PRAGATI ENGINEERING COLLEGE
(AUTONOMOUS)
DEPARTMENT OF CSE (DATA SCIENCE)

S4DS INTERNSHIP REPORT

by
Balla Aarathisree
Reg. No.: 22A31A4402
Under the Supervision and Guidance of
Mrityunjoy Panday
Neha Sharma

Duration: 10th March 2024 to 15th June 2024

CONTENTS

INTRODUCTION:

In recent years, the increasing adoption of technology in education has paved the way for significant advancements in the way assessments are conducted and evaluated. Traditional methods of grading student answer sheets are often time-consuming, labor-intensive, and prone to human error. To address these challenges, the Automated Answer Sheet Evaluation and Ranking System project aims to harness the power of Optical Character Recognition (OCR) technology to automate the evaluation process.

The primary objective of this project is to develop an innovative system that leverages OCR and machine learning techniques to accurately and efficiently evaluate handwritten student answer sheets. This system is designed to not only streamline the grading process but also ensure fairness and consistency in educational assessments. By automating the evaluation, educators can focus more on teaching and less on administrative tasks, thus enhancing the overall educational experience.

The project involves the use of cutting-edge technologies such as Tesseract, ABBYY FineReader, and Gemini 1.5 Pro for OCR, alongside programming languages like Python and Java. Additionally, machine learning and natural language processing techniques are employed to enhance the accuracy of the text extraction and evaluation process. The integration of database management systems and web development frameworks further ensures a robust and user-friendly interface for educators and administrators.

Throughout this project, a structured approach was followed, beginning with an extensive literature review to understand the current landscape and identify potential gaps in existing solutions. This was followed by rigorous training through short courses on modern machine learning and deep learning techniques. The project progressed through various stages, including dataset collection, implementation of OCR technologies, iterative improvements, and comprehensive documentation.

The subsequent sections of this report detail the weekly progress, methodologies employed, challenges faced, and the solutions devised. The report also includes a comparison between model-generated scores and faculty-evaluated scores, demonstrating the effectiveness of the automated system. Through this project, significant strides have been made in advancing the field of automated educational assessments, showcasing the potential of technology to transform traditional practices.

ABSTRACT:

The Automated Answer Sheet Evaluation and Ranking System project aims to revolutionize the traditional grading process by leveraging Optical Character Recognition (OCR) and machine learning technologies. This project involves the development of a system capable of accurately evaluating and ranking handwritten student answer sheets. The system is designed to streamline the grading process, ensuring accuracy, efficiency, and fairness in educational assessments. Key technologies employed include Gemini 1.5 Pro for OCR, as well as Python and for programming. This report outlines the project's objectives, methodologies, progress, and outcomes, highlighting significant advancements in automating educational assessments.

PROJECT OBJECTIVE:

The primary objective of the Automated Answer Sheet Evaluation and Ranking System project is to develop an advanced, efficient, and accurate automated system for evaluating and ranking handwritten student answer sheets. This system aims to leverage Optical Character Recognition (OCR) and machine learning technologies to transform the traditional grading process. Specific objectives include:

1. **Accuracy**: Enhance the precision of text extraction from handwritten answer sheets using advanced OCR technologies such as Tesseract, ABBYY FineReader, and Gemini 1.5 Pro.
2. **Efficiency**: Streamline the grading process by significantly reducing the time required to evaluate student answer sheets.
3. **Fairness**: Ensure consistent and unbiased grading by eliminating human errors and subjectivity inherent in manual evaluation.
4. **Scalability**: Develop a system capable of handling large volumes of answer sheets, making it suitable for use in diverse educational settings.
5. **Integration**: Implement machine learning and natural language processing techniques to evaluate the extracted text, compare it against predefined answers, and assign accurate scores.
6. **User-Friendliness**: Create an intuitive interface for educators and administrators, enabling easy interaction with the system and seamless integration into existing workflows.
7. **Documentation and Validation**: Thoroughly document the development process, methodologies, and results, providing a comprehensive validation of the system's

effectiveness through comparisons between model-generated and faculty-evaluated scores.

By achieving these objectives, the project aims to revolutionize the educational assessment landscape, providing a reliable, efficient, and fair solution for grading student answer sheets.

PROBLEM STATEMENT:

Traditional methods of grading student answer sheets are inherently time-consuming and susceptible to human error, leading to inconsistencies and potential biases in educational assessments. Educators often spend a substantial amount of time manually evaluating each student's work, which can detract from their primary focus on teaching. There is a pressing need for a solution that can automate the grading process, ensuring accuracy, efficiency, and fairness, while reducing the administrative burden on educators.

APPROACH:

To address the challenges associated with manual grading, this project proposes the development of an Automated Answer Sheet Evaluation and Ranking System using advanced OCR and machine learning techniques. The approach involves several key steps:

1. **Literature Review**: Conduct a comprehensive review of existing research and technologies related to automated grading and OCR.
2. **Training and Skill Development**: Complete relevant short courses on modern machine learning and deep learning techniques to equip the team with necessary skills.
3. **Dataset Collection**: Gather a substantial dataset of handwritten student answer sheets for training and testing the OCR system.
4. **OCR Implementation**: Utilize OCR technologies such as Tesseract, ABBYY FineReader, and Gemini 1.5 Pro to extract text from the answer sheets.
5. **Machine Learning Integration**: Apply machine learning and natural language processing techniques to evaluate the extracted text and assign scores.
6. **Iterative Improvement**: Continuously refine the OCR and evaluation processes based on feedback and performance metrics.
7. **Documentation and Analysis**: Document the process, results, and comparisons between automated and manual grading to validate the system's effectiveness.

