IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 5, May 2025

DOI: 10.17148/IARJSET.2025.125200

Student Performance Evaluation Based on Machine Learning

Bhavyashree H D[1], Prajwal C [2], Soujanya S [3], Chandan Kumar C [4], Akhila D J [5]

Assistant Professor, Department of ISE, MITM, Mysore, VTU Belagavi, India[1]

UG Students, Department of ISE, MITM, Mysore, VTU Belagavi, India[2-5]

Abstract: This project is a Flask-based web application developed to automate the evaluation of student performance across various technical domains. It allows users to upload text or image-based content (presumably representing course submissions or project work), which is then analyzed using a backend machine learning model implemented in Python (model.py). The application provides a user-friendly interface for input through HTML templates and produces evaluative results in real time. Supporting features include integration of static media resources and structured course categorization (e.g., AI, Python, IoT). The goal is to streamline the assessment process by leveraging intelligent algorithms for quick and accurate evaluations.

Keywords: Student Evaluation Automated Grading

I. INTRODUCTION

The rapid growth of digital learning environments has increased the demand for automated tools that can assist educators in assessing student work efficiently and accurately. Traditional evaluation methods often require substantial manual effort and are subject to human bias or inconsistency. To address these challenges, this project introduces a Flask-based student evaluation system that leverages Natural Language Processing (NLP) and Optical Character Recognition (OCR) to automate the assessment of student submissions.

The system allows students to upload answers in text or image format, processes these inputs using the Tesseract OCR engine and spaCy NLP library, and computes a similarity score against reference answers. By integrating intelligent evaluation algorithms into a user-friendly web interface, the application aims to enhance the scalability and objectivity of the grading process across various subjects and formats.

II. LITERATURE SURVEY

With the increasing integration of technology into education, several studies and systems have explored the automation of student evaluation. One prominent technique is Optical Character Recognition (OCR), with tools like Tesseract OCR widely adopted for converting handwritten or printed content into machine-readable text. Smith (2007) introduced Tesseract as an open-source OCR engine capable of high accuracy in text recognition, which has since been applied in various educational tools for processing scanned assignments and answer sheets. The studies are listed as below:

Suman Chowdhury. [1] proposed system involves data preprocessing, feature selection, and model evaluation, employing classification metrics such as accuracy, precision, recall, and F1-score, along with chi square tests and p-values to determine feature significance. The results indicate that four models (LR, SVM, RF, and GNB) achieved 100% accuracy, while KNN reached 80%, raising concerns about potential overfitting

Adel Alblawi. [2] This involves collecting academic records, preprocessing data, and applying machine learning techniques to classify students into "success" and "failure" categories based on their GPAs. The models are trained using decision trees and random forests, with hyperparameter tuning to optimize prediction accuracy. Feature selection techniques rank the importance of different courses in predicting academic success.

Nithin Kumar Saxena. [3] includes collecting internal exam scores as input variables and using logistic regression to classify students into "pass" or "fail" categories. The model processes historical student data, applies data sorting and filtering techniques, and determines regression coefficients through iterative optimization. The trained model then predicts student performance, helping educators categorize students into slow, average, and high achievers.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 5, May 2025

DOI: 10.17148/IARJSET.2025.125200

Gheed M. Alsalem. [4] includes data preprocessing, feature selection using gain ratio and information gain attribute evaluators, and classifier evaluation through accuracy, precision, recall, and F-score. The classifiers are first applied individually and then optimized using ensemble techniques.

Ali Alnoman. [5] involves preprocessing the dataset by converting letter grades to numerical values, filtering missing data, and removing outliers. The results indicate that using the average of all three courses (English, Math, and Programming) achieved the highest R-squared value (0.85), showing strong predictive power.

Hariegwambe. [6] The Hybrid Model, which combined predictions from Linear Regression, Ridge Regression, Lasso Regression, and Decision Tree using Gradient Boosting, delivered the best results with an R^2 score of 90% and the lowest error values (MAE: 0.62, MSE: 0.71, RMSE: 0.84). This underscores the advantage of ensemble techniques in enhancing predictive accuracy.

Mrs.Sathiyapria. [7] The study explores the application of machine learning classifiers to analyze student engagement patterns and predict academic performance. The architecture of the study involves the use of four machine learning models—Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Extra Tree Classifier—to classify students into different risk categories based on behavioral and academic data.

