

OTDK-dolgozat

2025.

Corvinus University of Budapest
Institute of Data Analytics and Information Systems



**Osztályzatelemzési keretrendszer automatizálása a felsőoktatási
képzések struktúrájának fejlesztésére**

*Automating Grading Analytics for Learning Pattern Recognition in Higher
Education Programs*

Anna Bartha

Supervisor: Dr. Vas Réka Franciska

Manuscript closed: November 11, 2024



Supported by the ÚNKP-23- 2-I New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.

Consultant: Dr. Reka Franciska Vas

2024



KULTURÁLIS ÉS INNOVÁCIÓS
MINISZTERIUM

Supported by the ÚNKP-23- 2-I New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.

Table of Content

1. Introduction	1
2. Practical application of a data analysis solution.....	2
2.1. Overview of the Grading Analytics Framework.....	2
2.1.1. Data Preparation	3
2.1.2. Design of metrics.....	4
2.1.3. LAD regression.....	5
2.2. Difficulties with the practical use of the framework.....	6
3. Technology overview	7
3.1. R and related technologies	7
3.1.1. R packages.....	8
3.1.2. Shiny.....	9
3.2. Version control	9
3.2.1. Basics of Git	10
4. Development of Business Intelligence solutions.....	11
4.1. Life cycle of BI solutions.....	11
4.1.1. Analysis	12
4.1.2. Design.....	12
4.1.3. Development.....	13
4.1.4. Implementation.....	13
4.1.5. Evolution	13
5. Technology Acceptance Model.....	14
5.1. Perceived Usefulness and Perceived Ease of Use	14
5.2. The evolution of TAM	16
6. Grading Analytics Application.....	16
6.1. Analysis and Design	16
6.2. Development	19
6.2.1. Introduction	19
6.2.2. Import data.....	20
6.2.3. Data preparation	22
6.2.4. Curriculum structure.....	25
6.2.5. Visualization.....	26
6.2.6. Metrics	28
6.2.7. Dataset	29
6.2.8. Modeling.....	30

6.3. Testing and error handling	35
6.3.1. Changes in course names	35
6.3.2. Error related to explanatory variables	35
6.3.3. Error related to outcome variables	37
6.3.4. No optimal model	40
6.4. Implementation	41
6.5. Evolution	43
6.5.1. Designing the interviews	43
6.5.2. Results of the interviews	45
7. Overview	46
8. Figure List	3
9. Table List	4
10. References	49
11. Appendix	52
11.1. Appendix 1 – Grading Analytics questionnaire - Dr. Ildikó Borbásné Szabó	52
11.2. Appendix 2 – Grading Analytics questionnaire - Dr. László Kacsirek	55
11.3. Appendix 3 – User interface of the Grading Analytics application	58

Figure List

Figure 1 - Result of LAD regression (Source: Bartha & Galambos, 2021)	5
Figure 2 – The 5 main phases of a BI System (Source: Agui et al., 2014)	11
Figure 3 - Technology Acceptance Model (TAM) (Source: Davis, 1987)	14
Figure 4 - Measurement Items for Perceived Usefulness (PU) and Perceived Ease of Use (PEOU) (Source: Warkentin et al., 2007)	15
Figure 5 - Application template and colors (Source: own source)	18
Figure 6 - Import data (Source: own source)	18
Figure 7 - Commits in GitHub (Source: github.com)	19
Figure 8 - Introduction page (Source: own source)	20
Figure 9 - Import Data page (Source: own source)	20
Figure 10 - Prepare data - filtering options (Source: own source)	23
Figure 11 - Dataset ERD (Source: own source)	24
Figure 12 - Data preparation result (Source: own source)	25
Figure 13 - Curriculum Structure page (Source: own source)	26
Figure 14 - Visualization page (Source: own source)	27
Figure 15 - Metrics page (Source: own source)	28
Figure 16 - Dataset page (Source: own source)	29
Figure 17 - Modeling page - Control (Source: own source)	30
Figure 18 - Modeling dataset (Source: own source)	30
Figure 19 - Modeling page - LAD results (Source: own source)	33
Figure 20 - Microeconomics II - Distribution of grades (Source: own source)	33
Figure 21 - Modeling page - Compare subjects (Source: own source)	34
Figure 22 - Error handling 1. (Source: own source)	36
Figure 23 - Visualization page - warning message (Source: own source)	37
Figure 24 - Error handling 2. (Source: own source)	38

Figure 25 - Methods of Staff Selection course diagrams (Source: own source)	38
Figure 26 - Metrics page - warning message (Source: own source)	39
Figure 27 - Modeling page - no optimal models (Source: own source)	40
Figure 28 - Modeling page - example (Source: own source)	41
Figure 29 - RStudio - Publish to Server (Source: own source)	42
Figure 30 - Application in the browser (Source: own source)	42

