

Result Extraction and Analysis

Designed and developed by: Mehul Choksi

Mentored by: Prof. Potdar Sir

Problem statement

- Given a Portable Document Format(PDF) of the college result, design and develop an application that is able to extract the marks, and store it in a ready-to-analyse format
- Develop a module that is further able to perform the analysis on the individual scores.

This includes:

- Calculating the 5 stat figure (min, lower quartile, median, upper quartile, max)
- Calculate standard deviation in each subject
- Present this data in a tabular format
- Additional histograms for graphical visualization of performance of students

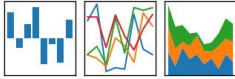
Technology stack



- Web application: Python flask

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



- Data analysis and visualization: Python Pandas, Seaborn, matplotlib, weasyprint



- Data extractor: (Pdf to CSV module) : Java 8, Maven, Apache PdfBox

Method adopted : Data Extraction

- Tokenize the lines using space as delimiter
- Based on the department, year and semester we find the rule file and load it in a map
- These rules are then used to parse each and every result in the pdf.
These are then stored temporarily in member as list of objects
- After the entire pdf is parsed, we convert the list of objects into a csv

Method adopted: Analysis and visualization

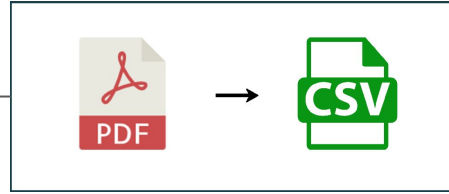
- CSV data is loaded in Pandas dataframe
- Subsequently, the 5 value summary, standard deviation are obtained
- Bar plots are done using matplotlib
- The outputs are saved as html. After that html files are converted into pdf format, using the weasyprint library
- Finally, the results analysis are merged and presented in pdf form

Challenges

- Handling anomalous formats: students with backlog often have varying schema
- Handling line overflows in pdf
- Handle NaN values, for eg, absent students have score: (AB/50)
- Fill up the gap of first year result of diploma students by substituting the mean

