, users can easily upload their questions, grading rubrics, and multiple answer files in the required format. The LLM will automate the grading process based on the provided rubric and display the results to the user in a structured format.

For cloud deployment, Render was chosen due to its ease of deployment, scalability, and seamless integration with GitHub. Its free-tier service which able to support Python applications also makes it a cost-effective hosting solution.

V. Results & Discussion

A. Model Evaluation

The performance of different prompting techniques— Zero-shot, Chain-of-Thought (CoT), and Reflexion was evaluated using five key metrics: Accuracy, F1 Score, Precision, Recall, and Quadratic Weighted Kappa (QWK). The results of the evaluation are presented in Table 1.

Table 1: Performance Comparison of Different Prompting Techniques

Prompting Technique	**Accuracy**	**F1 Score**	**Precision**	**Recall**	**QWK**
Zero-shot	0.625	0.627	0.635	0.625	0.752
Chain-of-Thought	0.763	0.764	0.766	0.763	0.871
Reflexion	0.641	0.646	0.656	0.641	0.782

Zero-shot prompting is used as the baseline to evaluate the other prompt engineering techniques. While it performed moderately across all evaluation metrics, it lacked the structured reasoning necessary to handle complex grading tasks effectively due to its simplicity. In contrast, the two advanced prompting techniques—Chain-of-Thought (CoT) and Reflexion demonstrated noticeable improvements across all performance metrics. This highlights the effectiveness of prompt engineering in enhancing the LLM's ability to perform the grading task more accurately and consistently.

Although Reflexion prompting allows the model to iteratively reflect and improve upon its initial responses generated using Zero-shot prompting, it only showed slight improvements. This limitation may be due to the model incorrectly adjusting correct predictions, potentially assigning wrong labels to initially correct answers. Reflexion's focus on self-correction without structured reasoning sometimes led to over-corrections or inconsistent grading outcomes.

Chain-of-Thought (CoT) prompting significantly outperformed both Zero-shot and Reflexion techniques across all evaluation metrics, including Accuracy (0.763), F1 Score (0.764), Precision (0.766), Recall (0.763), and Quadratic Weighted Kappa (QWK) (0.871). CoT's step-by-step reasoning allowed the model to break down the grading task into intermediate logical steps, enabling a more thorough assessment

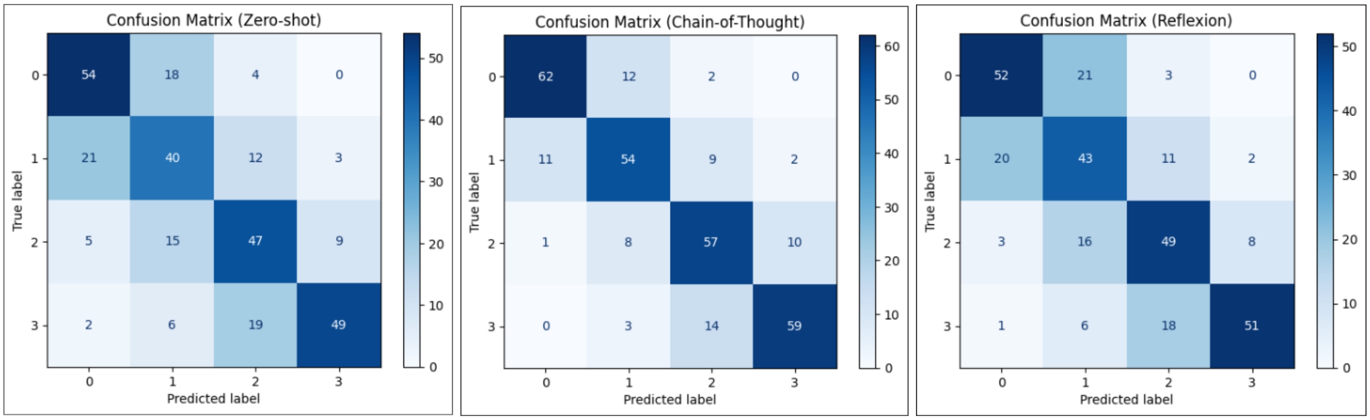


Figure 6.1: Confusion Matrix (Zero-shot)

Figure 6.2: Confusion Matrix (Chain-of-Thoughts)

Figure 6.3: Confusion Matrix (Reflexion)

Figure 8: Prompting Engineering: Zero-shot, Chain-of-Thought (CoT) and Reflexion.

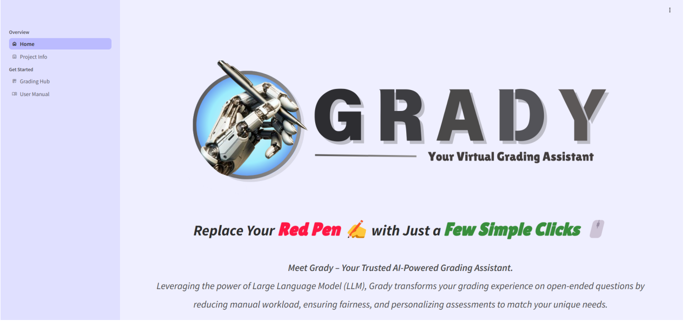
of student answers against the key elements comparison. High score in standard classification metrics highlights the model's ability effectively distinguishes correct from incorrect answers and consistently assigns the correct grades. On the other hand, a high QWK score indicates a strong agreement between the model's predicted scores and human-graded scores.

Figures above shows the confusion matrix generated based on the predicted and actual labels for all prompting techniques. While it is already evident that the Chain-of-Thought (CoT) prompting will shows the best performance across all metrics, it is important to note that all prompting techniques demonstrate consistent grading across each class label. This consistency suggests that the grading process is unbiased, with no particular score class being disproportionately favored or overlooked.

B. Data Product

The deployed data product, named Grady, is a web-based application developed using the Streamlit framework. The application is designed to provide users with a seamless and interactive interface for automated grading of open-ended questions. It consists of four main page:

1. **Home Page:** This is the landing page that offers an overview of the application and captures users' attention. It provides a brief introduction to Grady and highlights the key features of the website.