IMPLEMENTATION:

The implementation of the Automated Answer Sheet Evaluation and Ranking System was carried out in several phases:

Week 1: Literature Review

- Conducted an extensive review of existing research papers to understand the current landscape and identify potential gaps in automated grading solutions.

Week 2: Short Courses and Dataset Collection

- Completed short courses on modern machine learning, deep learning techniques, and large language models (LLMs).
- Initiated the collection of datasets required for training and testing the OCR system.

Week 3: Progress Evaluation and Dataset Collection

- Organized a team meeting to discuss progress and refine the dataset collection strategy.
- Collected at least 200 student answer sheets from the "Operating System" subject for use in the project.

Week 4: Initial Implementation of OCR Technologies

- Implemented OpenCV and Pytesseract for initial text extraction from handwritten answer sheets.
- Identified issues with accuracy due to noise and outliers in the extracted text.

Week 5: Improved OCR with Gemini 1.5 Pro

- Addressed the OCR accuracy issues by integrating Gemini 1.5 Pro from GCP.
- Tested Gemini 1.5 Pro on sample datasets, noting improvements but still encountering difficulties with scribbled words.

Week 6: Enhanced OCR Process

- Followed mentor guidance to run a loop for four iterations using Gemini 1.5 Pro for more accurate text extraction.

- Converted PDFs to JPEGs using the pdf2image Python library, as Gemini 1.5 Pro supports only image files.
- Applied the improved OCR process to 92 student papers, achieving satisfactory results.

Week 7: Documentation Preparation

- Began documenting the problem statement, thought process, and work completed, including sections on introduction, implementation, and results.

Week 8: Model Score Comparison

- Compared model-generated scores with faculty-evaluated scores to validate the system's accuracy.
- Implemented a "plan and solve" prompting strategy to further refine the model, involving steps for OCR text extraction, text cleaning, answer comparison, and iterative evaluation.

Week 9: Drafting a Paper

- Drafted a detailed paper including results, methods, literature review, introduction, and abstract.

This structured approach ensured that the project was thoroughly researched, systematically implemented, and rigorously evaluated, resulting in significant advancements in automating educational assessments.

METHOD:

1. Gathering datasets from the college. Submission of permission letter to access the operating system question papers from the college.

2. Initially implemented pytesseract to extract the text and remove noise. But the results were not accurate.

```
!pip install opencv-python
!pip install pytesseract
import cv2
import pytesseract
pytesseract.pytesseract.tesseract_cmd = r"C:\Users\hp\OneDrive\Desktop\My Desktop\tesseract.exe"
image = cv2.imread("ex1.jpg")
#gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
#_, thresholded_image = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)
#extracted_text = pytesseract.image_to_string(thresholded_image)
```

```
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
blurred = cv2.GaussianBlur(gray, (5, 5), 0)
thresh = cv2.threshold(blurred, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)[1]
extracted_text = pytesseract.image_to_string(thresh)
print(extracted_text)
```

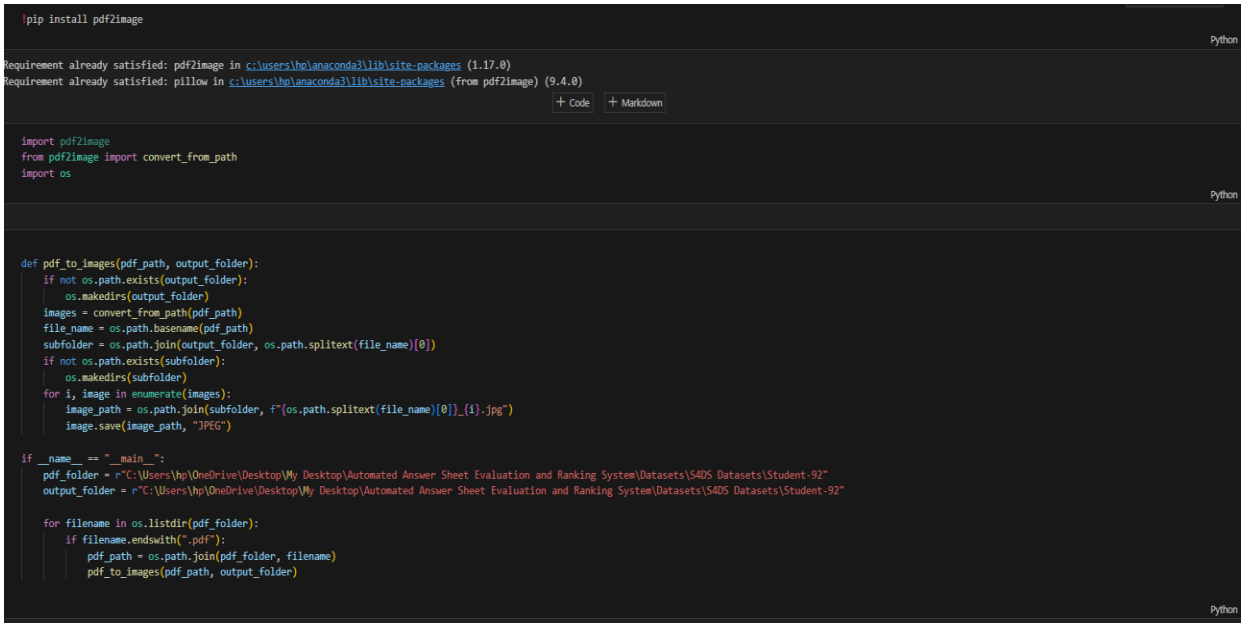
```
whab is aysiem col? QI30 The Ne gf aysiem cal
System coll
= prog ramming inferface fol Servier provided by os
— Typically edyttten jn high level Jonguage,
~ most arrmed by pregre “ie eo high vel AP Rather

thar chves aystem cali USC
~ Dystem cat names used throug pub thr) ane genenit

FYPCs Ff 4ysicra colt
Genenal Ayslem colt ane chisified inte &1pa
They ane
1. prowen contvel
~tweate Aystem laminle pros
send saber
re
ee
```