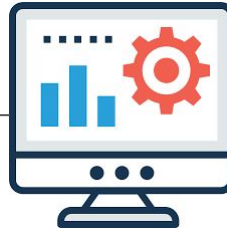


Web application



Pdf to csv converter

Api: localhost:5000/pdftocsv



Data analysis and visualization

Api: localhost:5000/analytics

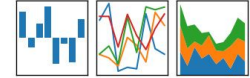


maven

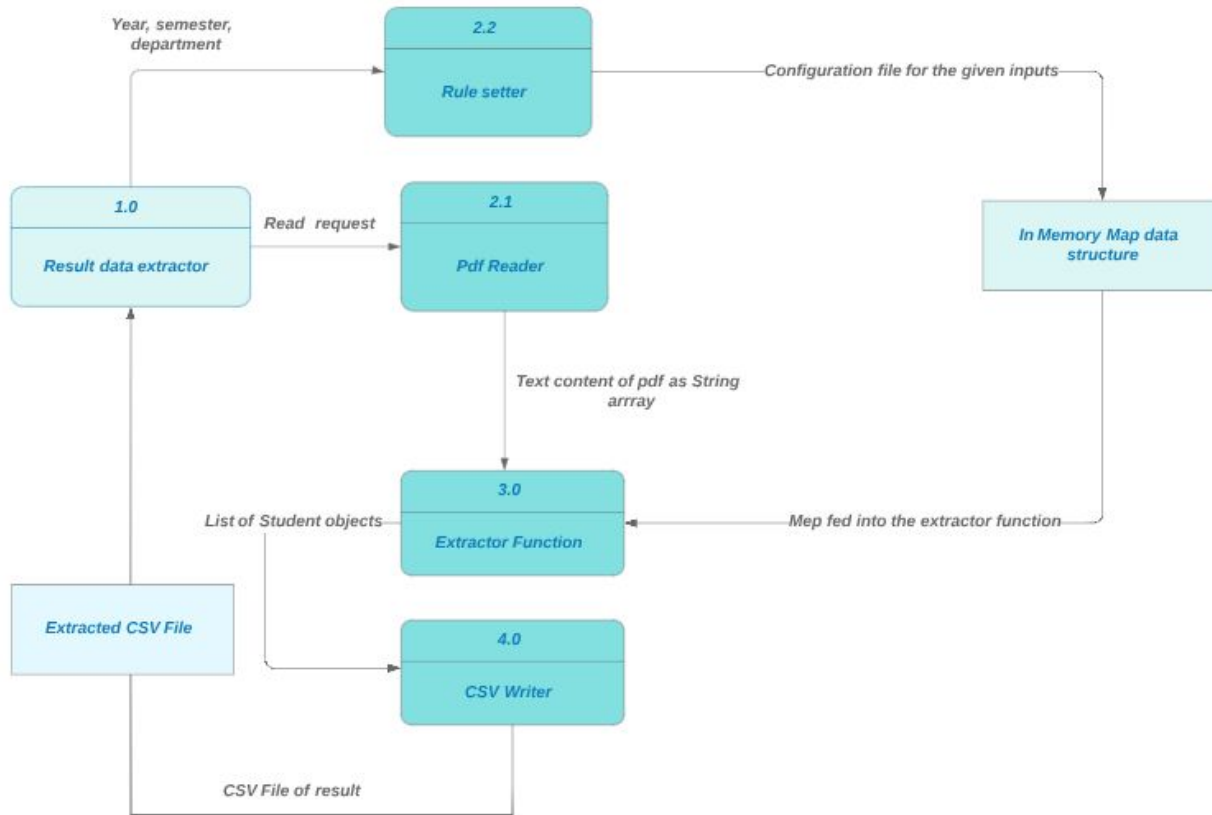


pandas

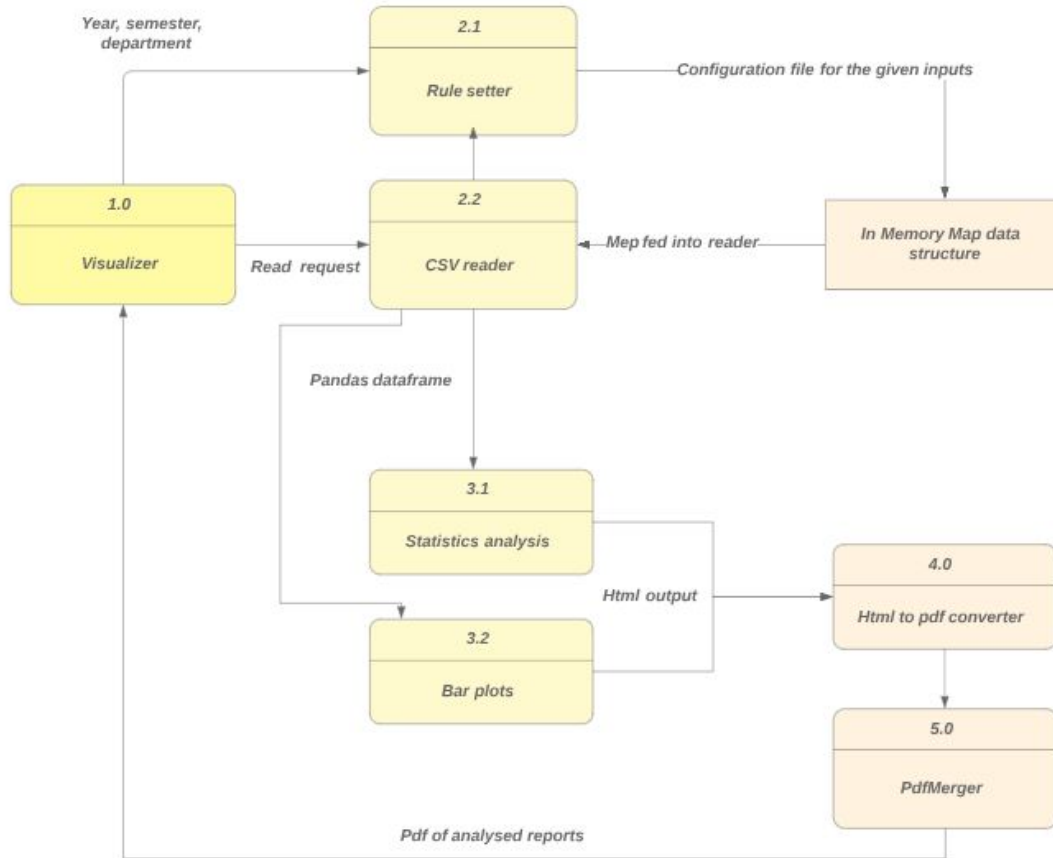
$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



