Development of a Tool for Quick Result Analysis

Mohit Sunil Pansare[1]; Gladys Gince Skariah[1]; Rhea Nilesh Bhalekar[1]; Garima Tripathi (Professor)[1]
[1]Department of Artificial Intelligence and Data Science, Mumbai University, Maharashtra, India
[1]Fr. Conceicao Rodrigues College of Engineering, Bandra, India

Abstract:- This research paper introduces the "Development of a tool for Quick Result Analysis for Mumbai University", a system created to provide detailed reports based on the outcomes of a given branch for a certain semester. Currently there are four branches of Engineering included in our system. The analyzer provides rapid and precise insights into student performance, grade distributions and comparative assessments across the courses by utilizing modern data processing techniques. Mumbai University's Engineering programme will see continual academic progress as students, teachers and administrators will have access to useful information that will help to improve academic performance and decision-making procedures.

Keywords:- *Result Analysis, Python, Report Generation, Template, Data Visualization.*

I. INTRODUCTION

In today's digital age, data plays a vital role in decision making, analysis, and various other domains. However, data comes in different formats and converting it into a more accessible and versatile format is often necessary.

The result analyzer system[1] retrieves raw result data for a specific branch and semester from the Mumbai University database and applies statistical algorithms to process and analyze the data. The generated reports provide key information such as the name of the branch, year of graduation, the selected semester and the class strength. Additionally, the system presents graphical representations including a CGPA graph which allows one to understand visually, the distribution of students' grades.

Moreover, the system provides student statistics for the entire class, which includes student percentage ranges, top three students based on CGPA, the maximum marks obtained in theory subjects, the number of students with a 10 CGPA, the number of students with KT's (failures), and the number of students scoring 60 in theory subjects.

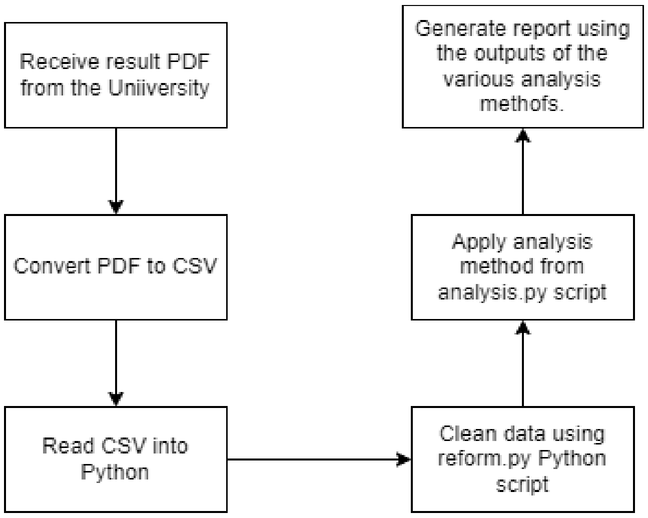
These insightful statistics allow students to assess their performance and understand their relative position within the class. Faculty members can utilize this information to identify subjects which may require additional support and tailor their teaching strategies accordingly. Administrators benefit from the system by gaining a comprehensive overview of a class performance and can identify trends and patterns that may inform the curriculum and assessment improvements.

Stakeholders at Mumbai University may make data-driven decisions to improve the quality of instruction and encourage student achievement by integrating these capabilities into the system. This technology makes it possible to evaluate class performance in great detail, making the process of evaluating results more effective and efficient.

The existing approach requires instructors or other staff members to manually note and enter the grades issued by Mumbai University. This work is time-consuming and prone to human mistakes. Additionally, keeping track of the extra points and any grade adjustments is challenging. The input from the university alone is now insufficient for the teachers to assess the performance criteria of the class as a whole. As a result, a system is required for the upkeep and updating of records from the text file the institution has submitted.

This research paper explores the usage of the Zamzar[2] website for converting a PDF file into a CSV (Comma-Separated Values) file format. PDF (Portable Document Format) files are widely used for sharing and presenting documents due to their consistent formatting across different platforms. However, extracting data from a PDF file for further analysis can be challenging, as the format is primarily designed for display rather than data manipulation. On the other hand, CSV files offer a structured, tabular format that is easily readable by spreadsheet software and can be readily used for data analysis and manipulation.