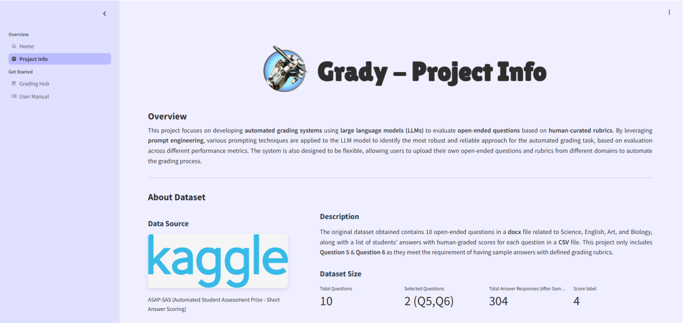

Overview
Home
Project Info
Get Started
Grading Hub
User Manual
GRADY
Your Virtual Grading Assistant
Replace Your Red Pen ✍ with Just a Few Simple Clicks

Meet Grady – Your Trusted AI-Powered Grading Assistant.

Leveraging the power of Large Language Model (LLM), Grady transforms your grading experience on open-ended questions by reducing manual workload, ensuring fairness, and personalizing assessments to match your unique needs.

Figure 9: Streamlit Home Page

2. **Project Info Page:** This page presents detailed information about the project, including the dataset used, the selection of the LLM model, insights into prompt engineering, the prompting techniques applied in this project and the evaluation results. It allows users to gain an in-depth understanding of the entire project.



Grady – Project Info

Overview

This project focuses on developing **automated grading systems** using **large language models (LLMs)** to evaluate **open-ended questions** based on **human-curated rubrics**. By leveraging **prompt engineering**, various prompting techniques are applied to the LLM model to identify the most robust and reliable approach for the automated grading task, based on evaluation across different performance metrics. The system is also designed to be flexible, allowing users to upload their own open-ended questions and rubrics from different domains to automate the grading process.

About Dataset

Data Source

ASAP-SAS (Automated Student Assessment Prize - Short Answer Scoring)

Description

The original dataset obtained contains 10 open-ended questions in a **docx** file related to Science, English, Art, and Biology, along with a list of students' answers with human-graded scores for each question in a **CSV** file. This project only includes **Question 5** & **Question 6** as they meet the requirement of having sample answers with defined grading rubrics.

Dataset Size

Total Questions	Selected Questions	Total Answer Responses (After Sam...	Score label
10	2 (Q5,Q6)	304	4

Figure 10: Streamlit Project Info Page

3. **User Manual Page:** This page serves as a guide for users that offers step-by-step instructions including file formatting, uploading and interpreting grading results to show the user how to use the application effectively.

Grady – Project Info

Overview

This project focuses on developing **automated grading systems** using **large language models (LLMs)** to evaluate **open-ended questions** based on **human-curated rubrics**. By leveraging **prompt engineering**, various prompting techniques are applied to the LLM model to identify the most robust and reliable approach for the automated grading task, based on evaluation across different performance metrics. The system is also designed to be flexible, allowing users to upload their own open-ended questions and rubrics from different domains to automate the grading process.

About Dataset

Data Source

ASAP-SAS (Automated Student Assessment Prize - Short Answer Scoring)

Description

The original dataset obtained contains 10 open-ended questions in a **docx** file related to Science, English, Art, and Biology, along with a list of students' answers with human-graded scores for each question in a **CSV** file. This project only includes **Question 5** & **Question 6** as they meet the requirement of having sample answers with defined grading rubrics.

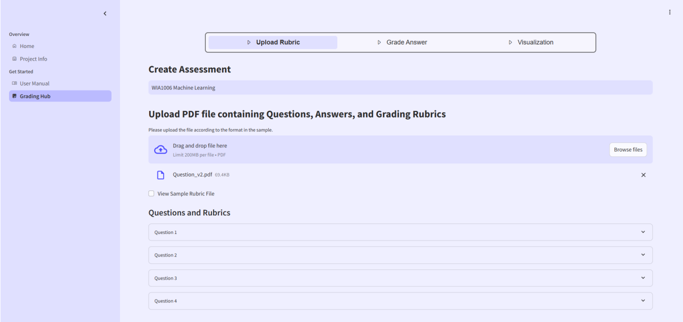
Dataset Size

Total Questions	Selected Questions	Total Answer Responses (After Sam...	Score label
10	2 (Q5,Q6)	304	4

Figure 11: Streamlit User Manual Page

4. **Grading Hub Page:** This is the core component of the application where the grading model is integrated which can be sub-divided into three more pages:

4.1. **Upload Rubric Page:** This page allows users to create an assessment and upload a question rubric in PDF format based on the required template.



Create Assessment

WIA1006 Machine Learning

Upload PDF file containing Questions, Answers, and Grading Rubrics

Please upload the file according to the format in the sample.

Drag and drop file here
Limit 200MB per file • PDF

Question_v2.pdf 69.4KB

□ View Sample Rubric File

Questions and Rubrics

Question 1

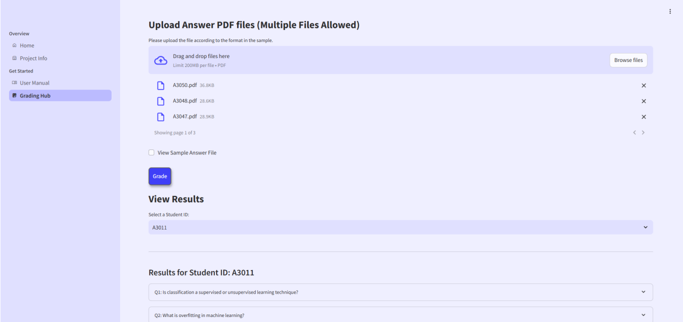
Question 2

Question 3

Question 4

Figure 12: Streamlit Upload Rubric Page

4.2. **Grade Answer Page:** This page allows users to upload multiple PDF answer files for grading. The system employs parallel processing which enables simultaneous grading to improve efficiency.



Upload Answer PDF files (Multiple Files Allowed)

Please upload the file according to the format in the sample.