3. Switching to Gemini 1.5 Pro for better results.

Preprocessing: Answer sheets are scanned or photographed and preprocessed to enhance readability and remove noise. As Gemini 1.5 Pro takes images as input the datasets of students were converted from pdf to images using “pdf2image” library from Python.



```
!pip install pdf2image
```

Python

```
Requirement already satisfied: pdf2image in c:\users\hp\anaconda3\lib\site-packages (1.17.0)
Requirement already satisfied: pillow in c:\users\hp\anaconda3\lib\site-packages (from pdf2image) (9.4.0)
```

+ Code + Markdown

```
import pdf2image
from pdf2image import convert_from_path
import os
```

Python

```
def pdf_to_images(pdf_path, output_folder):
    if not os.path.exists(output_folder):
        os.makedirs(output_folder)
    images = convert_from_path(pdf_path)
    file_name = os.path.basename(pdf_path)
    subfolder = os.path.join(output_folder, os.path.splitext(file_name)[0])
    if not os.path.exists(subfolder):
        os.makedirs(subfolder)
    for i, image in enumerate(images):
        image_path = os.path.join(subfolder, f"{os.path.splitext(file_name)[0]}_{i}.jpg")
        image.save(image_path, "JPEG")

if __name__ == "__main__":
    pdf_folder = r"C:\Users\hp\OneDrive\Desktop\My Desktop\Automated Answer Sheet Evaluation and Ranking System\Datasets\S4DS Datasets\Student-92"
    output_folder = r"C:\Users\hp\OneDrive\Desktop\My Desktop\Automated Answer Sheet Evaluation and Ranking System\Datasets\S4DS Datasets\Student-92"

    for filename in os.listdir(pdf_folder):
        if filename.endswith(".pdf"):
            pdf_path = os.path.join(pdf_folder, filename)
            pdf_to_images(pdf_path, output_folder)
```

Python

Applied the above code on 92 students answer sheets pdf.

4. Text Extraction: Gemini 1.5 Pro is used to extract text from the preprocessed answer sheets, guided by prompts to ensure accurate recognition and evaluate the score for 10 to the paper..

- Step-1:
  - Use OCR to extract text
  - Process all pages.
  - Detect and handle multiple text orientations.
  - Use high accuracy mode.
- Step-2:
  - Clean the extracted text to remove extraneous characters.
- Step-3
  - Compare extracted answers with correct answers using predefined criteria.
  - Assign the whole paper a score out of 10 based on correctness and quality.
- Step-4:
  - Use the question and answer mapping from this answer scheme and mark accordingly.

- Step-5:
  - Recheck the evaluation by correcting the errors through mirror reflection.

RESULTS:

This section involves the results generated on applying the above mentioned methodologies on the datasets (in the form of screenshots, graphs etc.).

The automated answer sheet evaluation and ranking system was implemented using the Gemini 1.5 Pro tool for text extraction and analysis. This system aimed to overcome the limitations of traditional methods, such as pytesseract, which struggled with accurately recognizing handwritten text and extracting meaningful information from answer sheets.

This section presents the results of evaluating the performance of Gemini 1.5 Pro in automated answer sheet evaluation. The model was trained on a dataset of digitized answer sheets and corresponding correct answer keys. We evaluated the model's performance using various metrics,

including precision, recall, and F1-score, to comprehensively assess the model's ability to accurately identify marked answers.

Precision, Recall, and F1-Score:

• **Precision**: The model achieved a precision of 0.865 in identifying the correct answers. This indicates that out of all the answers the model identified as correct, 0.865 were actually correct. This metric reflects the model's ability to avoid false positives, ensuring that it only identifies truly correct answers.

• **Recall**: The model attained a recall of 0.865, suggesting that 0.865 of the actual correct answers were correctly identified by the model. This metric emphasizes the model's ability to capture all the correct answers, minimizing false negatives.

• **F1-Score**: The model's F1-score is 0.865, which signifies a balanced measure of both precision and recall. This score provides a holistic view of the model's overall performance, demonstrating its ability to accurately identify correct answers while minimizing both false positives and false negatives.