Table List

Table 1 - Steps of data preparation (Source: own source)	3
Table 2 - Table of metric exponent (Source: own source)	4
Table 3 - Difficulties of applying the Grading Analytics framework (Source: own source)	7
Table 4 - R packages (Source: Grolemond, 2023)	8
Table 5 - Input file - Order of the columns (Source: own source)	21
Table 6 - Translated example of raw data (Source: own source)	22

1. Introduction

Business Intelligence (BI) plays a crucial role not only in the corporate world but also in the life of universities. Using BI tools can significantly impact their efficiency and decision-making processes. In the academic world, BI systems make it possible for institutions to analyze a great amount of data related to student performance, enrollment trends, and faculty activities. By using BI tools, universities can gain valuable insights into student demographics, course popularity, and graduation rates, allowing them to make informed decisions about curriculum development and resource allocation.

The successful implementation of a Business Intelligence solution is just as important as the quality of the underlying model. An effective BI system should not only include a proper data and adequate analytical model but also prioritize user-friendliness and accessibility. A user-friendly interface ensures that any stakeholder can easily navigate and extract insights from the BI platform.

In my research, I aim to explore the successful development and automation of a data analytics model. The data analytics framework has already been developed in a previous research project, where we identified and carried out the necessary data cleaning and processing steps and developed a model to examine the learning patterns and the structure of specific training programs. However, this process involves several manual steps and utilizes diverse platforms (Excel, R Tableau, SQL). This means that although the framework is ready, it cannot be used in everyday life because it requires a lot of time and technical knowledge. Therefore, in the current research I aim to answer the following questions: is it feasible not only to automate the entire process but to integrate it into a single platform? Is it possible to develop a solution that can be accessible to anyone without having to know any specific model building and technological knowledge? The goal is to improve efficiency by minimizing manual interventions and simplifying the workflow.

This BI solution will provide an easy to apply tool to analyze learning patterns in higher education programs. Having one platform for all the steps makes sure that the users do not need any specific data analytics or IT-related knowledge to use the Grading Analytics framework. Furthermore, the automated steps reduce the time spent on analyzing the data which enables educational institutions to use the model anytime they would like to see up-to-date results. To sum up, I believe that this automated and integrated framework can be a useful part of curriculum development in any higher education institution.

To answer my research question, I will develop the Grading Analytics application. Accordingly, the thesis consists of five major parts. First, I will introduce the problem that would be solved by the application. To do so, I will briefly describe the existing grading analytics framework. After that, I will examine the possible technologies that could be used to support the development process. As the third point, I will examine what strategies exist for the implementation of BI solutions according to the literature and what methodologies can be applied to develop an adequate application. Then, I will introduce the Grading Analytics application and go through all the functions it has. Last, but not least, I will show how the application can be used and include some feedback from users.

2. Practical application of a data analysis solution

In 2020 Ágnes Galambos and I started to work on a research project which was led by the Institute of Data Analytics and Information Systems at Corvinus University of Budapest. In our research, we sought to answer the question of whether a learning analysis process can be developed based on the grading data stored in the Neptun system used in Hungarian higher education to support data-driven decision-making. We were curious to see what information can be extracted from these data for individual courses and degree programs in general.

Although we managed to develop a framework to uncover correlations between courses, its practical implementation was not feasible. We spent a year working on obtaining a model from data cleaning to calculate the results. This required not only a long period but also significant technological and statistical knowledge. Therefore, if someone wants to get these results for future years or new programs, they would need not only a lot of time but also specific expertise. The problem was thus given: how to solve a data analysis problem from model building to practical application.

2.1. Overview of the Grading Analytics Framework

Before getting into the details of the automation, integration, and development processes of the framework, it is important to provide an overview of the Grading Analytics framework. In the next section, I will introduce the steps of it and the related technologies as well. Knowing the foundational elements of the framework will make it easier to have a better understanding of the application, its use, and its purpose. We have summarized and written down all the steps in our thesis work: *Developing a Grading Analytics Framework to Improve the Structure of Higher Education* (Bartha & Galambos, 2021). The following sections will shortly introduce and highlight the most important elements of the results.

2.1.1. Data Preparation

The source system for the initial data was the Neptun system of Corvinus University of Budapest. The university provided us with the student identifiers of the business informatics bachelor program who started their studies in the autumn semesters of 2011/2012, 2012/2013, and 2013/2014. In addition, we received the student results in thirteen mathematics and computer science related courses from the autumn semester of 2011. We did not have direct access to the database of the Neptun system, so all this data was anonymized and extracted in the format of Excel sheets.

The goal at this point was to identify and eliminate all data quality issues and then create a structure that could be used for further analysis. Table 1 summarizes all the steps that we made during the data preparation process. Having the data in Excel sheets, we decided to use Excel for this part.