Drag and drop files here
Limit 200MB per file • PDF

A3050.pdf 36.8KB

A3048.pdf 28.6KB

A3047.pdf 28.9KB

Showing page 1 of 3

□ View Sample Answer File

View Results

Select a Student ID:

A3011

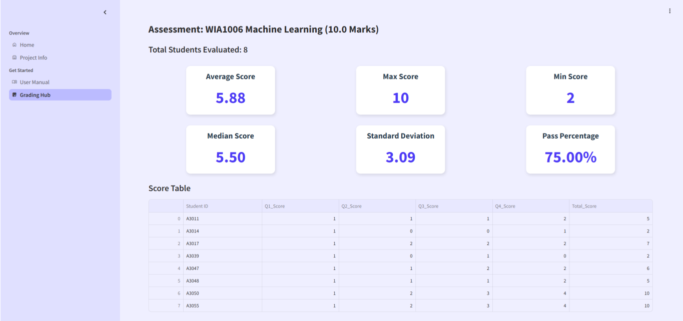
Results for Student ID: A3011

Q1: Is classification a supervised or unsupervised learning technique?

Q2: What is overfitting in machine learning?

Figure 13: Streamlit Grade Answer Page

4.3. **Visualization Page:** This page provides users with visualizations of the overall performance of the grading results.



Assessment: WIA1006 Machine Learning (10.0 Marks)

Total Students Evaluated: 8

Average Score	Max Score	Min Score
5.88	10	2

Median Score	Standard Deviation	Pass Percentage
5.50	3.09	75.00%

Score Table

	Student ID	Q1_Score	Q2_Score	Q3_Score	Q4_Score	Total_Score
0	A3011	1	1	1	2	5
1	A3014	1	0	0	1	2
2	A3017	1	2	2	2	7
3	A3039	1	0	1	0	2
4	A3047	1	1	2	2	6
5	A3048	1	1	1	2	5
6	A3050	1	2	3	4	10
7	A3055	1	2	3	4	10

Figure 14: Streamlit Visualization Page

VI. Conclusion

In conclusion, the objectives of this project is successfully achieved by developing an automated grading model for open-ended questions using the GPT-4o Mini Large Language Model (LLM). The model was evaluated using key performance metricsincluding Accuracy, Precision, Recall, F1 Score, and Quadratic Weighted Kappa (QWK) across different prompting techniques: Zero-shot, Chain-of-Thought, and Reflexion prompting, with Chain-of-Thought (CoT) outperforming the others based on these metrics. Additionally, the model was effectively deployed as a user-friendly web application named Grady with Streamlit and hosted on Render. This application enables users to upload grading rubrics and multiple answer files to automate grading tasks. This project is expected to contribute significantly to the educational sector by providing a reliable solution that reduces educators' manual grading workload and promotes a fair, unbiased and consistent grading system.

Despite the successes of the implemented solution, it is important to acknowledge several limitations that need to be addressed for future improvements. Firstly, the model was evaluated using a limited dataset, which restricts the diversity and complexity of student responses. This limitation may hinder the model's ability to generalize effectively across various domains, grade levels, and question types. Secondly, the evaluation primarily focused on quantitative metrics, without integrating qualitative analysis or human feedback. This may result in overlooking subtle grading nuances and the contextual understanding required for accurate assessment. Moreover, the current system lacks essential functionalities, such as a database to store previous assessments and the capability to handle multiple file formats for uploads as the project primarily focused on the data science and modeling aspects rather than system functionality.

To address these limitations, future work should focus on expanding the dataset by collecting more real-world student responses across diverse academic domains, grade level and question formats. This will improve the model's generalizability and robustness in handling a wider variety of open-ended responses. Incorporating Human-in-the-Loop (HITL) validation is another essential step to allow educators to provide feedback and corrections to the system's grading to enhance model reliability. Furthermore, integrating adaptive rubrics that evolve with changing assessment standards and applying explainable AI (XAI) techniques could provide more transparent and interpretable grading justifications. Lastly, system enhancements may include implementing a database for storing past assessments, supporting multiple file formats for uploads, and offering detailed insights into the answer comparison process to improve user experience and system functionality.

Acknowledgement

I would like to sincerely express my deepest appreciation to my supervisor, Prof. Dr. Nor Liyana for her continuous guidance and support throughout this project. Her insightful feedback and dedicated mentorship have been instrumental in shaping the success of this research.

References

Bisht, R. (2024). *What is stratified sampling? definition, types, and examples* [Researcher.Life]. https://researcher.life/blog/article/what-is-stratified-sampling-definition-types-examples/

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language models are few-shot learners [arXiv preprint]. https://doi.org/10.48550/arXiv.2005.14165

CapeStart. (2024). *The battle of the llms: Llama 3 vs. gpt-4 vs. gemini* [Accessed: 2024-01-13]. https://www.capestart.com/resources/blog/the-battle-of-the-llms-llama-3-vs-gpt-4-vs-gemini/?utm_source=chatgpt.com

Chu, Y., Li, H., Yang, K., Shomer, H., Liu, H., Copur-Gencturk, Y., & Tang, J. (2024). A llm-powered automatic grading framework with human-level guidelines optimization [arXiv preprint]. https://doi.org/10.48550/arXiv.2410.02165

Cohn, C., Hutchins, N., Le, T., & Biswas, G. (2024). A chain-of-thought prompting approach with llms for evaluating students' formative assessment responses in science [In press at EAAI-24: The 14th Symposium on Educational Advances in Artificial Intelligence, arXiv preprint]. https://doi.org/10.48550/arXiv.2403.14565

Gnanaprakasam, J., & Lourdusamy, R. (2024). The role of ai in automating grading: Enhancing feedback and efficiency. In S. Kadry (Ed.), *Artificial intelligence and education – shaping the future of learning*. IntechOpen. https://doi.org/10.5772/intechopen.1005025

Gobrecht, A., Tuma, F., Möller, M., Zöller, T., Zakhvatkin, M., Wuttig, A., Sommerfeldt, H., & Schütt, S. (2024). Beyond human subjectivity and error: A novel ai grading system [arXiv preprint]. https://doi.org/10.48550/arXiv.2405.04323