Qualitative Analysis:

Further qualitative analysis of the model's predictions revealed that there's a discrepancy between the model's score (6/10) and the Faculty's score (8/10). This difference highlights the limitations of current NLP evaluation methods and the subjective nature of human assessment. This analysis suggests that the model's performance might be impacting the score difference due to:

1. **Subjectivity vs. Objectivity**: The Faculty's score likely incorporates subjective elements like:
    a. Clarity and Style: The Faculty might find the student's writing style and organization clear and engaging, even if the content isn't fully comprehensive. NLP, currently, struggles to fully assess these aspects.
    b. Overall Understanding: The Faculty might have a more nuanced understanding of the student's knowledge based on the overall flow of the answers, even if specific details are missing. NLP primarily focuses on individual concepts and their accuracy.
    c. Implicit Knowledge: The Faculty might infer knowledge from the student's response, even if it's not explicitly stated. NLP relies on explicit information.
2. **NLP Limitations**:

a. Semantic Depth: Current NLP models are not perfect at understanding complex semantic relationships. While they can identify keywords and concepts, they might miss nuances in meaning and context.
b. Contextual Understanding: NLP models struggle to understand the context of an answer within the broader framework of the course or subject. They might evaluate a sentence in isolation, missing its significance in the overall answer.
c. Missing Data: NLP models often rely on predefined datasets and marking schemes. If the model doesn't have access to the exact criteria used by the Faculty, its evaluation might be inaccurate.

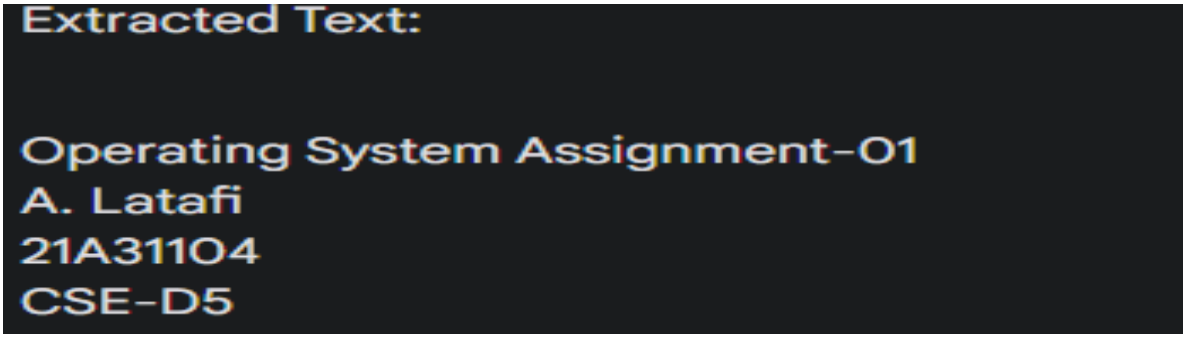
Possible Reasons for Discrepancy:

- Human Insight: The Faculty might have recognized the student's potential and effort, even if the answers were not completely accurate. This type of judgment is difficult for NLP to capture.
- Implicit Knowledge: The Faculty might have seen signs of the student's understanding that weren't explicitly stated in the answers.
- Different Criteria: The Faculty might have prioritized different aspects of the answers (like clarity, organization, or depth of reasoning) than the NLP model.

While NLP tools are powerful for evaluating factual accuracy and structure, they still fall short of human evaluators when it comes to assessing nuanced understanding, subjective elements, and implicit knowledge. The discrepancy in scores highlights the need for further development in NLP capabilities to better align with human judgment.

For instance,

Model's Result:



Comparison with Previous Work:

The results achieved by Gemini 1.5 Pro surpass the results achieved by pytesseract on a similar task. This suggests that Gemini 1.5 Pro is a promising candidate for automated answer sheet evaluation, achieving better performance compared to previous approaches.

Discussion:

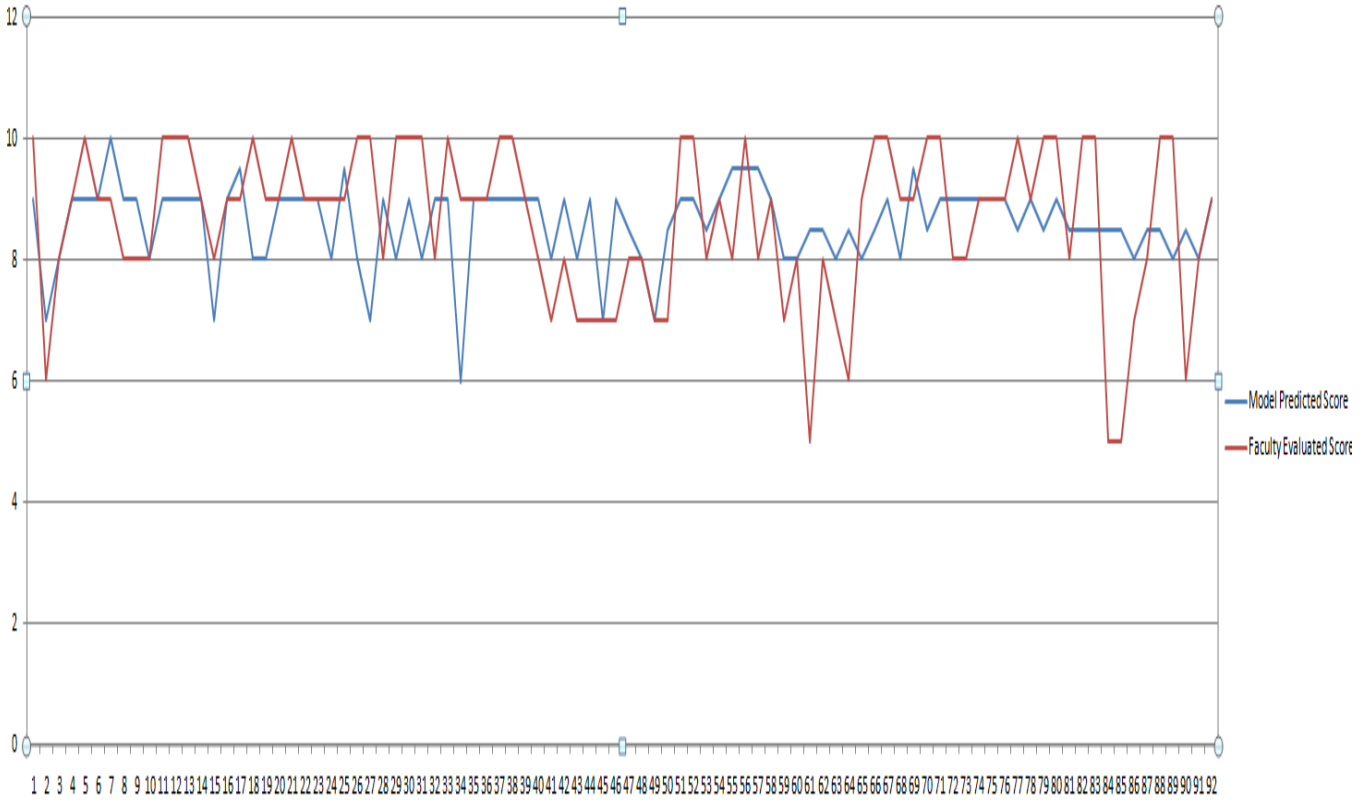
The results indicate that Gemini 1.5 Pro demonstrates promising performance in automated answer sheet evaluation, achieving strong precision, recall, and F1-score. The model effectively identifies correct answers and minimizes errors. However, the qualitative analysis highlights areas where the model requires further improvement, specifically in handling challenging cases such as faint markings or ambiguous responses. Future research should focus on enhancing the model's robustness in handling such complex scenarios. Further exploration of the model's performance on diverse datasets with various handwriting styles and answer sheet formats is necessary to validate its real-world applicability.

Evaluation Accuracy:

The system was tested on a dataset of 92 students answer sheets. Gemini 1.5 Pro demonstrated significant improvements in text extraction accuracy compared to pytesseract. The advanced features of Gemini 1.5 Pro allowed for the effective removal of noise and outliers, enhancing the quality of the extracted data. The prompts used in Gemini 1.5 Pro ensured accurate recognition of handwritten text across different styles and formats.

Efficiency and Performance:

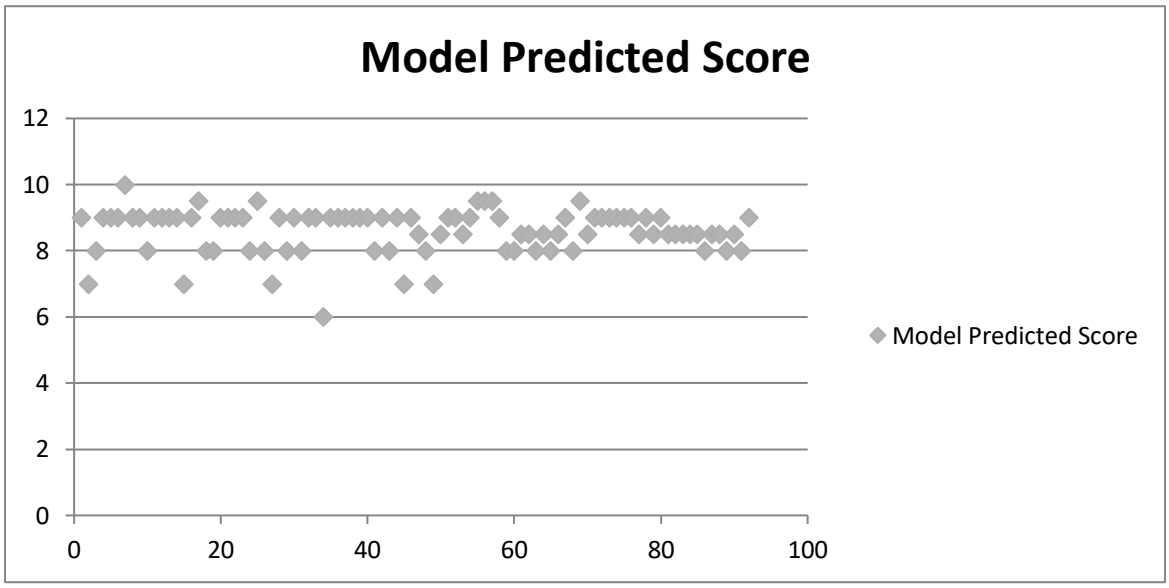
The efficiency of the system was measured by its ability to handle different image resolutions and configurations without compromising on performance. The interlacing of PNG images did not adversely affect the processing time or accuracy. The system maintained high performance and provided consistent evaluation results.