Step	Action	Description
1.	Creating unified attributes	For each course and each student, we have identified these variables: gender, final grade, type of grade (practical or examination grade), number of courses taken, number of exam registration, and number of examinations.
2.	Handling data quality issues	Missing data: students who were admitted to the program but never started. Duplicates: students who were admitted to the program in multiple years. Inconsistency: courses taken in a semester when it would not have been possible. All these data were filtered and eliminated from the dataset.
3.	Creating the data structure	Joining data through classes and courses. Dimensions: Years, courses Measures: number of course registrations, number of exam registration, and number of exams

Table 1 - Steps of data preparation
(Source: own source)

After having a structured dataset, the next step was to analyze it. First, we wanted to gain a deeper insight into the data. We used descriptive statistical methods to obtain a comprehensive picture of the thirteen courses. We calculated the descriptive statistical indicators and plotted the distribution of each variable using graphs. These analyses were made using R and Tableau.

2.1.2. Design of metrics

Due to the narrow set of values of grades (1 to 5), the execution of statistical analyzes can in many cases only be used for illustration, and it cannot be specifically used for more complex analyses. Therefore, we considered it important to create a new performance metric that can place student performance on a wider set of values, all by expanding it with variables other than the grade and deepening its meaning. We examined which attributes of the cleaned, available data are relevant and can be easily incorporated into this metric. After a deep analysis, we decided to use the following variables: final grade, number of course registrations, and number of examinations. We also made a difference between practical graded or examination courses. In the case of practical graded courses, we reduced the value of the metric to a greater extent when a student took more exams. In the case of examination courses, we used the below formula for determining the value of the metric. x denotes the grade achieved, y the number of course registration, c its exponent, z the number of examinations, and d its exponent:

$$x + 1 - \frac{(1,5y^c + 0,5z^d)}{2}$$

In the case of practical graded courses, d receives a multiplier of 1,3 times, thereby increasing the weight of the exams.

$$x + 1 - \frac{(1,5y^c + 0,5z^{1,3d})}{2}$$

In the model, a and b are not the same for all grades. The weights for each grade have been constructed in a separate table to make the model easier to tune as can be seen in the below table.

Grade	Course admission exponent	Examination exponent
0	0,7	0,5
1	0,6	0,4
2	0,5	0,3
3	0,5	0,3
4	0,6	0,4
5	0,7	0,5

Table 2 - Table of metric exponent
(Source: own source)

Using the created formula, we have calculated the value of the metric for all the courses. Our aim was then to use these values to define optimal models in which the course chosen as

the outcome variable could be identified as being significantly affected by other explanatory variables. The metric values allowed us to use the LAD regression because this has increased sufficiently the set of values for the results.

2.1.3. LAD regression

In Least Absolute Deviation (LAD) regression analysis, a function can be defined that tries to estimate the chosen outcome variable using the explanatory variables. However, this was not yet appropriate for us, as we wanted to define an optimal model that only includes the variables that are significant. To this end, we decided to use the backward variable selection method and measured the goodness of fit using the AIC indicator.

These optimal models were also calculated using the R programming language. Since we could not find an available package with a function that met all the criteria that were mentioned before, we developed it ourselves using existing functions. Initially, we were only interested in examining the grades for each course, but there were other demographic data available in our database that we thought would be worth analyzing. Therefore, the models were extended with the gender and year variables. However, it is important to see that both are qualitative variables, categorical values coded by number. They could therefore only be treated as dummy variables in the LAD regression analyses.

We then used the curriculum to determine which explanatory variables belong to which outcome variables: in the initial models, we assigned the previously taught courses to each course. The LAD regression analysis was run on all these datasets resulted in the below matrix:

	β_0	Nem	Évf. 12-13	Évf. 13-14	anal	infa	statl	lin	szamt	szfl	valsz	adatb	opk	statll	szfll	irf
anal																
infa																
statl																
lin	0.28	-0.22			0.34****	0.36****	0.28****									
szamt	0.00				0.43****	0.18**	0.32****									
szfl	-0.64		0.64**	0.93****		0.45****	0.23*	0.27**	0.31**							
valsz	0.00						0.19**	0.39****	0.32****							
adatb	-0.53					0.16		0.31***		0.21**	0.29**					
opk	-0.11				0.16*		0.12	0.19*	0.20**	0.16**	0.12					
statll	-0.29						0.42****		0.29**	0.18*	0.29**					
szfll	0.78**					0.13			0.123	0.52****	0.18*					
irf	-0.26		0.38**					0.17*				0.21**	0.33****		0.25***	
ui	1.16****											0.18**		0.19***		0.39****

Figure 1 - Result of LAD regression
(Source: Bartha & Galambos, 2021)

The interpretation of the table is illustrated using the course Linear Algebra (lin). $\beta_0 = 0.28$ indicates that if all other variables are 0, then students are expected to get this metric value from the course. We can see that Gender (Nem) is also included in the model, but it is not yet significant at the 5% significance level, so the probability of the Type II error occurring is higher. This value means that by checking students who performed the same in the other courses, then males are expected to get -0.22 lower metric values than females. In addition, we can see that all three previous semester courses are included in the model, so for example, we can say that a 1 metric higher score in Analysis increases the Linear Algebra metric by 0.34 on average when all the other variables are the same.