II. METHODOLOGY


Receive result PDF from the Uniiversity
Convert PDF to CSV
Read CSV into Python
Clean data using reform.py Python script
Apply analysis method from analysis.py script
Generate report using the outputs of the various analysis methofs.
Fig 1 Process of Result Analysis and Report Generation

The entire process of result analysis and report generation can be divided into six broad steps. The first and second step involve the result document being received by the user and it being converted into a CSV document by them. Once the file is converted, the user now uploads the document into the Result Analyzer tool. The tool reads the document and stores the data into a Pandas DataFrame. At this step, the data is now ready for cleaning and analysis operations. The next step involves cleaning of the data using different methods and storing it in a new DataFrame. This cleaned data is then put through numerous analysis methods that extract useful insights from the data. Once the data is analyzed, a report is generated using the insights that summarizes the entire result document using parameters like average CGPA, average percentage, subject toppers, class topper, etc.

The development of the project involved the utilization of various materials and technologies to create an efficient and effective system for result analysis.

➢ *The following Tools/Technologies were used in the Development Process:*

- *Programming Languages: Python 3.7 or Higher.*
- *Data Analysis Libraries:*

- *The following Data Analytics Libraries were used:*

✓ *Pandas*
✓ *Numpy*
✓ *Scikit-Learn*
✓ *Seaborn*
✓ *Statistics*
✓ *Werkzeug*
✓ *Flask*

➢ *IDE: Visual Studio Code.*

- *Web Development Framework: Flask.*

Flask[3][4] is a lightweight and flexible web framework for building web applications in Python. With its minimalist design, Flask provides developers with a simple and intuitive approach to web development. It offers a routing system for handling different URLs and HTTP methods, a templating engine for dynamic HTML generation, and supports the integration of various extensions for added functionality. Flask promotes modular development through blueprints, allowing developers to organize their applications into reusable components. It also includes a built-in development server for easy testing and debugging. With its versatility and ease of use, Flask is a popular choice among developers looking to create web applications efficiently.

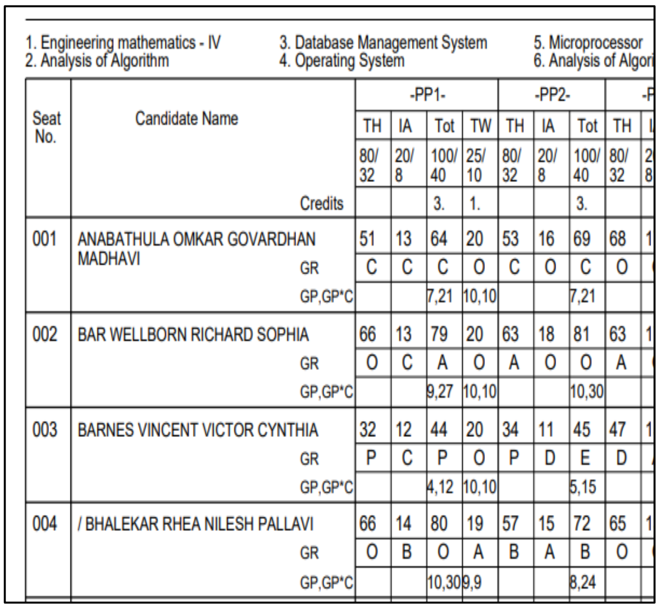
- *PDF to CSV Conversion Tool: Zamzar*

Zamzar[5] is an online file conversion service that allows users to convert files from one format to another. With Zamzar, users can convert a wide range of files, including documents, images, audio, and video files. It supports conversions between popular file formats such as DOCX to PDF, MP4 to AVI, JPEG to PNG, and more. The service is accessible through a web browser, eliminating the need for software installations. Zamzar offers a user-friendly interface, making it easy for individuals and businesses to convert files quickly and efficiently. It provides convenience and flexibility for managing file formats and is widely used for its reliable and comprehensive conversion capabilities.

- *Hardware:*

The hardware used for this project can be any computer or laptop with a minimum of 4GB RAM and 1 GHz processor.