Seaborn **matplotlib**



Data flow
diagram of pdf
extractor



Data flow
diagram for
visualizer

Results: Extracted csv

be_comp_may.pdf.csv - LibreOffice Calc

LibreOffice Calc interface showing a spreadsheet with data extracted from a CSV file. The spreadsheet has columns labeled A through T and rows numbered 1 to 30. The data includes Roll No, HPC, AIR, DA, Elec_1, Elec_2, LP1_TW, LP1_PR, LP2_TW, LP2_OR, BE_PROJ_STAGE1, ML, ICS, Elec_3, Elec_4, LP3_TW, LP3_PR, LP4_TW, LP4_OR, and BE_PROJ_STAGE2_TW.

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
|----|------------|-----|-----|----|--------|--------|--------|--------|--------|--------|----------------|----|-----|--------|--------|--------|--------|--------|--------|-------------------|
| | Roll No | HPC | AIR | DA | Elec_1 | Elec_2 | LP1_TW | LP1_PR | LP2_TW | LP2_OR | BE_PROJ_STAGE1 | ML | ICS | Elec_3 | Elec_4 | LP3_TW | LP3_PR | LP4_TW | LP4_OR | BE_PROJ_STAGE2_TW |
| 1 | B150054201 | 67 | 59 | 61 | 67 | 67 | 40 | 35 | 38 | 30 | 36 | 55 | 53 | 59 | 45 | 38 | 38 | 43 | 41 | 74 |
| 2 | B150054202 | 75 | 63 | 73 | 76 | 71 | 44 | 41 | 40 | 40 | 45 | 64 | 76 | 72 | 64 | 46 | 41 | 45 | 44 | 97 |
| 3 | B150054203 | 65 | 51 | 63 | 74 | 50 | 40 | 36 | 37 | 39 | 40 | 56 | 76 | 77 | 63 | 40 | 40 | 45 | 43 | 82 |
| 4 | B150054204 | 68 | 55 | 58 | 44 | 54 | 47 | 44 | 42 | 39 | 46 | 57 | 70 | 64 | 55 | 42 | 40 | 45 | 44 | 98 |
| 5 | B150054205 | 73 | 65 | 79 | 72 | 81 | 47 | 44 | 38 | 38 | 43 | 58 | 79 | 64 | 70 | 44 | 40 | 45 | 43 | 95 |
| 6 | B150054206 | 65 | 70 | 68 | 70 | 75 | 47 | 45 | 43 | 38 | 45 | 67 | 80 | 81 | 58 | 43 | 42 | 42 | 40 | 96 |
| 7 | B150054207 | 64 | 64 | 64 | 74 | 68 | 43 | 40 | 44 | 40 | 41 | 57 | 75 | 76 | 70 | 38 | 38 | 42 | 40 | 91 |
| 8 | B150054208 | 60 | 51 | 66 | 74 | 69 | 40 | 29 | 40 | 39 | 43 | 54 | 71 | 71 | 68 | 38 | 41 | 40 | 38 | 95 |
| 9 | B150054209 | 74 | 69 | 72 | 75 | 78 | 46 | 43 | 41 | 39 | 42 | 71 | 86 | 85 | 77 | 42 | 42 | 44 | 42 | 96 |
| 10 | B150054210 | 63 | 56 | 66 | 70 | 68 | 40 | 29 | 42 | 37 | 42 | 68 | 79 | 84 | 63 | 42 | 42 | 40 | 38 | 86 |
| 11 | B150054211 | 65 | 62 | 59 | 74 | 65 | 40 | 29 | 38 | 39 | 46 | 57 | 74 | 71 | 65 | 39 | 35 | 43 | 41 | 96 |
| 12 | B150054212 | 61 | 50 | 49 | 59 | 55 | 40 | 22 | 36 | 39 | 39 | 54 | 69 | 68 | 49 | 41 | 42 | 43 | 41 | 90 |
| 13 | B150054213 | 69 | 70 | 70 | 69 | 75 | 45 | 42 | 44 | 42 | 44 | 70 | 82 | 65 | 77 | 43 | 46 | 44 | 42 | 91 |
| 14 | B150054214 | 67 | 51 | 60 | 59 | 56 | 40 | 29 | 41 | 32 | 43 | 58 | 71 | 72 | 64 | 43 | 42 | 44 | 42 | 96 |