The table therefore provides a good summary of the degree of association between the courses. It is important to point out, however, that this does not mean cause and effect relationship between them. We can only know for sure which courses are related to each other.

2.2. Difficulties with the practical use of the framework

As can be seen, the framework is capable of revealing connections between courses, but numerous difficulties occurred with its usability. The table below summarizes the manual steps that were required for each stage, the problems that had to be addressed, and the technologies that were used.

	Step	Technology	Description
1	Data cleaning and preparation	Excel, Power Query	The initial dataset only contained text descriptions of the results that had to be interpreted into numerical variables. Handling missing values and duplicates. Joining together all the separated tables based on courses and years to get the proper data structure.
2	Descriptive analysis, visualization	R, Tableau	The code had to be written for all the courses, but they could not be reused entirely if the names of the courses changed.
3	Curriculum structure	Excel	The Neptun system does not store the curriculum structure, however, this information is needed to build the initial models for LAD regression. This variable had to be added manually to the data.
4	Calculating metric values	Excel	The metric formula was interpreted into an Excel function that could be used on the dataset. However, it also had to be applied to all the courses we had.

5	Modeling	R	The modeling part consists of 3 stages: 1. For each outcome variable, a new dataset had to be created with only the necessary variables in a proper format. 2. Run the LAD regression function for all the courses. 3. Create a summary and visualize all the results.
---	----------	---	---

Table 3 - Difficulties of applying the Grading Analytics framework
(Source: own source)

The Grading Analytics framework would primarily support program directors and teachers. To make this method usable in practice, all steps need to be automated. It is also very important to integrate the whole process on a single platform, which minimizes the technological knowledge required.

3. Technology overview

I started to address my research question with literature review. First of all, I examined the possible technologies that could serve as a tool to automate the manual steps and integrate the whole process into one platform.

3.1. R and related technologies

R, introduced in 1993 is a free and open-source programming language that was ranked among the top 20 most popular languages over the past few years. It has a broad user base, however, it specifically targets data analysis and statistical usage. Additionally, there is a history of community-generated 'packages' in R software that expand the fundamental functionalities, which also resulted in better popularity among R users. This ongoing development and user-driven packages could potentially lead to changes and evolution in the use of R over time (Staples, 2023).

RStudio is an Integrated Development Environment (IDE) that can be used for programming in R because it provides a user-friendly interface for that. It makes it easier for users to write, test, and debug their code. The IDE is designed to improve the productivity of users by providing a centralized platform for data analysis, statistical modeling, and visualization. RStudio is available in both open-source and commercial versions (Beckman, 2021).

3.1.1. R packages

The CRAN repository is the official place for R packages. CRAN maintains technical consistency by enforcing rules like including ownership information, ensuring code works on various platforms (Windows, Mac OS, UNIX), and setting size limits for package components. Packages can also be found in GitHub or other repositories (Joo et al., 2020). There are lots of existing ones that can be grouped as follows: (1) load data, (2) manipulate data, (3) visualize data, (4) model data, (5) report results, (6) spatial data, (7) time series and financial data, (8) write high-performance R code, (9) work with the web, (10) write your own R packages (Grolemund, 2023). Regarding the Grading Analytics application, the first 5 groups are relevant. Table 4. gives an overview of packages that were important during the development.

Phase	Package	Description
load data	readxl	It provides solution in reading and writing Microsoft Excel files.
manipulate data	dplyr	The Dplyr package excels in data manipulation, offering several shortcuts for rearranging, summarizing, subsetting, and merging datasets.
visualize data	tidyr	R prefers the tidy format, and tidyr package contains tools for changing the layout of any data sets into this format.
	ggplot2	ggplot2 is R's well-known solution for making graphics. It allows the users to build layered, customizable plots.
model data	tidymodels	Tidymodels includes several packages designed for modeling and machine learning while adhering to tidyverse principles. This package consists of various general and specialized packages such as broom, parsnip, rsample, recipes, and more.
report results	shiny	It is a package that makes it possible to easily make interactive web apps with R. It provides a good way to share findings with non-programmers and explore data.

Table 4 - R packages
(Source: Grolemund, 2023)

3.1.2. Shiny

As we can see in the Table 4, there is a solution that provides a method that enables users to create interactive web applications. This package is called Shiny. It is free a framework that makes it possible to easily interface R with web browsers.

One of its main advantages is that users do not have to mix R code with HTML/JavaScript, and it gives access to all the capabilities of the R language. Because of this, these web apps can display any R-generated output. For instance, you can run complex statistical models and present the results in a user-friendly way in your browser. You can also fetch and combine data from different sources like SQL, XML, JSON, etc., and then organize process, and summarize the data based on your preferences. These features are typically not found in most commercial Business Intelligence tools (Resnizky, 2015, p 8).