Golchin, S., Garuda, N., Impey, C., & Wenger, M. (2024). Large language models as moocs graders [arXiv preprint]. https://doi.org/10.48550/arXiv.2402.03776

Henkel, O., Boxer, A., Hills, L., & Roberts, B. (2024). Can large language models make the grade? an empirical study evaluating llms ability to mark short answer questions in k-12 education [arXiv preprint]. https://doi.org/10.48550/arXiv.2405.02985

Kerner, H., Campbell, J., & Strickland, M. (2022). Chapter 1 - introduction to machine learning. In J. Helbert, M. D'Amore, M. Aye, & H. Kerner (Eds.), *Machine learning for planetary science* (pp. 1–24). Elsevier. https://doi.org/10.1016/B978-0-12-818721-0.00007-0

Kortemeyer, G. (2023). Performance of the pre-trained large language model gpt-4 on automated short answer grading [arXiv preprint]. https://doi.org/10.48550/arXiv.2309.09338

Meyer, G., Breuer, P., & Fürst, J. (2024). Asag2024: A combined benchmark for short answer grading. *Proceedings of the 2024 ACM Virtual Global Computing Education Conference V. 2 (SIGCSE Virtual 2024)*, December 5–8. https://doi.org/10.1145/3649409.3691083

Pinto, G., Cardoso-Pereira, I., Ribeiro, D. M., Lucena, D., de Souza, A., & Gama, K. (2023). Large language models for education: Grading open-ended questions using chatgpt [arXiv preprint]. https://doi.org/10.48550/arXiv.2307.16696

Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Scao, T. L., Raja, A., Dey, M., Bari, M. S., Xu, C., Thakker, U., Sharma, S. S., Szczechla, É., Kim, T., Chhablani, G., Nayak, N., & Rush, A. M. (2022). Multitask prompted training enables zero-shot task generalization [arXiv preprint]. https://arxiv.org/abs/2110.08207

Schneider, J., Schenk, B., & Niklaus, C. (2024). Towards llm-based autograding for short textual answers [arXiv preprint]. *Proceedings of the 16th International Conference on Computer Supported Education (CSEDU 2024)*. https://doi.org/10.48550/arXiv.2309.11508

Shinn, N., Cassano, F., Berman, E., Gopinath, A., Narasimhan, K., & Yao, S. (2023). Reflexion: Language agents with verbal reinforcement learning [arXiv preprint]. https://doi.org/10.48550/arXiv.2303.11366

Stevens, D. D., & Levi, A. J. (2012). *Introduction to rubrics: An assessment tool to save grading time, convey effective feedback, and promote student learning* (2nd). Routledge. https://doi.org/10.4324/9781003445432

Vanbelle, S., Engelhart, C. H., & Blix, E. (2024). A comprehensive guide to study the agreement and reliability of multi-observer ordinal data. *BMC Medical Research Methodology*, *24*(310). https://doi.org/10.1186/s12874-024-02431-y

Wiser, M. J., Mead, L. S., Smith, J. J., & Pennock, R. T. (2016). Comparing human and automated evaluation of open-ended student responses to questions of evolution. *Proceedings of the Fifteenth International Conference on Artificial Life*, 116–122. https://doi.org/10.7551/978-0-262-33936-0-ch025

Xie, W., Niu, J., Xue, C. J., & Guan, N. (2024). Grade like a human: Rethinking automated assessment with large language models [arXiv preprint]. https://doi.org/10.48550/arXiv.2405.19694

Yamamoto, M., Umemura, N., & Kawano, H. (2017). Automated essay scoring system based on rubric. In *Applied computing & information technology* (pp. 177–190). Springer.

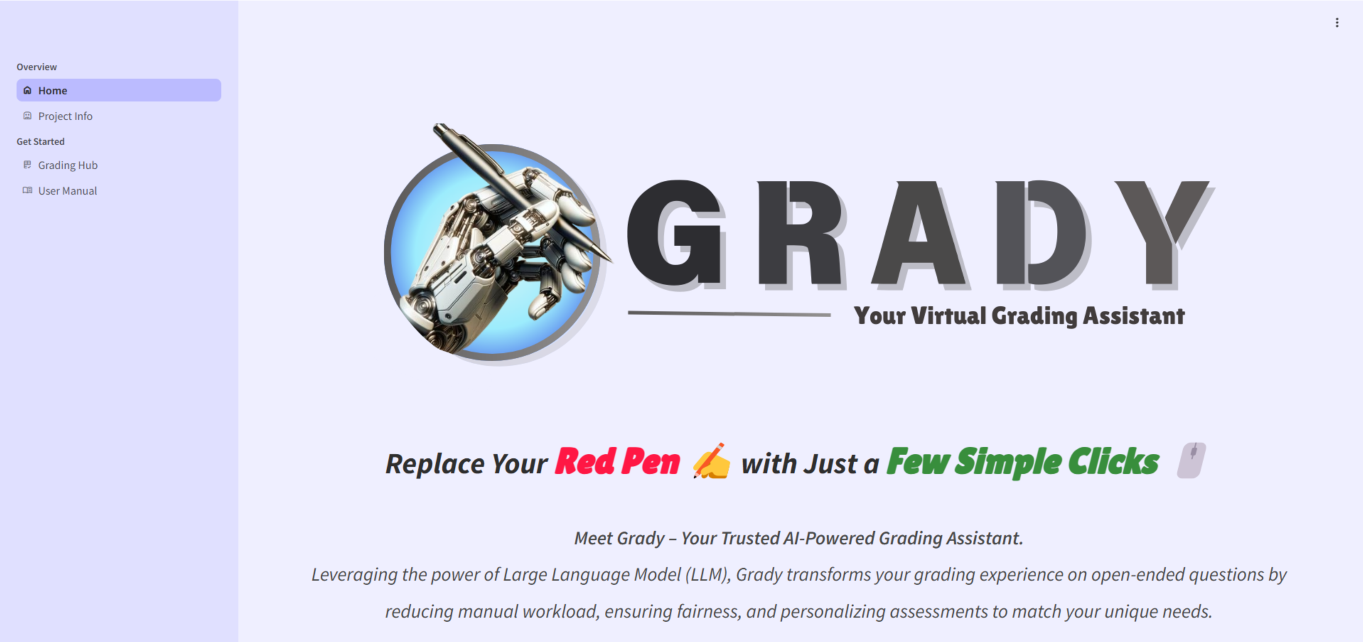
Yoon, S.-Y. (2023). Short answer grading using one-shot prompting and text similarity scoring model [arXiv preprint]. https://doi.org/10.48550/arXiv.2305.18638

Appendix - User Manual

This user manual provides step-by-step guidance on navigating the website. Users can access the website via this link: https://dsp-llm-open-ended-question-grader-based.onrender.com.