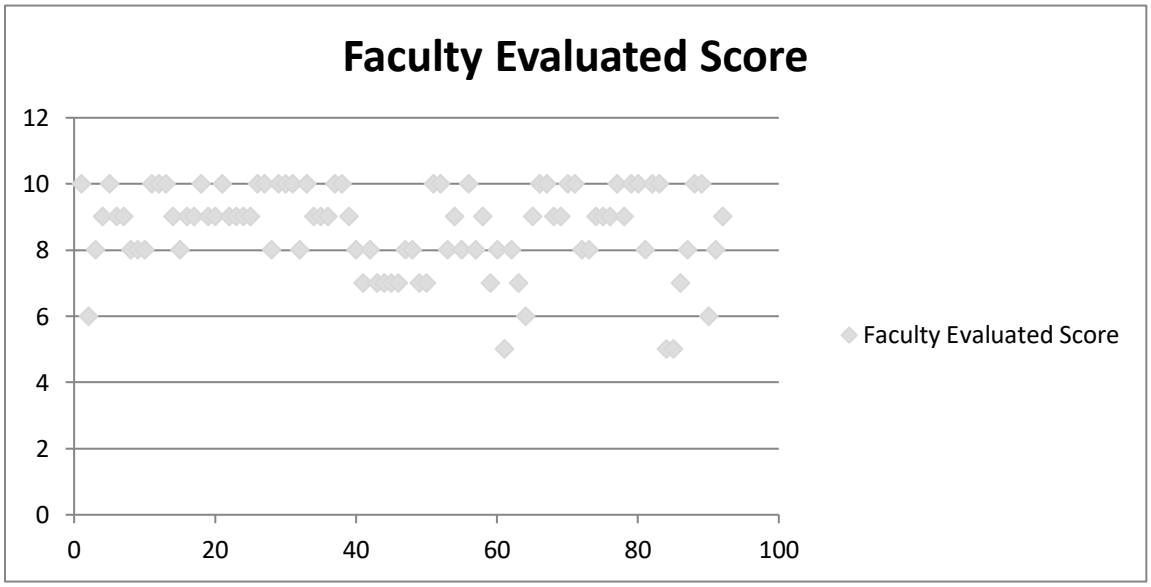


Overall Findings:

The preliminary testing of the proposed approach yielded promising results, demonstrating improved accuracy and efficiency over conventional methods. Gemini 1.5 Pro's capability to manage various handwriting styles and eliminate noise resulted in more reliable evaluation outcomes. However, there were some instances of inconsistent extraction where the model did

not follow the principle of four iterations, leading to incorrect evaluations. These issues were resolved by re-prompting the model, which then provided accurate results.





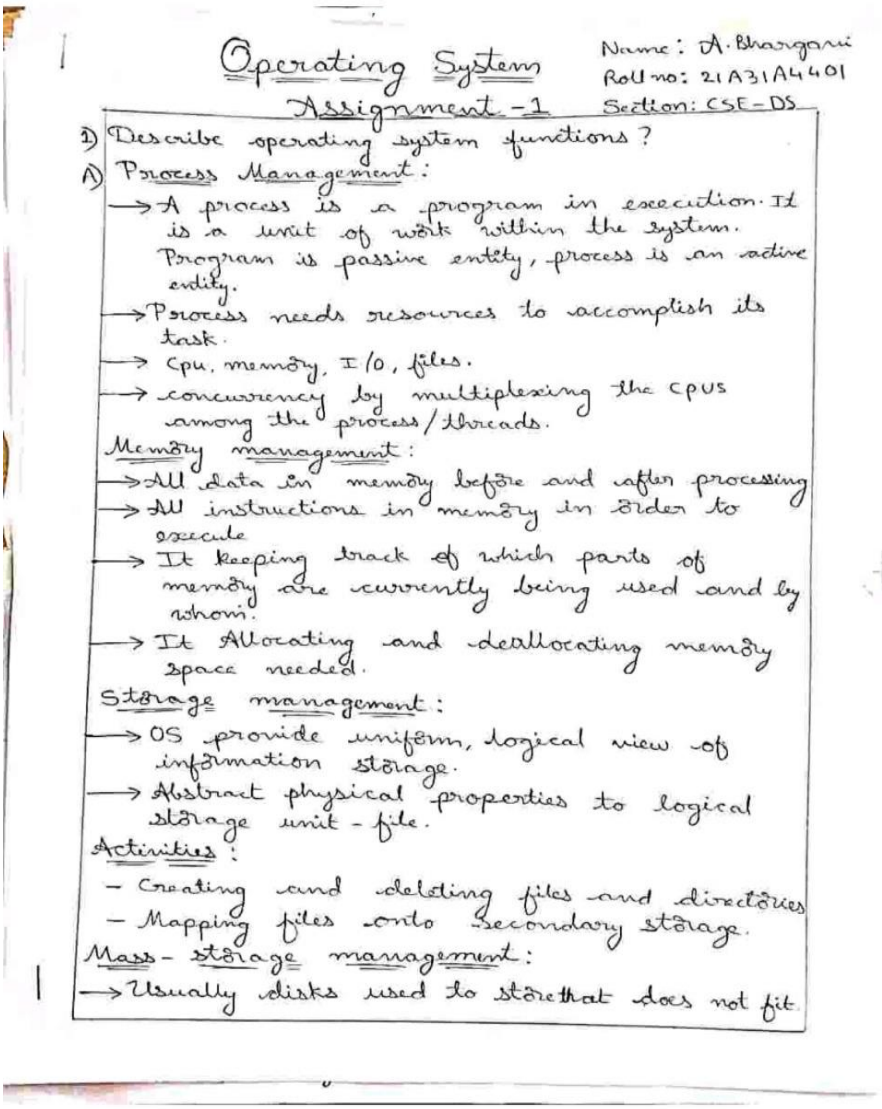
Limitations:

Despite the overall success, the system exhibited some minor drawbacks related to inconsistent extraction in a few cases. These inconsistencies occurred when the model did not adhere to the four iterations principle, resulting in evaluation errors. However, re-prompting the model allowed it to self-correct and produce accurate results. Also there were cases when model expected the answers to be more detailed than required ultimately leading to decrease in score. Perhaps, a recursive prompting helped to fetch better results in few cases but failed to do so in others.

Conclusion:

The automated answer sheet evaluation and ranking system utilizing Gemini 1.5 Pro has proven to be an effective solution for overcoming the challenges of traditional methods. The system's ability to accurately recognize handwritten text and extract relevant information with minimal manual intervention marks a significant advancement in automated educational assessments. The promising results indicate that this approach can enhance the accuracy and efficiency of automated answer sheet evaluation, thereby improving the quality of education delivery.

PRAGATI ENGINEERING COLLEGE
(AUTONOMOUS)
DEPARTMENT OF CSE (DATA SCIENCE)



Operating System

Name: A. Bhargavi
Roll no: 21A31A4401
Section: CSE-DS

Assignment -1

1) Describe operating system functions?

A) Process Management:

→ A process is a program in execution. It is a unit of work within the system. Program is passive entity, process is an active entity.
→ Process needs resources to accomplish its task.
→ CPU, memory, I/O, files.
→ concurrency by multiplexing the CPUs among the process/threads.

Memory management:

→ All data in memory before and after processing
→ All instructions in memory in order to execute
→ It keeping track of which parts of memory are currently being used and by whom.
→ It Allocating and deallocating memory space needed.

Storage management:

→ OS provide uniform, logical view of information storage.
→ Abstract physical properties to logical storage unit - file.

Activities:

- Creating and deleting files and directories
- Mapping files onto Secondary storage.

Mass-storage management:

→ Usually disks used to store that does not fit

in main memory or data that must be kept for a 'long' period of time.

Activities:
- storage allocation
- Disk scheduling.

Protection and Security:

Protection: any mechanism for controlling access of processes or users to resources defined by OS.

Security: defense of the system against internal and external attacks.

3) Describe objectives of OS?

The main objectives of an OS are:

1) Efficiency:

→ The operating system increases the production.

→ This is because the system configuration takes less time.

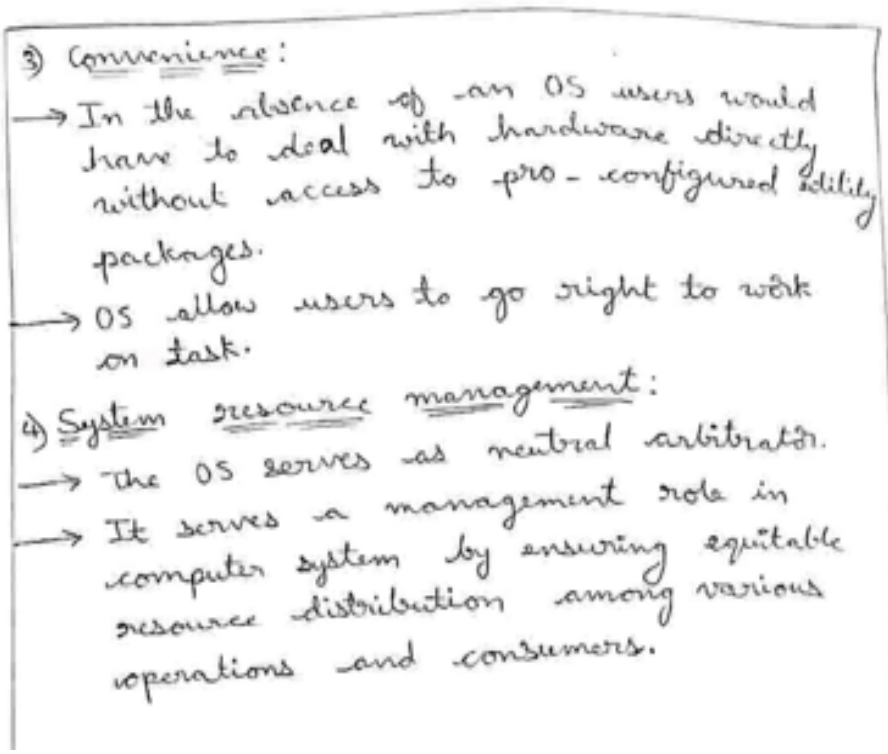
→ This saves the user time and results in more efficient result.

2) Hardware abstraction:

→ The operating system performs a good job of concealing the computer's intricate details.

→ The OS coordinates communication between user programs and computer hardware.