3.2. Version control

A version control system serves the purpose of documenting modifications made to files over time, enabling tracking of changes and retrieval of specific file versions at a later time. The most used solution is a Distributed Version Control System (DVCS), wherein clients not only retrieve the latest file snapshot but also replicate the entire repository, encompassing its complete history. Consequently, each clone functions as a comprehensive backup of all data, ensuring that if the server malfunctions, any client repository can be used to restore it (Chacon & Straub, 2023, p. 10-13).

Git, established in 2005 by Linus Torvalds, is the most used distributed version control system and has been under the maintenance of Junio Hamano ever since. It is free and open source and because of its flexible architecture, Git can be effectively integrated with almost any workflow. Once the (remote) repository is set up, it can be cloned (local) and then any changes can be added to it. Git can be extended with other technologies like GitHub. It provides a web-based hosting service for Git repositories. It makes Git more user-friendly by providing a web interface to view file changes. Furthermore, it contains an API that allows developers to integrate GitHub into their own applications and workflows. This means that this integration can also be done between RStudio and GitHub. Users can make pulls from and push to remote repositories in GitHub (Chacon & Straub, 2023, p 166).

Due to these reasons, version control can provide significant assistance even in smaller projects. I have decided to adopt this method during the development process. Accordingly, I aimed to initially review the necessary theoretical background.

3.2.1. Basics of Git

The initial step for using Git is to set up the repository. These are the two most common ways to do that: users can either clone an existing repository which can be done with the git clone command. Secondly, they can also turn a local directory into a Git repository, if it is not yet version controlled. This way there will be a remote repository that can be accessed by the user and a local directory which is also called a working directory (Atlassian, n.d. - Setting up a repository).

Commits are the main building block units of a Git project. This command creates a snapshot of the current changes in the project. These committed snapshots represent stable versions of the project, as Git will retain them unchanged unless specifically instructed otherwise (Atlassian, n.d. - Saving Changes/Git Commit).

Fetch or pull commands can be used to update the local repository and get the latest versions of the files from the remote one. The difference between fetch and pull is that the fetch command does not automatically merge any of the local work. It only downloads the data to the local repository, so this has to be done manually. If the users get to a point, where they would like to share their project or just want to save it to the remote repository to have a backup, then a push command can be carried out. This command only works if the user has write access to the remote repository (Chacon & Straub, 2023, p 52-53).

Git also provides the so-called branching feature. It gained popularity, particularly with the emergence of open-source projects. Users can create their own branch, which makes it possible to isolate their work. This enables individual developers to contribute features at their own speed, keeping the main branch untouched until after code review and merging. This is even more important when many people work on the same project because this can prevent them from ruining each other's work. If a new feature is completed and tested, then it can be merged back to the main (Split Software, n.d.).

To sum up, version control is important for both big and small projects. It helps keep track of changes, encourages teamwork, and maintains a history of the files. This way any state can be reverted which is a great help when it comes to problem fixing. In smaller projects, it makes handling code easier and lets users try out new things without messing up the whole project.

4. Development of Business Intelligence solutions

After clarifying the exact goal of this research and having an overview of the technologies that can be used to solve the problem, I wanted to see what methodologies can be found for Business Intelligence application development in the literature. This can provide a guideline for succeeding in my project.

In the literature, Business Intelligence (BI) originally came from Decision Support Systems (DSS), which refers to a computer system that helps managers in decision-making. DSS used to be on its own in organizations and did not always connect well with other systems. As decision support applications improved, Howard Dresner introduced the term BI in 1989, which is based on DSS and its related techniques. Unlike DSS, BI creates a complete decision-support environment for management (Rouhani et al., 2012).

According to KM consultant Tom Davenport (2005), BI is described as "IT applications assisting organizations in decision-making through technology for reporting, data access, and analytical applications". Generally, business intelligence refers to the technical infrastructure that collects, stores, and analyzes company data. This plays a vital role in transforming raw data into meaningful insights (Davenport, 2006). BI is not only about number crunching; it creates reports and synthesizes information, that enables managers to make well-informed decisions. Nowadays, it is essential to have an efficiently functioning BI to be able to navigate through business challenges (Agui et al., 2014).

4.1. Life cycle of BI solutions

Despite the growing investments and widespread adoption of Business Intelligence systems, there is still a concerning problem. Recent evidence shows that over 70% of BI projects fail to deliver the expected returns. This means that most of the organizations cannot take advantage of implemented business intelligence systems. The potential benefits of BI remain unused for many, which raises the question of how to implement it effectively or overcome challenges with these advanced systems (Mehrdad & Navid, 2023). As the BI market continues to expand, closing the gap between implementation and actual benefits becomes more and more important in order to get the advantage of data-driven decision-making.

