%Part 2 technical basics

2.1 Regression basics:

Regression/Linear Regression: In the context of this work, linear regression is a technique that aims to find a line that best fits a given set of data points. It does so by minimizing the distance between the line and the data points. The formula of the line is y = mx+b, which is a straight line

Polynomial Regression: Refers to the line that has N number of polynoms in its formula (y = mx^n + mx^(n-1) + … + mx + b, where m is a slope and b is an intercept. Both, m and b parameters are approximated by regression. N is heuristical). Polynoms enable the line to be non-linear and have some curvature

%Part 3.1

3.1 Description of the current status:

The application is currently in an operational state, allowing users to input their exam plans for evaluation. The system utilizes algorithms and rule-based systems to process the exam plans based on the predefined criteria. It assigns weights to each criterion to reflect their importance in the overall assessment.

The rules are following:

* Big exams have to be early
* Students mustn’t have two exams in the same day
* Students should have one day gap between exams
* Rooms have to be not too big or too small for the exam
* If two rooms are assigned for the exam, they have to be as close as possible
* Professor is not available on some date
* Professor wants to come on minimal amount of days

The current version of the application provides a score indicating the overall quality of the exam plan. Additionally, it generates an HTML report file that includes visualizations, conflict dataframes, and scores for each individual criteria. This report serves as a valuable tool to help professors spot potential problems and gain insights into the strengths and weaknesses of the exam plan.

%Part 3.2

3.2 Functional requirements

The application should assess the exam plans based on predefined criteria, evaluating each criterion individually. The perfect exam plan would satisfy both, professors and students. Thus, we should aim to provide a comprehensive solution that would take into consideration both perspectives.

The application should calculate an overall score for each exam plan, indicating its quality. The scoring mechanism will be based on weighted averages, taking into account the relative importance of each criterion.

The application should provide detailed explanations for the assigned score, clearly stating the reasoning and factors that contributed to it. This feature helps professors understand the strengths and weaknesses of the exam plan and provides transparency in the evaluation process.

The application should generate an HTML report file, presenting visualizations, conflict dataframes, and scores for each individual criterion. The report should be easily accessible and designed in a clear and intuitive manner, allowing professors to review the assessment results effectively.

The application should have conflict detection capabilities, identifying any conflicts within the exam plan. It should highlight inconsistencies or contradictions between criteria in the report, enabling professors to address and resolve these conflicts during the exam planning process.

By fulfilling these functional requirements, the solution aims to comprehensively evaluate exam plans, calculate scores, provide explainability, generate informative HTML reports, and assist professors in identifying and resolving conflicts to enhance the quality of their assessments.

%Part 3.3

3.3 Non-functional requirements

The application should be reliable, ensuring accurate and consistent evaluation results. It should be able to handle errors and exceptions gracefully, with minimal impact on the overall functionality. On top of that, the application should be designed and developed with maintainability in mind, allowing for easy updates, bug fixes, and future enhancements

%Part 5.1

5.1 Dependencies on other software system

The application has dependencies on specific software systems and packages to ensure its smooth functioning. In order for the application to run successfully, Python 3.10.9 must be installed, along with the following required packages:

* Matplotlib 3.7.0: for data visualization
* Pandas 1.5.3: for data manipulation
* Numpy 1.23.5: to operate with matrices
* Json (comes prebuilt with python): to read json file
* Csv (comes prebuilt with python): to read csv file
* Byte64 (comes prebuilt with python): required for html creation
* Io (comes prebuilt with python): required for html creation

%5.2 part

5.2 Software Availability:

The application is available in a BitBucket repository, accessible via the following link: https://bitbucket.student.fiw.fhws.de:8443/projects/PRGPROJSS23/repos/programmierprojekt-ss-23---gruppe-99---exam-quality-control/browse

The BitBucket repository contains the source code, documentation, and any other relevant resources for the software. By accessing the repository, users can review the code, download the necessary files, or contribute to the development process.

%Part 5.3

5.3 Installing the Software:

To install the software, follow these steps:

Access the BitBucket repository using the provided link.

Clone the repository to your local machine by either downloading the repository as a ZIP file or using a Git client to clone the repository.

Ensure that Python 3.10.9 is installed on your system. If not, download and install Python 3.10.9 from the official Python website (https://www.python.org).

Open a command-line interface or terminal and navigate to the location where the repository was cloned or extracted.

Install the required dependencies (Matplotlib, NumPy, Pandas) by executing the following command:

pip install -r requirements.txt

This command will install all the required packages specified in the requirements.txt file.

Once the dependencies are installed, the software is ready to run.