III. DATA DESCRIPTION



1. Engineering mathematics - IV 2. Analysis of Algorithm 3. Database Management System 4. Operating System 5. Microprocessor 6. Analysis of Algori

Seat No.	Candidate Name		-PP1-				-PP2-			-P
			TH	IA	Tot	TW	TH	IA	Tot	TH
			80/ 32	20/ 8	100/ 40	25/ 10	80/ 32	20/ 8	100/ 40	80/ 32
		Credits			3.	1.			3.	
001	ANABATHULA OMKAR GOVARDHAN MADHAVI		51	13	64	20	53	16	69	68
		GR	C	C	C	O	C	O	C	O
		GP,GP*C			7,21	10,10			7,21	
002	BAR WELLBORN RICHARD SOPHIA		66	13	79	20	63	18	81	63
		GR	O	C	A	O	A	O	O	A
		GP,GP*C			9,27	10,10			10,30	
003	BARNES VINCENT VICTOR CYNTHIA		32	12	44	20	34	11	45	47
		GR	P	C	P	O	P	D	E	D
		GP,GP*C			4,12	10,10			5,15	
004	/ BHALEKAR RHEA NILESH PALLAVI		66	14	80	19	57	15	72	65
		GR	O	B	O	A	B	A	B	O
		GP,GP*C			10,30	9,9			8,24	

Fig 2 Example of the Result Document Received from the University of Mumbai

➢ *Data Description:*

- *Seat (Roll) number, Candidate name.*
- *PP#: each # column indicates the marks obtained by the student in each subject. (# indicates a number: 1,2,3,..)*
- *TH/IA/Tot/…: Bifurcations of the subject marks.*

Table 1 Data Description

Abbreviation	Marks obtained in
TH	Theory Exam
IA	Internal Assessment
TW	Term Work
OR	Oral/Viva Exam
PR	Project
Tot	Total

- *∑C, ∑CG: Summation of ‘Credits’ and ‘Product of Credits & Grades’.*
- *GPA: ∑C / ∑CG*
- *Remark: Pass(P) or Fail(F)*

IV. DATA PREPROCESSING

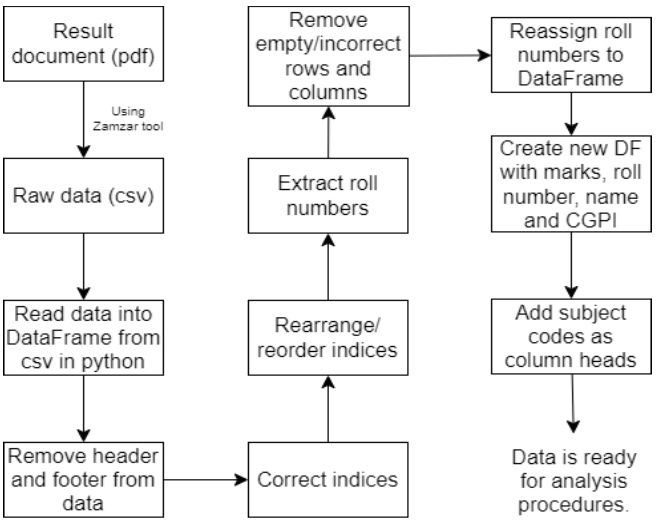
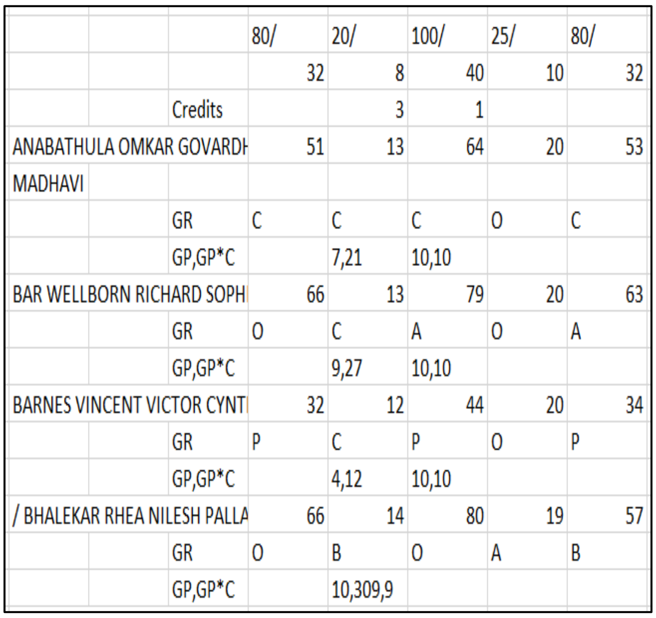

Result document (pdf)
Using Zamzar tool
Raw data (csv)
Read data into DataFrame from csv in python
Remove header and footer from data
Correct indices
Rearrange/ reorder indices
Extract roll numbers
Remove empty/incorrect rows and columns
Reassign roll numbers to DataFrame
Create new DF with marks, roll number, name and CGPI
Add subject codes as column heads
Data is ready for analysis procedures.