| 15 | B150054215 | 63 | 51 | 63 | 76 | 50 | 45 | 42 | 37 | 35 | 44 | 60 | 72 | 77 | 63 | 40 | 41 | 45 | 44 | 93 |
| 16 | B150054216 | 56 | 55 | 65 | 75 | 60 | 46 | 43 | 41 | 43 | 43 | 63 | 76 | 73 | 68 | 38 | 40 | 45 | 43 | 91 |
| 17 | B150054217 | 69 | 68 | 64 | 71 | 57 | 47 | 44 | 39 | 38 | 41 | 66 | 76 | 76 | 75 | 43 | 42 | 37 | 35 | 91 |
| 18 | B150054218 | 69 | 70 | 56 | 66 | 67 | 40 | 38 | 43 | 36 | 40 | 50 | 78 | 68 | 50 | 40 | 35 | 39 | 37 | 79 |
| 19 | B150054219 | 56 | 51 | 51 | 60 | 49 | 40 | 29 | 40 | 37 | 38 | 45 | 62 | 59 | 53 | 41 | 38 | 37 | 35 | 90 |
| 20 | B150054220 | 63 | 62 | 61 | 75 | 67 | 45 | 42 | 40 | 40 | 45 | 63 | 75 | 57 | 65 | 38 | 40 | 41 | 39 | 96 |
| 21 | B150054221 | 71 | 70 | 66 | 72 | 48 | 43 | 40 | 42 | 37 | 40 | 62 | 77 | 66 | 66 | 43 | 40 | 41 | 39 | 90 |
| 22 | B150054222 | 46 | 45 | 42 | 40 | 48 | 47 | 44 | 40 | 36 | 35 | 46 | 49 | 49 | 45 | 35 | 39 | 39 | 37 | 73 |
| 23 | B150054223 | 74 | 61 | 66 | 73 | 68 | 41 | 38 | 41 | 38 | 44 | 61 | 69 | 64 | 51 | 39 | 39 | 45 | 44 | 96 |
| 24 | B150054224 | 73 | 59 | 58 | 70 | 79 | 44 | 41 | 42 | 41 | 42 | 56 | 72 | 64 | 60 | 42 | 43 | 39 | 37 | 96 |
| 25 | B150054225 | 55 | 46 | 53 | 70 | 54 | 41 | 38 | 42 | 39 | 42 | 50 | 61 | 51 | 51 | 43 | 35 | 44 | 42 | 96 |
| 26 | B150054226 | 72 | 67 | 68 | 62 | 76 | 45 | 42 | 40 | 35 | 41 | 54 | 71 | 54 | 47 | 40 | 41 | 44 | 42 | 90 |
| 27 | B150054227 | 57 | 54 | 58 | 60 | 54 | 47 | 45 | 39 | 40 | 45 | 55 | 66 | 53 | 48 | 41 | 42 | 44 | 42 | 96 |
| 28 | B150054228 | 73 | 71 | 69 | 71 | 66 | 43 | 40 | 40 | 39 | 43 | 60 | 75 | 59 | 66 | 44 | 43 | 43 | 41 | 95 |
| 29 | B150054229 | 71 | 73 | 71 | 76 | 78 | 43 | 40 | 40 | 38 | 41 | 56 | 75 | 67 | 63 | 41 | 42 | 43 | 41 | 92 |

Sheet 1 of 1

Find

Find All Search Formatted Display String Match Case

Sum=0

100%

Results : Generated analysis report

| | HPC | AIR | DA | Elec_1 | Elec_2 |
|-------|------------|------------|------------|------------|------------|
| count | 311.000000 | 311.000000 | 311.000000 | 311.000000 | 311.000000 |
| mean | 65.491961 | 62.636656 | 65.138264 | 63.935691 | 65.295820 |
| std | 9.410997 | 9.214635 | 9.248026 | 11.461396 | 10.098353 |
| min | 35.000000 | 36.000000 | 23.000000 | 0.000000 | 26.000000 |
| 25% | 59.000000 | 57.000000 | 59.500000 | 58.000000 | 60.000000 |
| 50% | 67.000000 | 62.000000 | 67.000000 | 66.000000 | 67.000000 |
| 75% | 72.500000 | 70.000000 | 72.000000 | 71.000000 | 73.000000 |
| max | 86.000000 | 84.000000 | 85.000000 | 87.000000 | 84.000000 |