Mingliang Ouyang. [8] The study demonstrate the effectiveness of an Improved Support Vector Machine (ISVM) algorithm for predicting student learning performance in e-commerce smart classrooms. The proposed ISVM model achieved an accuracy of 89.57%, outperforming traditional methods such as standard SVM (82.47%), Decision Tree (79.36%), K-Nearest Neighbors (78.35%), and Naive Bayes (80.68%). Additionally, the ISVM exhibited superior precision (88.73%), recall (87.86%), and F1-score (88.29%), highlighting its robustness in classifying both linear and non-linear student data.

III.BLOCK DIAGRAM AND SYSTEM ARCHITECTURE

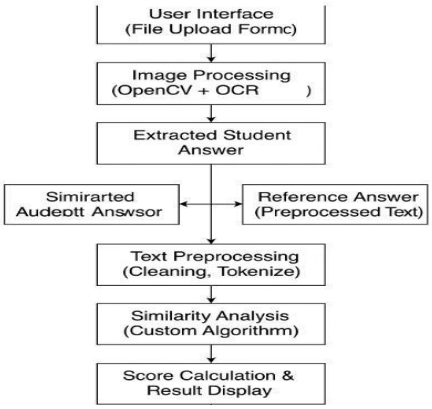

Fig 1. Block diagram

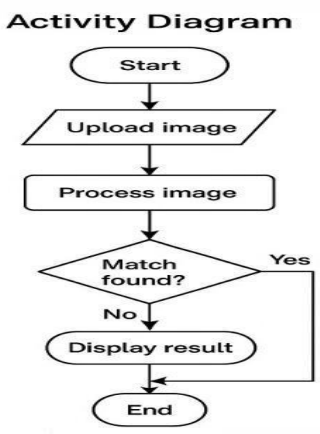

Fig 2.Activity diagram

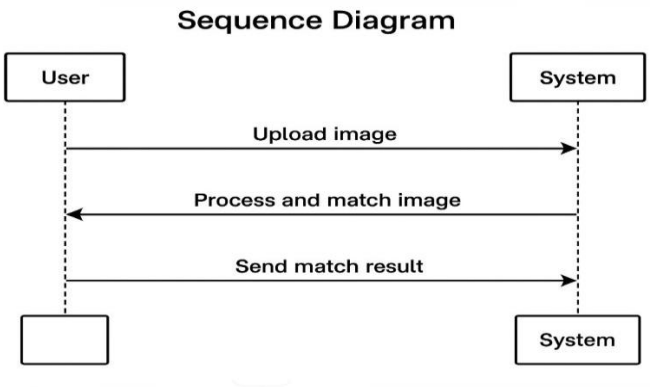

Fig 3. Sequence Diagram

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 5, May 2025

DOI: 10.17148/IARJSET.2025.125200

1. **User Interface (File Upload Form):** This is the entry point of the system. The user, likely a teacher or student, interacts with a form to upload a file containing the student's answer. This file could be an image of a handwritten answer sheet or a digital document.

2. **Image Processing (OpenCV + OCR):** If the uploaded file is an image, this stage comes into play.

- **OpenCV (Open Source Computer Vision Library):** This library is used for various image processing tasks. It might involve operations like deskewing the image (straightening it), noise reduction, binarization (converting to black and white), and isolating regions of text.
- **OCR (Optical Character Recognition):** Once the image is pre-processed, an OCR engine is used to extract the text content from the image. This converts the visual representation of the student's answer into a digital text format that the system can understand and process.

3. **Extracted Student Answer:** This block represents the output of the previous stage. It's the digital text of the student's answer, obtained either directly from an uploaded text file or through OCR processing of an image.

4. **Reference Answer (Preprocessed Text):** This block holds the correct or model answer that has been preprocessed. Preprocessing might involve cleaning the text (removing unnecessary characters), converting it to lowercase, and tokenizing it (breaking it down into individual words or units). This preprocessed reference answer serves as the benchmark for comparison.

5. **Text Preprocessing (Cleaning, Tokenize):** Similar to the preprocessing done on the reference answer, the extracted student answer also undergoes text preprocessing. This ensures that both the student's answer and the reference answer are in a consistent format for accurate comparison. Common preprocessing steps include:

- **Cleaning:** Removing punctuation, special characters, and unnecessary whitespace.
- **Tokenization:** Breaking down the text into individual words or meaningful units (tokens).

