CT Marks Finder - Version 1

Sazid Hasan

October 13, 2024

1 Overview

This Python application uses `tkinter` for a GUI-based approach to allow users (students or teachers) to input student roll numbers and select multiple PDF files containing class test (CT) marks. The program then extracts the student's CT marks from the PDFs, calculates class averages, generates a comparison plot (student's marks vs. class average), and saves the plot and merged PDFs into a new combined PDF file.

Key Features

- **Extract Marks for Students**: Extract marks from multiple PDFs for a specific student based on their roll number.
- **Calculate Class Averages**: Calculate average marks for each CT from all students' marks found in the PDFs.
- **Generate Plots**: Create line plots showing the student's marks and class averages for comparison.
- **Merge PDFs**: Combine the original PDFs and the generated plot into one PDF file.
- **GUI Interface**: The application provides a simple interface for user interaction using `tkinter`.

2 Code Structure

2.1 1. Dependencies

The program uses several libraries:

- **PyPDF2**: To extract text from PDF files and merge multiple PDFs.
- **matplotlib**: For plotting graphs.
- **NumPy**: To handle numerical operations such as calculating averages.
- **tkinter**: For the graphical user interface.

Listing 1: Library Imports

```
import tkinter as tk
from tkinter import filedialog, messagebox
import PyPDF2
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.backends.backend_pdf import PdfPages
```

2.2 2. Functions

2.2.1 extract_all_marks_from_pdf(pdf_file)

Purpose: Extracts the marks for all students in the PDF file.
Process:

- Opens the PDF file, extracts the text, and splits it line by line.
- For each line, it checks if it contains valid student data (a roll number followed by CT marks).
- Returns a list of marks for all students found in the file.

Listing 2: Function to Extract Marks of All Students

```
def extract_all_marks_from_pdf(pdf_file):
    ...
    return all_marks
```

2.2.2 extract_student_marks_from_pdf(pdf_file, roll_number)

Purpose: Extracts marks for a specific student based on the roll number.
Process:

- Scans the PDF, identifies the student's roll number, and extracts their CT marks (up to 4 CTs).
- Returns the list of marks for the student.

Listing 3: Function to Extract Specific Student Marks

```
def extract_student_marks_from_pdf(pdf_file, roll_number):
    ...
    return ct_marks
```

2.2.3 accumulate_student_ct_marks(pdf_files, roll_number)

Purpose: Accumulates the CT marks for a student from multiple PDF files.
Process:

- Iterates over each PDF and extracts the student's marks, accumulating them sequentially across files.

Listing 4: Function to Accumulate Student Marks Across PDFs

```
def accumulate_student_ct_marks(pdf_files, roll_number):
    ...
    return accumulated_marks
```

2.2.4 calculate_averages(all_students_marks)

Purpose: Calculates the average marks for each CT across all students.
Process:

- Converts the list of marks into a NumPy array and calculates the average for each CT.

Listing 5: Function to Calculate Class Averages

```
def calculate_averages(all_students_marks):
    ...
    return avg_marks
```

2.2.5 plot_marks(marks, avg_marks, roll_number)

Purpose: Generates a line plot comparing the student's CT marks to the class averages and saves it as a PDF.

Process:

- Plots the student's marks and the class averages for each CT in different colors.
- Saves the plot to a PDF file named `Marks_for_{roll_number}.pdf`.

Listing 6: Function to Plot Marks and Averages

```
def plot_marks(marks, avg_marks, roll_number):
    ...
    return pdf_filename
```

2.2.6 merge_pdfs(pdf_files, output_path)

Purpose: Merges multiple PDF files into one.

Process:

- Uses PyPDF2 to combine the student's results and the original PDFs into a single output file.

Listing 7: Function to Merge PDFs

```
def merge_pdfs(pdf_files, output_path):
    ...
```

2.3 3. GUI Interface

The GUI uses `tkinter` to create an interface for users to input data and select PDF files.

2.3.1 browse_files()

Purpose: Allows the user to select multiple PDF files using a file dialog.

Listing 8: File Browser

```
def browse_files():
    ...
```

2.3.2 find_marks()

Purpose: Main function that processes the selected PDF files, extracts marks for the target student, calculates averages, plots the results, and merges the PDFs.

Process:

- Verifies input fields (roll number and PDFs).
- Accumulates the student's marks from all PDFs.
- Extracts and averages the class marks.
- Calls the `plot_marks` function to generate and save the plot as a PDF.
- Merges the original PDFs and the generated plot into one final PDF.

Listing 9: Main Function to Find Marks

```
def find_marks():
    ...
```

2.4 Main Window Setup

The GUI window is set up using `tkinter`. It includes:

- **Roll Number Input**: An entry box for the user to input the roll number.
- **PDF File Selection**: A button to open the file dialog and select multiple PDF files.
- **Submit Button**: A button to start the process of extracting marks, calculating averages, and generating results.

Listing 10: Main Window Setup

```
root = tk.Tk()
root.title("CT Marks Finder")
...
root.mainloop()
```

3 How the Program Works

1. **Input Roll Number**: The user enters a roll number.
2. **Select PDFs**: The user selects one or more PDF files that contain class test (CT) marks.
3. **Process PDFs**: The program extracts the marks for the student and for all students in the class from the PDFs.
4. **Calculate Averages**: The class averages for each CT are calculated.
5. **Plot Results**: The student's marks are plotted against the class averages and saved to a PDF.
6. **Merge PDFs**: The original PDFs and the generated plot are merged into a final output PDF.
7. **Save Results**: The final PDF file is saved to the disk.

4 Error Handling

- **File Selection**: If no files are selected, the user is alerted with a message box.
- **Roll Number Input**: If no roll number is entered, the user is prompted to provide one.
- **Merging PDFs**: Errors that occur during the merging of PDFs are handled with a message box alert.

5 How to Use

1. Run the Program.
2. Enter the Roll Number of the student.
3. Select the PDF Files containing the CT marks.
4. Click "Find Marks".
5. The program will extract the marks, calculate class averages, and generate a plot.
6. The original PDFs and the plot will be combined into a single output PDF.

6 Limitations and Future Enhancements

- **Multiple CT marks**: Currently, the program can handle up to 4 CT marks per student.
- **More Advanced Merging**: Future versions can include more advanced PDF processing features.
- **Handling Missing Data**: Future versions may improve the handling of missing or inconsistent data in the PDFs.

7 Version 1 Conclusion

This version provides a simple and effective way to extract and visualize student CT marks from multiple PDF files, offering a clear comparison against class averages. The use of Python libraries like `PyPDF2`, `matplotlib`, and `tkinter` makes this tool flexible and easy to use in academic settings.