To reach this goal, a methodology, and strategy must be inevitably followed when a BI solution is introduced. In Figure 2, the 5 main functions of a BI system can be seen.

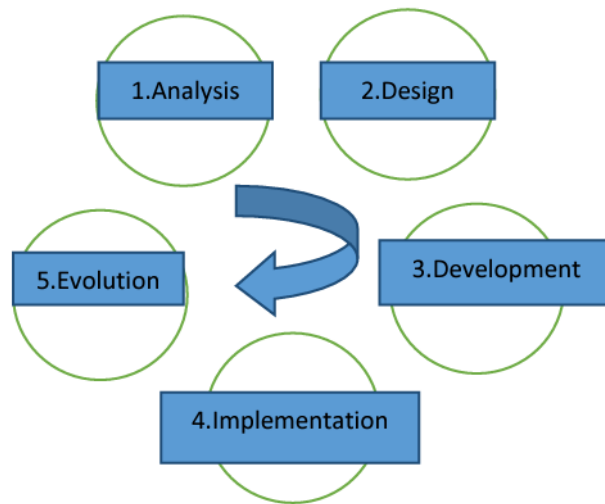


Figure 2 – The 5 main phases of a BI System
(Source: Agui et al., 2014)

4.1.1. Analysis

Implementing BI systems in organizations requires a clear vision that is aligned with business goals. This initial phase involves identifying informational needs, prioritizing them based on importance, and engaging key IT decision-makers. Although project goals are also important, they may be addressed at a later stage (Olszak & Ziemba, 2007). The evaluation of projects must consider technical and organizational perspectives as well. This makes sure that the new BI system follows the company's objectives. A BI project must effectively demonstrate the cost and benefits of solving a specific business problem. This can be measured by using predefined key performance indicators and insights from end users (Agui et al., 2014).

4.1.2. Design

The design stage is crucial for the successful implementation of BI systems and can be a time-consuming process. It includes the design of individual interfaces and makes sure that the overall logic is appropriate, and the entire BI application is consistent. The project plan should not only have the high-level design of different components of the solution but should also include the relevant data sources (Olszak & Ziemba, 2007). Furthermore, it is also essential to choose the right BI technologies based on the requirements of the users and the complexity of the implementation. A right design plan ensures the efficiency and effectiveness of the customized BI application (Agui et al., 2014).

4.1.3. Development

After the clear objective for the BI application is established and a detailed design plan has been made, the development stage can be started. This phase can be divided into two main components: the ETL process and the user interface. The development of the ETL phase means that the needed data is Extracted from the source that was identified during the design stage. After that, it is Transformed into a structured data model for further analysis and then it is Loaded into a layer where it can be accessed by the functions built on this data. In a BI application, the user interface refers to the layer of data and information presentation which is a crucial element of the system. This interface must prioritize user- friendliness and include all the functions that were identified in the design stage (Bara et al., 2009).

During the entire development phase, it is crucial that all parts are tested, and the results are compared with the key performance indicators. The application has to meet all the requirements that were set before (Olszak & Ziemba, 2007).

4.1.4. Implementation

Implementation is also an important step that can be divided into two parts. On one hand, the application has to be deployed to the user level where it can be accessed. In addition to this, creating final documentation and providing training and technical support is also an essential part of the process. This phase is some kind of bridge between development and practical use. The success of this stage is essential for ensuring that the BI application is ready for use and easy to understand for everyone (Bara et al., 2009).

4.1.5. Evolution

Contrary to its title, this stage is often referred to as the consumption stage. This means that the users start using the application which makes it possible to make informed decisions. The primary objective during this stage is measuring the success of the project. However, as the application is used, new requirements can arise or there can be requests for changes (Agui et al., 2014). To sum up, completing the final stage in the life cycle of BI solutions does not mean the end of the whole process. This cycle is iterative, as the demands or requirements change, the solution has to be re-evaluated, which begins with analyzing the needs and then goes on with designing and so on (Olszak & Ziemba, 2007).

During my research, I followed these steps in order to answer my research questions. However, following this cycle does not ensure that the application will be accepted and used

by the users. There are some methodologies that can be of great help with this problem. The Technology Acceptance Model (TAM) provides a guideline to avoid this potential risk.

5. Technology Acceptance Model

Nowadays, it is still a major issue why people reject or accept particular technologies. The situation is the same regarding the BI solutions, that is why it is crucial to have a better understanding of why people react one way or another. The Technology Acceptance Model (Davis, 1989) has become a predominant model for examining the factors of user acceptance. It provides a guideline for designing and implementing a technology that will be used in practice (Sharp, 2006).