Fig 3 Flowchart for Data Cleaning

➢ *Data Preprocessing Steps[5]:*

- Convert .pdf to .csv: The result document generated from the University is in a .pdf format. PDF documents can directly be read in a python script[6] but the process is tedious and the results are not always as desired. Some libraries like PyPDF[7], PyPDF2[8] do exist that can be used to read data directly from a PDF file but for our particular application, the results were not as expected. A more reliable way of reading the data from the PDF document is to first convert it to CSV (Comma Separated Values) and then reading it into a Python script. There exist various tools that can be used to convert .pdf to .csv, one of which is Zamzar. There are various other tools that do the same thing but it was observed that Zamzar gives the best results in terms of least loss of data and least change in the formatting.



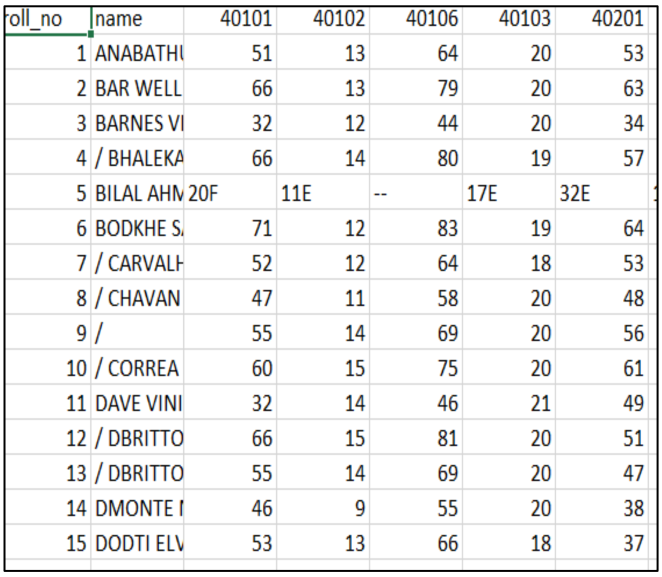
			80/	20/	100/	25/	80/
			32	8	40	10	32
		Credits		3	1		
ANABATHULA OMKAR GOVARDH			51	13	64	20	53
MADHAVI							
		GR	C	C	C	O	C
		GP,GP*C		7,21	10,10		
BAR WELLBORN RICHARD SOPH			66	13	79	20	63
		GR	O	C	A	O	A
		GP,GP*C		9,27	10,10		
BARNES VINCENT VICTOR CYNT			32	12	44	20	34
		GR	P	C	P	O	P
		GP,GP*C		4,12	10,10		
/ BHALEKAR RHEA NILESH PALLA			66	14	80	19	57
		GR	O	B	O	A	B
		GP,GP*C		10,309,9			

Fig 4 Format of Data after Conversion
No Preprocessing done.

- Read .csv into python script: Use the pandas. Read csv()[9] function to read the csv data.
- Remove headers and footers from the data
- Correct indices
- Rearrange/reorder columns
- Extract roll numbers
- Remove empty/incorrect rows and columns
- Reassign roll numbers to the dataframe
- Create new dataframe with roll number, name, subject marks and CGPI
- Add subject codes as column heads.

