%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.

%Part 5.4

5.4 Opportunities for later adaptation and further development

There are several potential areas for future development and improvement of the application. Let's discuss the suggested enhancements:

One could improve scoring system by including information about which examiners failed the exam and have to retake it. It can be a valuable addition to the scoring system. By considering individual examiner failures, the system can provide a fairer evaluation of exam plans, penalizing them less for consecutive exams for those specific students. This enhancement would require modifying the evaluation criteria and calculation algorithms to incorporate examiner failure data. By the time we were developing the project, we did not have data for that

To enhance the user experience and make the application more user-friendly, improvements can be made to the installation process. This could involve creating an installer or package that simplifies the installation steps, automatically handles dependencies, and provides clear instructions for setting up and running the application. Additionally, providing a user-friendly installation guide or script can help streamline the installation process

Also developing a Graphical User Interface (GUI) for the application can significantly enhance the user experience. A GUI would provide a visual interface with intuitive controls and interactive elements, making it easier for professors to input exam plans, view evaluation results, and interact with the system. The GUI can incorporate features such as drag-and-drop functionality, visual representations of data, and real-time updates, offering a more engaging and efficient user experience

%Part 6.1

6.1 Evaluation of the project results

The team's assessment of the project is that while it is not considered fully complete, it showcases promising potential for further enhancements, as mentioned earlier. However, the inner logic of the application is well-designed and demonstrates a high level of quality. The project successfully incorporates multiple evaluation criteria and effectively models them in an efficient and robust manner. This accomplishment represents a significant step forward in providing a viable solution to the problem of exam scoring.

The successful implementation of the evaluation criteria demonstrates the team's ability to handle complex requirements and deliver a functional solution. The application's capacity to assess exam plans based on predefined criteria contributes to improving the quality and fairness of the evaluation process. The team's effort in designing and implementing the inner logic of the application is commendable.

Moving forward, the team acknowledges the areas for improvement highlighted earlier, such as enhancing the scoring system, improving installability, and developing a graphical user interface. By addressing these aspects, the application can be further refined to offer an even better user experience and provide more accurate evaluation results.

Overall, the team's evaluation of the project is positive, recognizing the solid foundation and potential for future improvements. The successful implementation of the inner logic and the incorporation of multiple evaluation criteria demonstrate the team's competence and commitment to delivering a valuable solution to the problem of exam scoring.

% Part 6.2

Project management method and tools used

The project team employed the Agile and Scrum methodologies for project management. These methodologies provided a suitable framework for managing the workload while maintaining flexibility in terms of work schedules. Here are some details regarding the project management method and tools used:

For our project, we used Agile and Scrum methodologies. Agile principles emphasize flexibility, collaboration, and iterative development. Scrum, a specific Agile framework, was adopted to organize the work into time-boxed iterations (sprints) and facilitate regular communication and feedback

The team members worked individually or in pairs, depending on the task requirements. This approach allowed for a balanced distribution of workload and encouraged collaboration and knowledge sharing among team members

The team held regular meetings to ensure effective communication and progress tracking. This included weekly meetings with the supervisor to receive guidance, feedback, and align project goals. Additionally, weekly team meetings were conducted to discuss implementation details, and address any challenges or concerns

For the collaboration, we were having a WhatsApp group chat. Maybe it was unprofessional from us, but that made it very easy to collaborate and discuss our project. By doing it in a regular messenger, we established efficient and quick way to communicate. For the version control system, we used BitBucket, which was offered to us in the very beginning of the project

%Part 6.3  
Description of the project process

The project process began with a kick-off session, followed by the implementation of "BigExamsEarly" over the course of approximately three sprints. Initially, the team encountered challenges in modeling the rules and started with a naive approach, but ended up with an advanced way of handling the rule. As more rules were identified, the team made a decision not to spend excessive time on implementing a perfect solution but focused on implementing each rule incrementally.

During the subsequent sprints, the team worked on one rule at a time. However, it was later realized that task splitting among team members proved to be difficult, prompting a shift to working on two rules simultaneously. The duration of sprints varied, typically ranging from one to two weeks.

In the middle of the project, the team recognized the need to rework the code structure and organization due to its initial lack of organization. This restructuring phase took around 2-3 weeks to complete. It is worth noting that during an international week, all team members participated in external activities, resulting in no project work during that period. The team resumed working on the project the following week.

Towards the end of the project, the remaining rules, specifically those related to professors, were implemented. Extensive testing was performed, and final adjustments were made to the code. Due to time constraints, the decision was made to conclude the project at this stage.

%Part 6.4

Evaluation of one's own way of working and cooperation in the team

**Illia Rohalskyi:**

As a person that was having good knowledge of Python, I was having lots of questions in my team. At some point, it felt like I was taking a leadership role, since I was helping out each of the team members, answering their questions and pushing them to fight procrastination and finish the task before deadline. I found it to be difficult managing people, as everyone has their own way of thinking and you have to find individual way to interact with people. It came down to even explaining the same concept in different ways. It felt like speaking separate languages. But I think I did a good job, as my team was grasping what I was explaining to them. There were technical difficulties due to different backgrounds. Me and Kemal knew how to do linear regression and we proposed lots of ways of doing evaluation criterias, but we were not as proficient in object oriented programming and organising the project as Kaan and Muberra were. In the end, I believe, because we were very diverse, it turned out to be a very juiceful blend of our knowledge.