Home Page:

This is the landing page once the users had entered the website. It offers an overview of the application and captures users' attention. It provides a brief introduction to Grady and highlights the key features of the website.



Replace Your **Red Pen** ✍ with Just a **Few Simple Clicks**

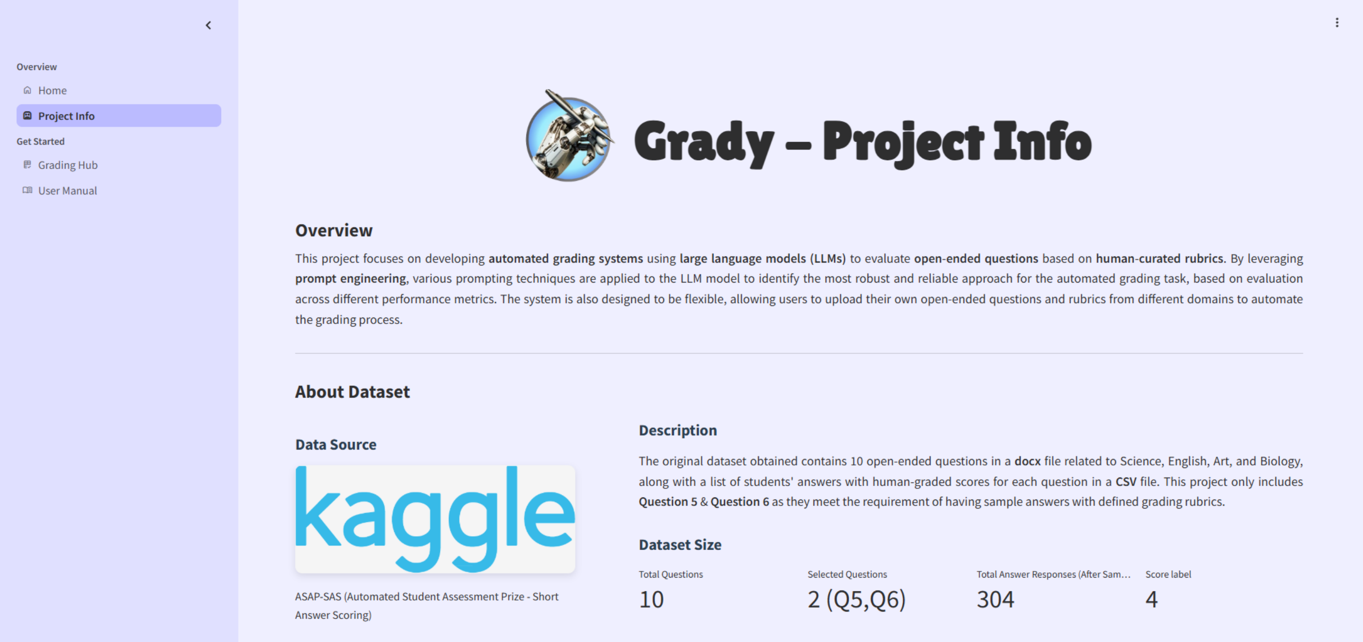
Meet Grady – Your Trusted AI-Powered Grading Assistant.

Leveraging the power of Large Language Model (LLM), Grady transforms your grading experience on open-ended questions by reducing manual workload, ensuring fairness, and personalizing assessments to match your unique needs.

Figure A1: Home Page

Project Info Page:

Users can navigate to this page to get detailed information about the project, including the dataset used, the selection of the LLM model, insights into prompt engineering, the prompting techniques applied in this project, and the evaluation results. It allows users to gain an in-depth understanding of the entire project.



Grady – Project Info

Overview

This project focuses on developing **automated grading systems** using **large language models (LLMs)** to evaluate **open-ended questions** based on **human-curated rubrics**. By leveraging **prompt engineering**, various prompting techniques are applied to the LLM model to identify the most robust and reliable approach for the automated grading task, based on evaluation across different performance metrics. The system is also designed to be flexible, allowing users to upload their own open-ended questions and rubrics from different domains to automate the grading process.

About Dataset

Data Source

ASAP-SAS (Automated Student Assessment Prize - Short Answer Scoring)

Description

The original dataset obtained contains 10 open-ended questions in a **docx** file related to Science, English, Art, and Biology, along with a list of students' answers with human-graded scores for each question in a **CSV** file. This project only includes **Question 5** & **Question 6** as they meet the requirement of having sample answers with defined grading rubrics.