➢ *Explanation of Data Preparation functions:*

- `Remove_rows_columns(result, branch, semester)`: Removes the initial rows and specific columns from the DataFrame `result` based on the given `branch` and `semester` values, effectively removing unnecessary information.
- `Correct_rows_columns(result)`: Sorts the columns of the DataFrame `result` in ascending order, resets the index, and reassigns new indices to the rows to ensure a clean and consistent data structure.
- `Find_junk_indices(result)`: Identifies the indices of rows in the DataFrame `result` that contain junk data based on a specific condition, helping locate irrelevant or inconsistent data points.
- `Cleared_result(result_cleared, result)`: Extracts and concatenates the relevant data from the DataFrame `result` into the `result_cleared` DataFrame by excluding the junk rows, resulting in a cleaned dataset.
- `Extract_rollnumbers(result)`: Extracts roll numbers from the first column of the DataFrame `result`, removing NaN values, longer strings, and the string 'No.', and converting the remaining values to integers.
- `Remove_empty_rows(result_cleared)`: Removes rows from the DataFrame `result_cleared` where certain columns have missing values (NaN), further refining the dataset by eliminating incomplete data.
- `Reassign_rollnos(roll_numbers, result_cleared)`: Reassign the extracted roll numbers obtained from `extract_rollnumbers` to the first column of the DataFrame `result_cleared`, ensuring accurate and consistent roll number representation.
- `Add_header(result_cleared, sem)`: Adds a header to the DataFrame `result_cleared` based on the given semester `sem`, assigning appropriate column names to the DataFrame for better data interpretation.



oll_no	name	40101	40102	40106	40103	40201
1	ANABATHI	51	13	64	20	53
2	BAR WELL	66	13	79	20	63
3	BARNES VI	32	12	44	20	34
4	/ BHALEKA	66	14	80	19	57
5	BILAL AHN	20F	11E	--	17E	32E
6	BODKHE S.	71	12	83	19	64
7	/ CARVALH	52	12	64	18	53
8	/ CHAVAN	47	11	58	20	48
9	/	55	14	69	20	56
10	/ CORREA	60	15	75	20	61
11	DAVE VINI	32	14	46	21	49
12	/ DBRITTO	66	15	81	20	51
13	/ DBRITTO	55	14	69	20	47
14	DMONTE I	46	9	55	20	38
15	DODTI ELV	53	13	66	18	37

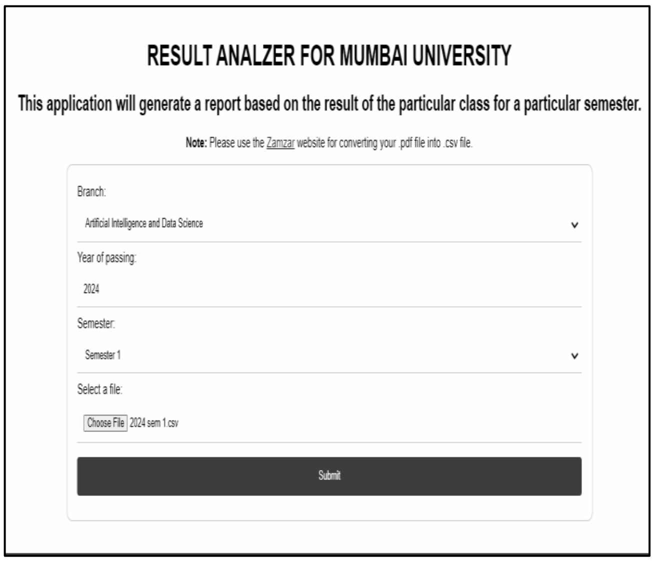
Fig 5 Format of the Data after Preprocessing Steps

V. DATA ANALYSIS

➢ *After the data has gone through all of the aforementioned functions, it is finally in a form that can be read and used for analysis purposes. The analysis process that we have employed in this project include plotting graphs, calculating aggregate percentage, CGPI, etc. The functions that have been created are explained below.*

- 'Count_students(percentages)`: Counts the number of students falling within specific percentage ranges, such as '>60', '(50, 60]', '(40, 50]', '(12, 40]', and 'Fail/KT', based on their percentages. It returns a dictionary with the counts for each range.
- `Get_avg_cgpa(cleared_result)`: Calculates the average CGPA (Cumulative Grade Point Average) from the 'cgpa' column of the `cleared_result` DataFrame. It also returns the list of CGPA values for further analysis.
- `Get_avg_percentage(CGPA)`: Calculates the average percentage based on the CGPA values provided. The function converts CGPA to percentage using specific conversion formulas and returns the average percentage along with the list of percentage values.
- `Find_toppers(cleared_result)`: Finds the top three students based on their CGPA from the `cleared_result` DataFrame. It returns a list of dictionaries where each dictionary contains the roll number and CGPA of a topper.
- `Find_sub_max(cleared_result, semester)`: Retrieves the maximum marks obtained in each subject for a given semester, as specified in the 'subjects.xlsx' file. It returns a dictionary where the keys are subject names and the values are the corresponding maximum marks.
- `Find_kts(cleared_result)`: Counts the number of students who have KT (Keep Terms) in any subject, identified by missing CGPA values in the `cleared_result` DataFrame. It returns the count of KT students.
- `Find_sub_mark_ranges(cleared_result, semester)`: Determines the count of students who have scored above 60 marks in each subject for a given semester, as specified in the 'subjects.xlsx' file. It returns a dictionary where the keys are subject names, and the values are the counts of students who scored above 60 marks.