**Kaan Özer:**

There were a few challanges. First of all, gathering every team members for a meeting was very difficult because every member had different exams, responsibilities, and needed to go other cities. However, we solved this issue by discussing on the days that suits everyone, met at the university to prevent distance problems and we sometimes did online meetings. Secondly, there were some technical problems, I was not used to programming with python and hadn't enough knowledge for regressions, pilots and machine learning topics. Thirdly, sometimes we had communication problems and couldn't agree on some topics. We arranged a meeting with Peter Braun to ask our questions and make everything clear. It was very difficult to writing codes with python. Therefore, I started with the fundamental rules and the time when the things get harder my teammates shared their experience with me. Also, I had another skills that comes from my past experiences. asides the implementation of some rules, I also helped for the output processes to make our outputs visible on the website so that we visualize our results, dicuss on them and have a general report with the total score. In addition, I had experience with latex. Therefore, I helped in the documentation by adding my team mates' findings by embeding latex. I attend the all meetings as my all team mates do, and often I represent my opinion about the tasks and implementation processes.

**Müberra Şeyma Uslu:**

At the beginning, it was a bit challenging to decide on which programming language to use, as we were familiar with different languages. Additionally, understanding each other's perspectives was difficult due to our individual coding styles. However, we overcame these communication problems by frequently coming together and having discussions. Through these conversations, we identified each team member's strengths and weaknesses, allowing us to allocate tasks accordingly and proceed with the project in that manner.

**Kemal Öztürk**

Communication Problems:

There were a few instances of communication breakdowns within the team. Sometimes ideas were not communicated clearly, leading to misunderstandings and confusion. This resulted in some inefficiencies and rework. However, we quickly realized the importance of proper communication and made efforts to improve our clarity and actively listen to each other.

Distributing Tasks:

Initially, all team members attempted to code a single Python program, which proved to be highly ineffective. As a result, we decided to split into two groups, each responsible for a specific aspect of the project. This division significantly improved our efficiency and allowed for smoother progress in both designing and coding.

Design:

The process of programming each rule was relatively easy and straightforward. However, designing the software architecture posed significant challenges. We struggled with finding the most efficient and effective way to structure the software, which caused delays and confusion. But in the end, we pushed ourselves to design a proper software design.

Procrastination:

Procrastination became a minor issue during the project. We occasionally found ourselves delaying tasks or not fully utilizing our time. This led to some unnecessary pressure and rushed work towards the end. However, we were able to recognize this problem early on and actively worked on improving our time management and productivity.

%Part 6.5

Lessons learned for future projects

**Illia Rohalskyi:**

I was not very familiar with git, so this really will help me in my future. I also was not aware of structuing the project, and, I guess, I would do it differently. I would think of structure in the very first meeting, instead of reworking everything in the middle of project development.

In contrast to my fellows, I believe, Python was a wise choice for this project. Python has a great ecosystem to work with data and is also very simple. In my perspective, if we would use other language (such as Java, Rust or any C-based language), our team would have harder times implementing the project, as those languages are more difficult to learn. We also did not need those extra miliseconds of faster running time. So overall, in my opinion, python was perfect for this task.

I learned how to communicate with people, share ideas and manage team with different backgrounds. I guess, this is the most valuable lesson from this project.

**Kaan Özer:**

I learned that there is no shame to restart everthing from the beginning several times in a project because sometimes we might change our minds and need to change our classes and implementations. Starting a project with a new language that you don't know at all is not a big deal after knowing some programming basics and having good team mates. Additionally, I learned many things regarding machine learning and python environment. In the next projects, I 'll never say that let's complete it then we arrange it because it tooks a lot of time. Everthing should be organized from scratch to end. otherwise there will be a huge mess and hussle at the end.

**Müberra Şeyma Uslu:**

I had never been involved in a team project where we collaborated on coding for such a long period before, so it was a significant plus and learning experience for me. Although I had previous experience with Git, I had not worked extensively with a version control system as a group, which allowed me to learn and utilize Git at a more advanced level. Our goal was to evaluate an existing exam plan based on specific criteria, and since each rule had different evaluation criteria, we had to analyze the available data in various ways. For example, we leveraged regression for one of the rules. That was bit intensive and hard for me but I had fun while working on it. If I were to develop a similar program again, I would opt for a C-based programming language instead of Python.

**Kemal Öztürk**

Don't code the first thing that comes to mind: Rushing into coding without sufficient planning and consideration can lead to inefficiencies and complications later on. It is essential to take the time to properly analyze and design the software architecture before diving into implementation.

Explaining ideas properly to team members is crucial: Clear and effective communication is vital for efficient teamwork. It is essential to ensure that ideas and concepts are communicated clearly to avoid misunderstandings and confusion among team members. Encouraging active listening and providing regular opportunities for clarification can significantly improve collaboration.

%Part 7

Summary and Outlook

The technical results of the project are significant, as the team successfully implemented the core functionality of the application. The application provides a scoring system for evaluating exam plans based on predefined criteria, generating reports, visualizations, and identifying conflicts. The implemented solution demonstrates a good understanding of the problem domain and effectively addresses the primary objectives.

For the future development, one could refer to part 5.4, where it is broken down in details. But in nutshell: UI, better installability, more rules

Throughout the project process, regular meetings were conducted with the project supervisor and team members to discuss progress, resolve challenges, and align goals. The Agile and Scrum principles guided the iterative development, allowing for flexibility and adapting to changing requirements as needed.

By following this iterative approach and incorporating stakeholder feedback, the project evolved over time, resulting in an application that addressed the core objectives of scoring exam plans while accommodating potential future enhancements and improvements