5.1. Perceived Usefulness and Perceived Ease of Use

Davis (1986) suggested that users' attitudes toward specific systems are influenced by two key factors: Perceived Usefulness (PU) and Perceived Ease of Use (PEOU). PU is described as "the extent to which a person believes that using a particular system would improve their job performance" (Davis, 1989). This means that if people perceive the usefulness of a system and improve their performance then they will have a positive attitude towards it. This would result in an increased willingness to use the system. On the other hand, PEOU is defined as "the extent to which a person believes that using a particular system would be effortless" (Davis, 1989). Comparing the two factors, it was statistically proven that perceived usefulness was considered to be more important. Both perceived usefulness and perceived ease of use have a significant influence on behavioral intention, but if users did not see a technology as helpful, or it did not improve their job performance, it is irrelevant if it was easy to use (Davis, 1989).

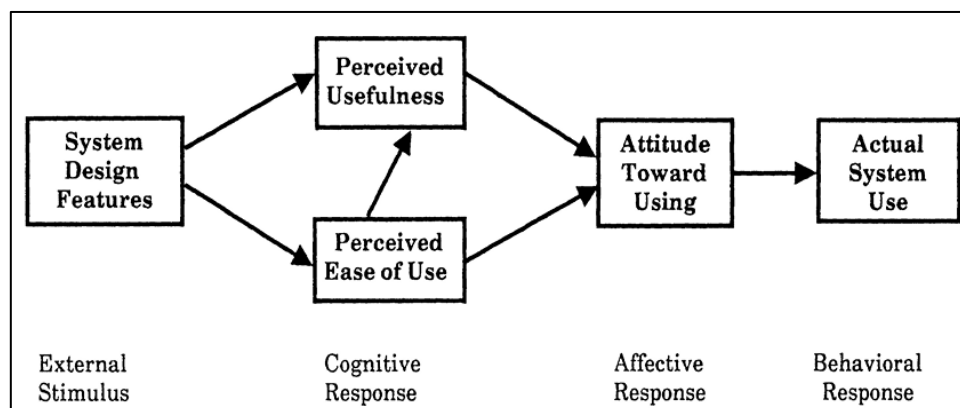


Figure 3 - Technology Acceptance Model (TAM)
(Source: Davis, 1987)

As can be seen on the Figure 3, system design features have an influence on both perceived ease of use and perceived usefulness which means that it also has an indirect impact on the attitude toward usage and actual usage behavior. In addition to this, perceived ease of use determines the perceived usefulness as well: even if the system could contribute to their job, if it is difficult to use then they will not be able to explore its benefits (Davis, 1987).

In order to get an overview of whether a technology would be used, it is important that both the PU and PEOU can be measured. The best method for this is creating a questionnaire. Davis's (1989) suggestion on this was to use standard 7-point semantic differential rating scales created by Ajzen and Fishbein (1980). A general approach was created by Warkentin et al. (2007) with the following items:

Perceived Usefulness (PU)	Construct	Perceived Ease of Use (PEOU)
Using [insert technology] in my job would enable me to accomplish tasks more quickly.	Item #1	My interaction with [insert technology] would be clear and understandable.
Using [insert technology] would improve my job performance.	Item #2	I would find [insert technology] to be flexible to interact with.
Using [insert technology] would improve increase my productivity.	Item #3	I would find it easy to [insert technology] to do what I want it to do.
Using [insert technology] would enhance my effectiveness on the job.	Item #4	Learning to operate [insert technology] would be easy for me.
Using [insert technology] would make it easier to do my job.	Item #5	It would be easy for me to become skilled at using [insert technology].
I would find [insert technology] useful in my job.	Item #6	I would find [insert technology] easy to use.

Figure 4 - Measurement Items for Perceived Usefulness (PU) and Perceived Ease of Use (PEOU)
(Source: Warkentin et al., 2007)

The PU items focus on enhancing task accomplishment speed (item #1), job performance (item #2), productivity (item #3), effectiveness (#4), job ease (#5), and the usefulness of technology (#6). Each of these metrics can contribute to the perception of usefulness. The PEOU items address clarity (item #1), flexibility (item #2), purpose (item #3), ease of learning (item #4), ease of becoming skilled (item #5), and ease of use (item #6). Activation of these indicators collectively shapes the perception of ease of use (Warkentin et al., 2007).

To sum up, replacing the [insert technology] with the actual system and creating the standard 7-point semantic differential rating scales will result in a questionnaire that can be used to measure PU and PEOU for any technology. At the end of my research, I would also like

to examine how useful and usable the developed application is. For this, I intend to use the method defined here.

5.2. The evolution of TAM

Since the introduction of TAM (1989), it has undergone various modifications and extensions in order to have a better understanding of technology and user behaviors. The model was extended by researchers. They added additional elements, like perceived enjoyment, trust, and subjective norms (Venkatesh & Davis, 2000). Understanding these extensions broadens the scope of TAM and can lead to a better framework for analyzing user acceptance in diverse technological contexts.