VI. RESULTS AND DISCUSSION



RESULT ANALZER FOR MUMBAI UNIVERSITY

This application will generate a report based on the result of the particular class for a particular semester.

Note: Please use the Zamzar website for converting your .pdf file into .csv file.

Branch: Artificial Intelligence and Data Science

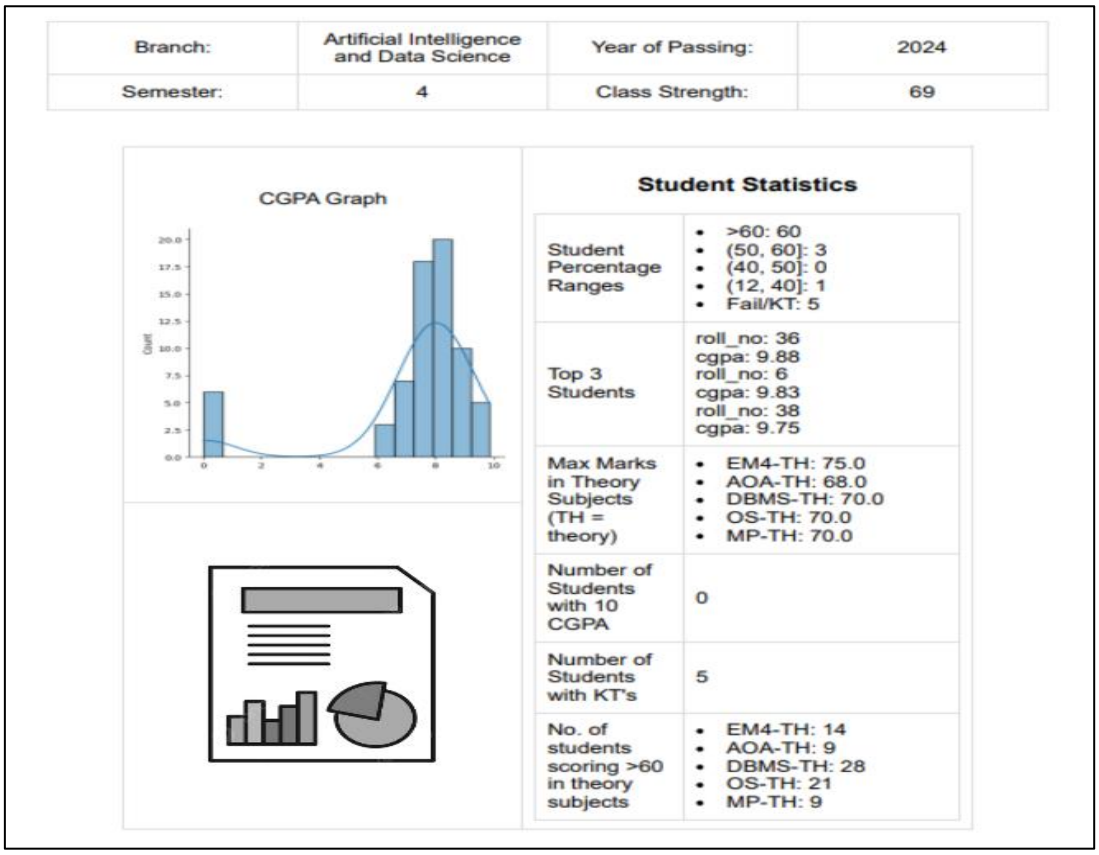
Year of passing: 2024

Semester: Semester 1

Select a file: Choose File 2024 sem 1.csv

Fig 6 Input form of the Toll. The users Need to Upload their Result Document in .csv Format Along with other Details

The product of the entire cleaning and analysis process of the result data is a report[10] that is generated from the tool. This report is a concise review of the data projected in the result of the students. This report is a simplified visualization of this information. This report can prove to be of great help for teachers to understand the trends that are seen in marks of a particular class over the eight semesters or engineering or the marks obtained by students in a particular subject in case students of continuous batches are struggling with the subject.