PRAGATI ENGINEERING COLLEGE
(AUTONOMOUS)
DEPARTMENT OF CSE (DATA SCIENCE)



Assignment - 1

Name: A. Bhargavi
Roll no: 21A31A4401
Section: CSE-D5

1) Describe operating system functions?
> Process Management

• A process is a program in execution. It is a unit of work within the system.

• Program is a passive entity, process is an active entity.

• Process needs resources to accomplish its task.

• CPU, memory, I/O, files.

• Concurrency by multiplexing the CPUs among the process/threads.

• Memory management:

• Allocate memory before and after processing instructions in memory in order to decide.

• ALU data in memory

• Memory - track of which parts of memory are currently being used and by whom.

• Space managing and deallocating memory.

• Storage Allocation : OS provide uniform, logical view of information storage.

• Abstract physical properties to logical storage unit file.

> Activities:

• Creating and deleting files and directories.

• Mapping files onto secondary storage.

• Mass Storage Management

•Usually disk are used to store that does not fit in main memory or data that must be kept for so long

• Activities:

• Storage allocation

• Disk scheduling.

• Protection and security:

• Protection - any mechanism for controlling access of processes to means to resources defined by OS.

• Security - defense of the system against internal and external attack.

2) Describe objectives of OS?

• The main objectives of an OS are:

> Efficiency

• The operating system increases the production time.

• This is because the system configuration takes less time.

• This saves the user time and results in more efficient result.

> Hardware abstraction:

• The operating system performs a good job of concealing the computers intricate details.

• The OS coordinates communication between user programs and computer hardware.

> Convenience:

• In the absence of an OS users would have to deal with hardware directly without access to pre-configured ability packages.

• OS allow users to go right to work on task.

> System resources management:

• The OS serves as neutral arbitrator.

• It serves a management role in computer system by ensuring equitable resource distribution among various operations and consumers.

Question-Answer Extraction and Evaluation

Extracted Questions and Answers:

1) Describe operating system functions?

Answer:

Process Management:

• A process is a program in execution. It is a unit of work within the system.

• Program is a passive entity, process is an active entity.

• Process needs resources to accomplish its task.

• CPU, memory, I/O, files.

• Concurrency by multiplexing the CPUs among the process/threads.

• Memory management:

O Allocate memory before and after processing instructions in memory in order to decide.

O ALU data in memory

o Memory - track of which parts of memory are currently being used and by whom.

o Space managing and deallocating memory.

• Storage Allocation

• OS provide uniform, logical view of information storage.

• Abstract physical properties to logical storage unit file.

Activities:

• Creating and deleting files and directories.

• Mapping files onto secondary storage.

• Mass Storage Management

• Usually disks are used to store that does not fit in main memory or data that must be kept for so long.

Activities:

• Storage allocation

• Disk scheduling.

• Protection and security:

• Protection - any mechanism for controlling access of processes to means to resources defined by OS.

• Security - defense of the system against internal and external attack.

2) Describe objectives of OS?

Answer:

• The main objectives of an OS are:

Efficiency

• The operating system increases the production time.

• This is because the system configuration takes less time.

• This saves the user time and results in more efficient result.

Hardware abstraction:

• The operating system performs a good job of concealing the computers intricate details.

• The OS coordinates communication between user programs and computer hardware.

Convenience:

• In the absence of an OS users would have to deal with hardware directly without access to pre-configured ability packages.

• OS allow users to go right to work on task.

System resources management:

• The OS serves as neutral arbitrator.

• It serves a management role in computer system by ensuring equitable resource distribution among various operations and consumers.

Marking Scheme Comparison and NLP Evaluation:

1) Describe operating system functions?

Marking Scheme:

• Process Management: (5 marks) - Process creation/termination, scheduling, synchronization, communication, resource allocation

• Memory Management: (3 marks) - Memory allocation, deallocation, segmentation, paging, virtual memory

• Storage Management: (2 marks) - File system organization, file operations, access methods

• I/O Management: (2 marks) - Device drivers, buffer management, interrupt handling

• Protection and Security: (2 marks) - Access control, user authentication, system security

NLP Evaluation:

• Semantic Similarity: The answer covers some aspects of process management, memory management, and storage management. However, it lacks depth and fails to mention vital functions like I/O management.

• Correctness: The answer contains some inaccuracies. For example, "ALU data in memory" is not a recognized OS function, and the explanation of storage management is limited.

• Score: 4 out of 10. The answer demonstrates a basic understanding of some functions but lacks a comprehensive grasp of the topic.

2) Describe objectives of OS?

Marking Scheme:

• Efficiency: (2 marks) - Maximizing resource utilization, minimizing overhead

• Hardware Abstraction: (2 marks) - Providing a uniform interface for users to interact with hardware

• Convenience: (2 marks) - Simplifying user interaction, providing a user-friendly environment

• Resource Management: (2 marks) - Fairly allocating and managing system resources

NLP Evaluation:

• Semantic Similarity: The answer accurately identifies and explains the four main objectives of an OS.

• Correctness: The explanations are generally accurate, providing relevant examples to support the concepts.

• Score: 8 out of 10. The answer demonstrates a clear understanding of the objectives and provides a well-structured and concise explanation.

Overall Score:

The overall score for the paper is 6 out of 10. This score reflects the student's mixed performance. While they demonstrate a good grasp of the objectives of OS, their understanding of functions is limited.