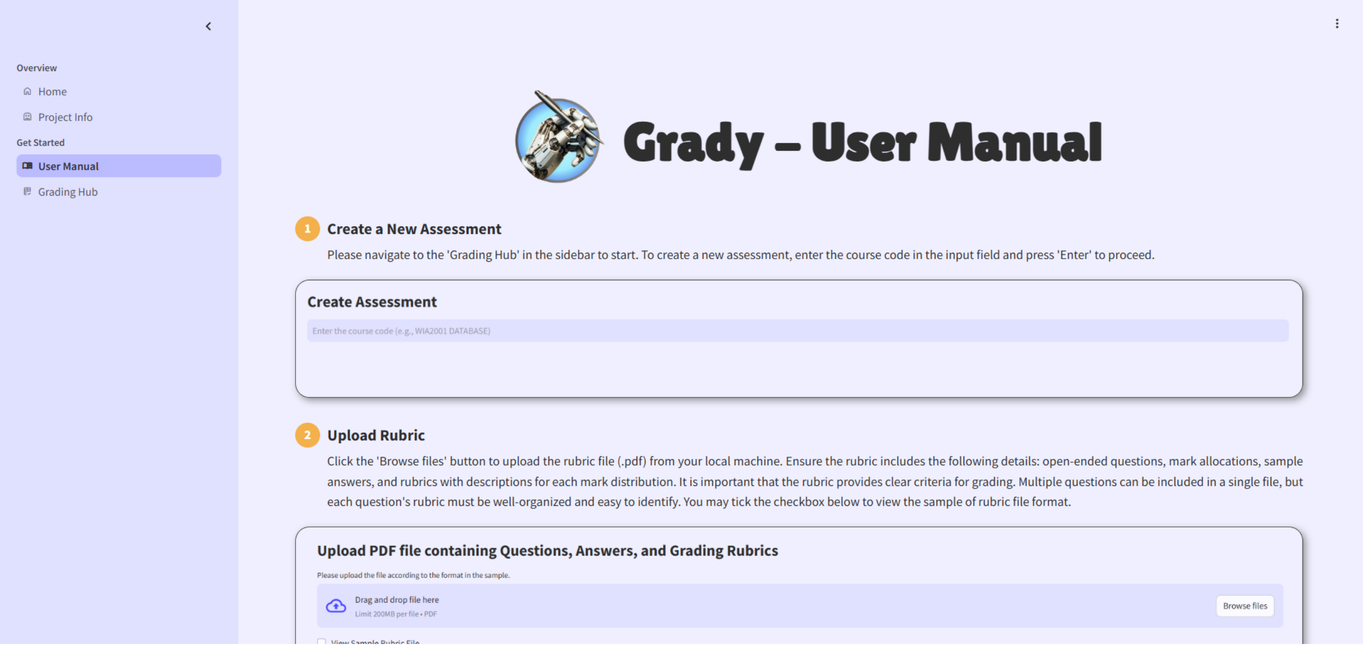
Dataset Size

Total Questions	Selected Questions	Total Answer Responses (After Sam...	Score label
10	2 (Q5,Q6)	304	4

Figure A2: Project Info Page

User Manual Page:

Users can refer to the user manual for step-by-step instructions on file formatting, uploading, and interpreting grading results to effectively use the application.



Grady – User Manual

1 Create a New Assessment

Please navigate to the 'Grading Hub' in the sidebar to start. To create a new assessment, enter the course code in the input field and press 'Enter' to proceed.

Create Assessment

Enter the course code (e.g., WIA2001 DATABASE)

2 Upload Rubric

Click the 'Browse files' button to upload the rubric file (.pdf) from your local machine. Ensure the rubric includes the following details: open-ended questions, mark allocations, sample answers, and rubrics with descriptions for each mark distribution. It is important that the rubric provides clear criteria for grading. Multiple questions can be included in a single file, but each question's rubric must be well-organized and easy to identify. You may tick the checkbox below to view the sample of rubric file format.

Upload PDF file containing Questions, Answers, and Grading Rubrics

Please upload the file according to the format in the sample.

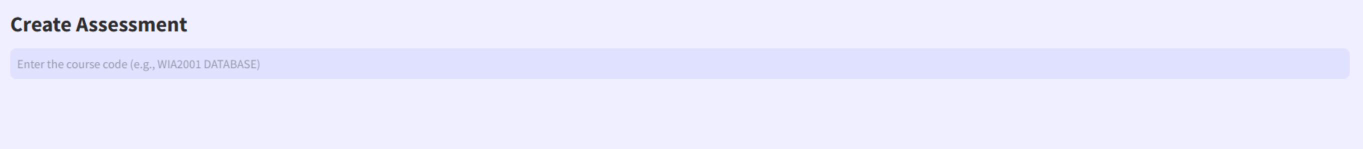
Drag and drop file here

Limit 200MB per file • PDF

Figure A3: User Manual Page

Grading Hub Page:

Users need to navigate to this page to start the grading assessment. To create a new assessment, enter the course code in the input field and press ’Enter’ to proceed.

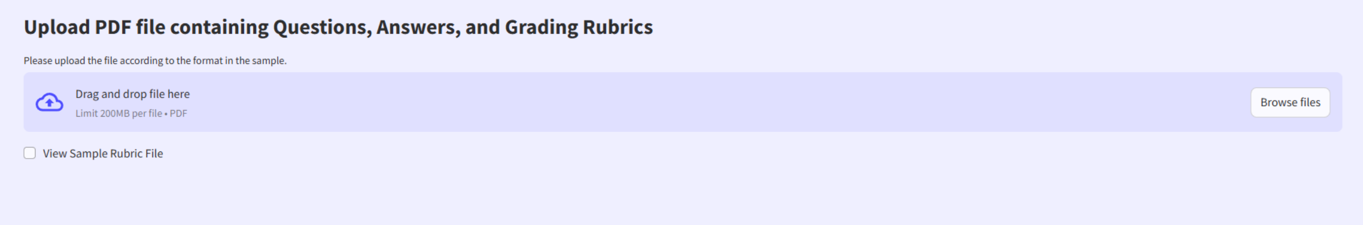


Create Assessment

Enter the course code (e.g., WIA2001 DATABASE)

Figure A4: Create New Assessment

Click the ’Browse files’ button to upload the rubric file (.pdf) from your local machine. Ensure the rubric includes the following details: open-ended questions, mark allocations, sample answers, and rubrics with descriptions for each mark distribution. Multiple questions can be included in a single file, but each question’s rubric must be well-organized and easy to identify. You may tick the checkbox below to view the sample of rubric file format.



Upload PDF file containing Questions, Answers, and Grading Rubrics

Please upload the file according to the format in the sample.