Branch:	Artificial Intelligence and Data Science	Year of Passing:	2024
Semester:	4	Class Strength:	69

Student Statistics

Student Percentage Ranges	• >60: 60 • (50, 60]: 3 • (40, 50]: 0 • (12, 40]: 1 • Fail/KT: 5
Top 3 Students	roll_no: 36 cgpa: 9.88 roll_no: 6 cgpa: 9.83 roll_no: 38 cgpa: 9.75
Max Marks in Theory Subjects (TH = theory)	• EM4-TH: 75.0 • AOA-TH: 68.0 • DBMS-TH: 70.0 • OS-TH: 70.0 • MP-TH: 70.0
Number of Students with 10 CGPA	0
Number of Students with KT's	5
No. of students scoring >60 in theory subjects	• EM4-TH: 14 • AOA-TH: 9 • DBMS-TH: 28 • OS-TH: 21 • MP-TH: 9

Fig 7 Output given by the Tool this is the 'Print Preview' View of the Report that is Generated

VII. CONCLUSION

In conclusion, we have successfully developed a system to process and analyze university result documents from the University of Mumbai using Python. The system has been designed to convert the result documents into a standard format that can be easily read and analyzed using data analysis techniques and libraries in Python. The system has been tested with a sample set of result documents and has been found to be effective in extracting meaningful insights from the data.

The website allows users to upload their result documents in CSV format, and the system automatically converts them into a format that can be analyzed by Python. The website displays the analysis results in an easy-to-understand format, including summary statistics, trends and patterns, and visualizations.

REFERENCES

[1]. Miss. S. I. Tamboli, Miss. P. V. Tate, Miss. S. P. Kagwade, Miss. H. T. Magdum, and Prof. S. S. Chavan, "Result Analysis," International Journal for Research in Applied Science and Engineering Technology, vol. 10, no. 6. International Journal for Research in Applied Science and Engineering Technology (IJRASET), pp. 821–826, Jun. 30, 2022. doi: 10.22214/ijraset.2022.43760.
[2]. Zamzar documentation https://developers.zamzar.com/docs
[3]. Flask tutorial, https://www.tutorialspoint.com /flask/index.html , Jan 2022
[4]. Adding HTML CSS in Flask, https://thinkinfi.com/flask-adding-html-and-css/
[5]. Huong Ngo, 'How to clean data in python 'https://towardsdatascience.com/how-to-clean-your-data-in-python-8f178638b98d ', July 2022
[6]. How to Extract Data from PDF Files with Python https://www.freecodecamp.org/news/extract-data-from-pdf-files-with-python/
[7]. PyPDF https://pypi.org/project/pypdf/
[8]. PyPDF2 documentation https://pypdf2.readthedocs. io/en/3.0.0/
[9]. Pandas.read_csv https://pandas.pydata.org/docs/ reference/api/pandas.read_csv.html
[10]. Gjorgjevikj, Dejan & Madjarov, Gjorgji & Chorbev, Ivan & Angelovski, Martin & Georgiev, Marjan & Dikovski, Bojan. (2011). ASGRT – Automated Report Generation System. 369-376. 10.1007/978-3-642-19325-5_38.
[11]. Baradwaj, Brijesh & Pal, Saurabh. (2011). Mining Educational Data to Analyze Students' Performance. International Journal of Advanced Computer Science and Applications. 2. 63-69. 10.14569/IJACSA. 2011. 020609.
[12]. Lei, Yujiao & Deng, Jiqiu & Lin, Jian & Dick, Jeffrey & Lessani, M.Naser & Liu, Chaoyue. (2020). Research of Automatic Generation for Engineering Geological Survey Reports Based on a Four-Dimensional Dynamic Template. ISPRS International Journal of Geo-Information. 9. 496. 10.3390/ijgi9090496.