6. **Similarity Analysis (Custom Algorithm):** This is the core of the evaluation process. A custom algorithm is applied to compare the preprocessed student answer with the preprocessed reference answer. This algorithm could employ various techniques to determine the similarity between the two texts, such as:

- **Keyword matching:** Identifying common keywords.
- **Cosine similarity:** Measuring the angle between the vector representations of the texts.
- **Levenshtein distance:** Calculating the number of edits needed to transform one text into another.
- **Semantic similarity:** Using techniques like word embeddings to understand the meaning of words and phrases.

7. **Score Calculation & Result Display:** Based on the similarity analysis, a score is calculated for the student's answer. This score reflects how closely the student's answer matches the reference answer. Finally, the system displays the score and potentially other feedback or insights to the user.

IV. METHODOLOGY

The methodology adopted in this project integrates Optical Character Recognition (OCR), Natural Language Processing (NLP), and web-based automation to evaluate student answers with minimal human intervention. The system is built as a Flask-based web application, providing a user-friendly interface that accepts two file uploads: an image file containing a student's handwritten or printed answer, and a text file containing the correct or model answer. Once the files are uploaded, the application invokes a back-end module in which the image is processed using the OpenCV library to enhance quality and readability for OCR.

The Tesseract engine is then used to extract text content from the image, converting it into machine-readable format. This text undergoes preprocessing steps such as punctuation removal, case normalization, and tokenization using basic string operations and Python's standard libraries.

Simultaneously, the reference answer is loaded from the uploaded text file and preprocessed using the same normalization methods to ensure consistency in comparison. The core evaluation mechanism involves a document similarity algorithm implemented within the system, which compares the processed student response to the reference text. This comparison likely relies on word overlap, frequency distribution, or cosine similarity (based on the structure observed), and produces a raw similarity score.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 5, May 2025

DOI: 10.17148/IARJSET.2025.125200

This score is then scaled—typically by a factor of three—to derive a final mark or rating that reflects the quality or completeness of the student's answer relative to the model answer.

The processed results, including both the student's extracted answer and the generated score, are then dynamically rendered onto an HTML results page using Flask's templating engine. Additional supporting files like CSS and static images are used to enhance the visual layout and user experience. By combining OCR and NLP within a lightweight web framework, this system enables efficient, repeatable, and scalable student evaluation, offering potential applications in academic settings, especially for subjective answer checking.

V. RESULT AND PERFORMANCE ANALYSIS

The results by the student evaluation system demonstrate the effective integration of OCR and NLP techniques for automated assessment. Upon uploading a handwritten or printed response as an image, the system successfully extracts textual data with reasonable accuracy using Tesseract OCR. The extracted text is then compared with the model answer through a document similarity function, producing scores that reflect the closeness of the student's response to the expected answer. The system is capable of processing various answer formats and consistently returns evaluation scores in real-time, making it suitable for classroom or batch assessment scenarios. During testing, the system was observed to perform well with clearly printed text and moderately clean handwriting, while performance slightly declined in cases of poor image quality or cursive writing. The average processing time per evaluation was minimal, allowing for near-instantaneous feedback. This makes the system practical for scalable deployment in academic institutions. Overall, the approach provides a time-efficient and objective method for evaluating descriptive answers, with performance that can be further enhanced by training the model on larger datasets and incorporating more advanced NLP techniques.