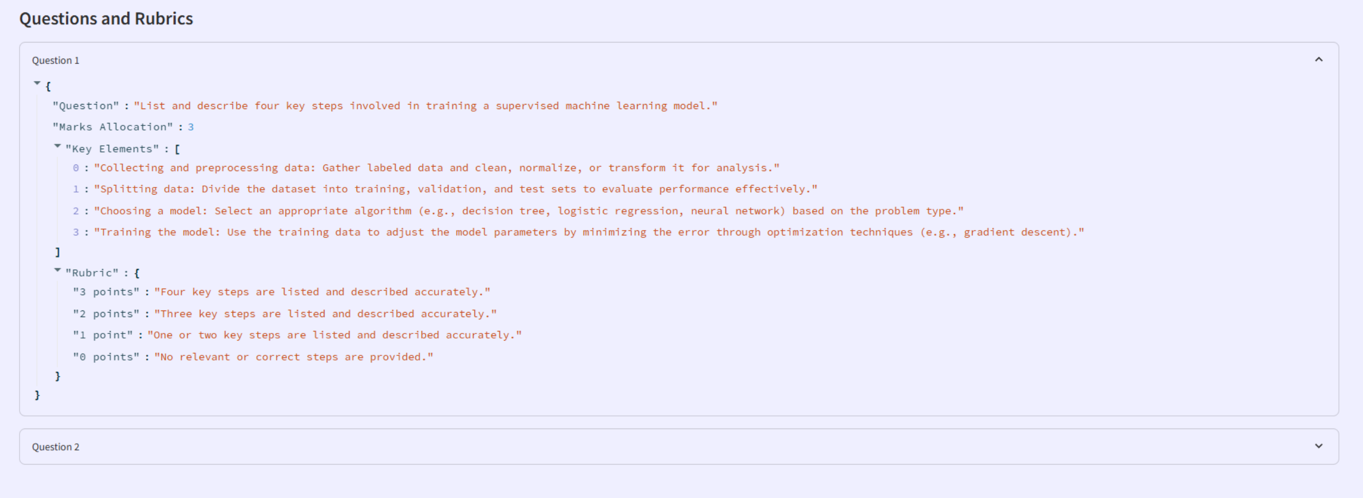
Drag and drop file here

Limit 200MB per file • PDF

□ View Sample Rubric File

Figure A5: Upload Rubric

After the system finishes loading the question, it will display the details of each question along with their respective rubrics. Click on the ”Question” tab to view the details and ensure the system has extracted the correct information.



Questions and Rubrics

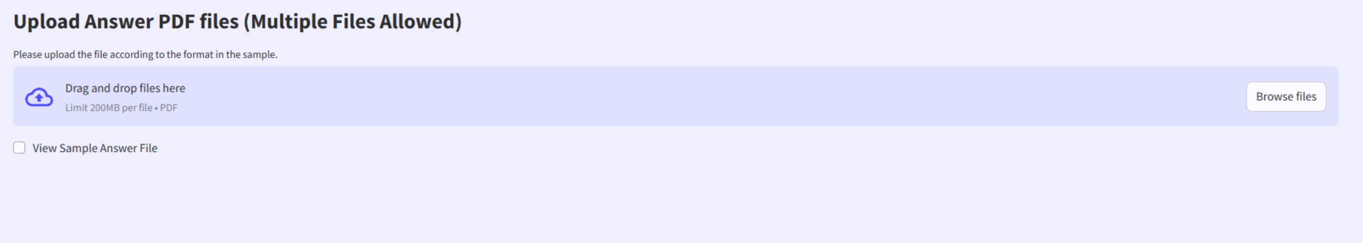
Question 1

```
{
  "Question" : "List and describe four key steps involved in training a supervised machine learning model."
  "Marks Allocation" : 3
  "Key Elements" : [
    0 : "Collecting and preprocessing data: Gather labeled data and clean, normalize, or transform it for analysis."
    1 : "Splitting data: Divide the dataset into training, validation, and test sets to evaluate performance effectively."
    2 : "Choosing a model: Select an appropriate algorithm (e.g., decision tree, logistic regression, neural network) based on the problem type."
    3 : "Training the model: Use the training data to adjust the model parameters by minimizing the error through optimization techniques (e.g., gradient descent)."
  ]
  "Rubric" : {
    "3 points" : "Four key steps are listed and described accurately."
    "2 points" : "Three key steps are listed and described accurately."
    "1 point" : "One or two key steps are listed and described accurately."
    "0 points" : "No relevant or correct steps are provided."
  }
}
```

Question 2

Figure A6: View Rubric

Next, click the "Grade Answer" tab in the navigation bar to proceed with uploading the answers. Click the "Browse files" button to upload the answer files (.pdf) from your local machine. Multiple answer files are supported as the system allow parallel processing for the grading task. The file name will be treated as a Student ID.



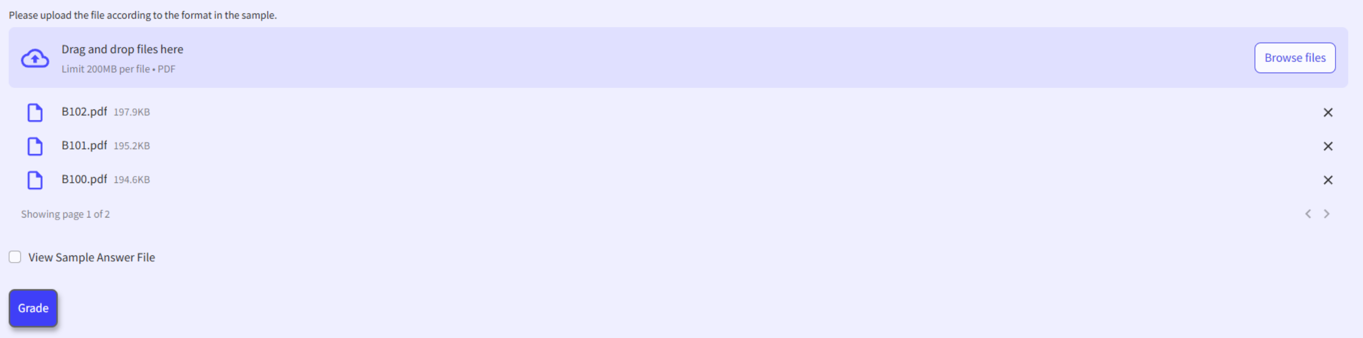
Upload Answer PDF files (Multiple Files Allowed)

Please upload the file according to the format in the sample.

□ View Sample Answer File

Figure A7: Upload Answer

Once the files are uploaded, click the "Grade" button to initiate the grading process. Please note that the grading process may take around one minute to complete, depending on the size of the uploaded files.



Please upload the file according to the format in the sample.

B102.pdf 197.9KB

B101.pdf 195.2KB

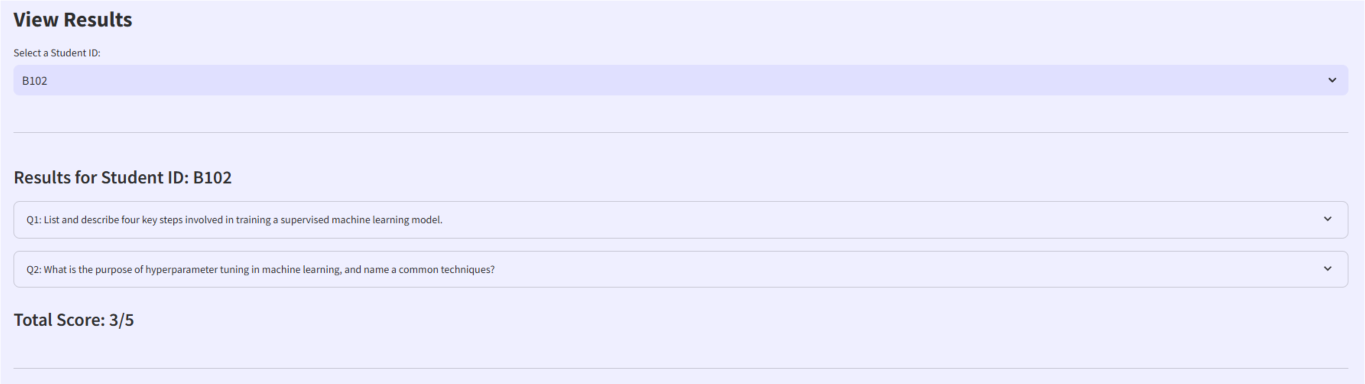
B100.pdf 194.6KB

Showing page 1 of 2

□ View Sample Answer File

Figure A8: Initialize Grading

Once the grading process is completed, the results will be displayed in a list format for each question that allow users to evaluate and review the grading outcomes. Users can utilize the drop-down menu to select the specific answer they wish to view. The results include the question, marks allocation, answer matching, graded score and an explanation for the assigned score.



View Results

Select a Student ID: B102

Results for Student ID: B102

Q1: List and describe four key steps involved in training a supervised machine learning model.

Q2: What is the purpose of hyperparameter tuning in machine learning, and name a common techniques?

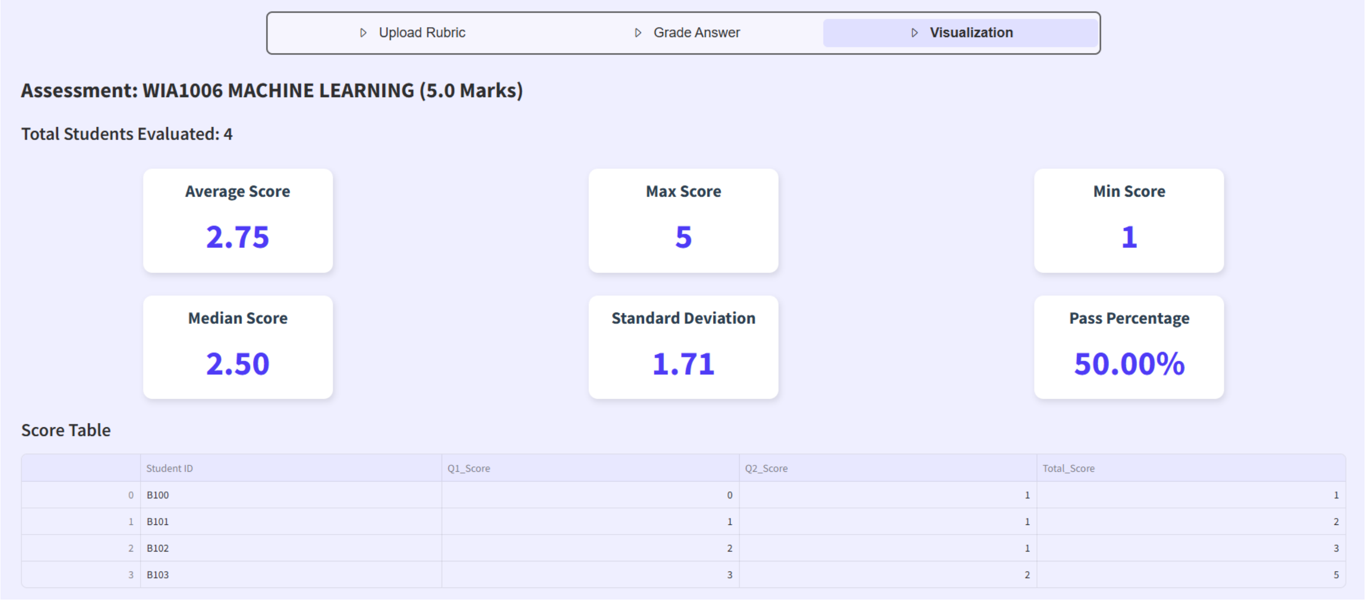
Total Score: 3/5

Figure A9: View Grading Result

After reviewing the grading results, users can download the results in CSV format for their records. Click the ”Download” button to save the results, which include all the evaluated details for each uploaded answer.

Figure A10: Download Result

Users can navigate to the ’Visualization’ tab to view the graphs and charts generated based on the overall total score results for the uploaded answers. These visualizations provide valuable insights, allowing users to analyze overall performance trends, identify patterns and evaluate the grading outcomes effectively.



Assessment: WIA1006 MACHINE LEARNING (5.0 Marks)

Total Students Evaluated: 4

Average Score: 2.75

Max Score: 5

Min Score: 1

Median Score: 2.50

Standard Deviation: 1.71

Pass Percentage: 50.00%

Score Table

	Student ID	Q1_Score	Q2_Score	Total_Score
0	B100	0	1	1
1	B101	1	1	2
2	B102	2	1	3
3	B103	3	2	5

Figure A11: Visualize Result