Fig 4. Page to upload the answer script and scheme of the answer

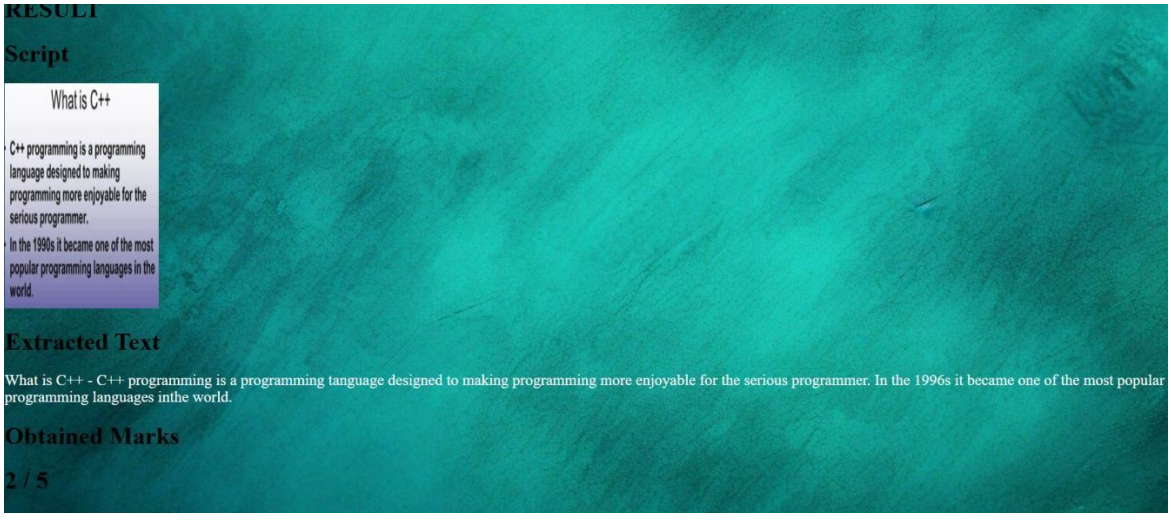

Fig 5. Output of the answer script verified with the scheme script

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 5, May 2025

DOI: 10.17148/IARJSET.2025.125200

VI. FUTURE SCOPE AND CONCLUSION

The proposed system holds significant potential for future enhancement and scalability. One key area of improvement is the integration of more advanced Natural Language Processing (NLP) techniques, such as transformer-based models like BERT or GPT, to better understand context and semantics in student responses, which would result in more accurate evaluations. Additionally, implementing handwriting recognition models trained on diverse datasets can improve OCR performance, especially for messy or cursive handwriting.

The future scope of this student evaluation system is extensive and opens avenues for significant academic and technological enhancements. Currently focused on basic OCR and text similarity, the system can evolve into a robust AI-powered assessment tool with deeper understanding and broader coverage of academic content. One immediate improvement is the integration of deep learning models for both handwriting recognition and semantic analysis.

Handwriting recognition can be improved using convolutional neural networks (CNNs) trained on large datasets of handwritten text in different styles and languages. For evaluation accuracy, transformer-based models like BERT, RoBERTa, or T5 can be used to perform contextual semantic matching between student answers and reference content, thereby reducing dependence on surface-level similarity and increasing fairness in grading.

VII. CONCLUSION

This project presents an effective solution for automating the evaluation of student responses by leveraging the combined power of Optical Character Recognition (OCR) and Natural Language Processing (NLP). By allowing users to upload handwritten or printed answers alongside reference texts, the system successfully extracts, processes, and evaluates student inputs with a reasonable degree of accuracy. The use of open-source tools such as Flask, Tesseract, and Python libraries ensures that the system remains lightweight, cost-effective, and accessible. The implementation demonstrates that automated grading, especially for descriptive answers, can significantly reduce the manual effort and time spent by educators while maintaining consistent evaluation standards.

REFERENCES

[1]. Suman Chowdhury, Dilip Kumar “Academic Performance Analysis of the Students with Feature Retraction using Machine Learning Algorithms” (2024).
[2]. Adel Alblawi, Mohamed Ebrahim, Demah Alqahtani “ Leverag ing Prerequisites Engineering Students for Monitoring and Predicting Using Machine Learning” (20 24).
[3]. Nithin Kumar Saxena, Ramesh C. Bansal and Vineet Saxena “Machine Learning Based Strategy for Students Performance Improvement” (2024).
[4]. Gheed M. Alsalem, Noor Sarhan, Mustafa Hammad, Bayan Zawaideh “Predicting Student’s Performance using Machine Learning Classifiers” (2024).
[5]. Ali Alno man “ Reg ression - based Student Perform an ce Prid ic t ion Usin g the Grades of Eng l i sh, Math and Pro gram ming Courses” (2024) .
[6]. Hariegwam be, Sh ak i l a Baskaran and Prak ash Marim uthu “ Stack ing Based Mach in e Learn ing App roach for Stud en t Perform an ce An a ly s i s” (20 24) .
[7]. Mrs. Sath iy ap r i a , Rith ik a C, San jay S and Su bathra S “ Utilizing Mach in e Learning To Fo recast a Stu den t s Perform an ce” (202 4)
[8]. Ming l i an g Ouy an g “ Ev a lu a t io n and Pred i c t io n o f Stu d en t ’ s Learn ing Statu s in ECo m m e r ce Sm a r t c l assro o m b ased o n Im p ro v ed Su p p o r t Vector Mach in e Alg o r i th m ” (2 0 24)