In addition to internal factors like perceived usefulness and perceived ease of use, external influences also play a significant role in technology acceptance. These external factors include facilitating conditions, social influence, and organizational support. Social influence underscores how subjective norms and others' opinions influence an individual's technology adoption decision. Moreover, organizational support, such as top management endorsement and technical support is crucial in creating a positive environment for technology acceptance. Additionally, facilitating conditions, such as the availability of necessary resources and infrastructure, can contribute to users' ease in adopting new technologies (Venkatesh, 2000).

6. Grading Analytics Application

At this stage of my research, I had a clear problem that had to be solved, I knew about the technologies that could be used and I also had two methodologies: one for the development life cycle and one for the user acceptance. I processed the necessary literature background, so the next step was creating the Grading Analytics application. I started to go through all the phases of the development lifecycle of business intelligence tools.

6.1. Analysis and Design

The whole development process started with clarifying the goals of the project. It is really important to differentiate between the Grading Analytics framework and the application because this thesis only focuses on the application. Accordingly, I already had a well-developed framework, but it was unusable for the targeted users. The purpose of this project was clear: create a solution that makes it possible for the program directors and teachers to use the framework in everyday practice without spending too much time with it and having to know

technical knowledge. To be able to succeed, first I needed to create a detailed design plan, that answers the following questions:

- How can the application be accessible to any users? How can the users interact with it?
- What technologies should I use for the development that can solve both the automation and modeling?
- How should the application look like (user-friendly interface)?
- How can I keep track of the development process and control the changes in the code?

The first question was easy to answer, as the internet makes it possible to access any application from anywhere. In addition, most higher education institutions have the infrastructure in place to do this, so if they want to make the app available, they have the possibility to host it. Accordingly, the answer was clear: I wanted to solve the problem through a web application.

To go on I had to consider what technologies can be used to create a web application. Nowadays, there are lot of programming languages, frameworks, and BI tools that provide solutions to this. I first thought about using BI tools like Power BI or Tableau. Unfortunately, I had to discard these solutions quickly as I wanted to create a dynamic application that could handle different course names, programs, years, etc. In addition, I wanted to allow users to have some level of interaction on how they build the final data set. The next limiting factor was the existing LAD regression code, as it was written in R. The chosen technology therefore had to be able to run it. At this point, I thought that if some code was already written in R, it would be worth using it for the whole application. There are packages in R that provide solutions to not only data preparation and automation but also building web applications. The Shiny package provides a great framework for building web-based solutions. That is why I chose R Shiny as my primary technology for development.

Once I knew I wanted to develop a web-based Shiny application, I started designing the user interfaces. First, I created the color palette and a template layout. Since the project was for Corvinus University of Budapest, I used the design elements of the institution. As can be seen in the image below, a menu bar was added on the left side, where users can easily navigate between the steps.

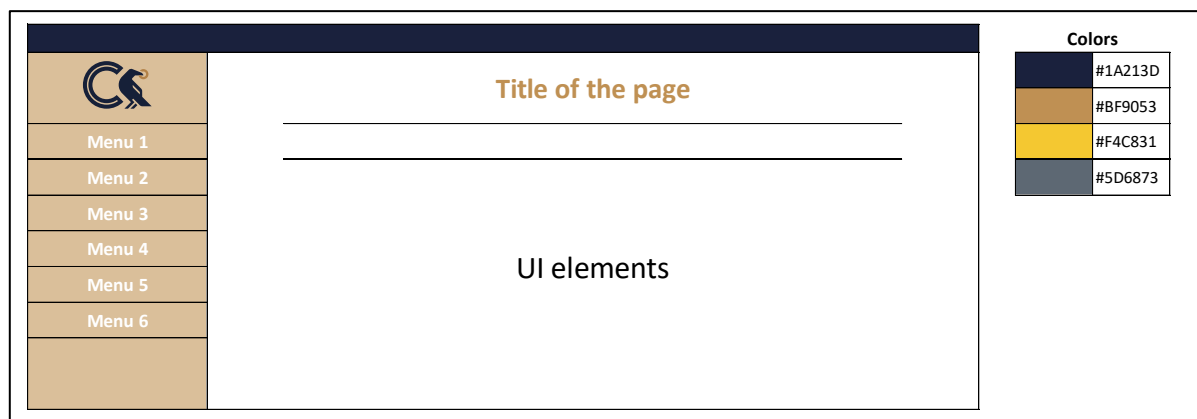


Figure 5 - Application template and colors
(Source: own source)

I used this template to design the pages of the application. During this process, I decided that each menu item would cover a step of the grading analytics framework. Accordingly, I developed the following tabs: Introduction, Import Data, Data Preparation, Curriculum Structure, Visualization, Metrics, Dataset, and Modeling.

During the development stages, I will present all menu items, but as an example, the design for the Import Data page is shown below. According to the plans, besides the title, the page will contain a file upload input box where users can select the document to upload. After clicking the Upload button, the file will be uploaded, and the data will appear in a table.

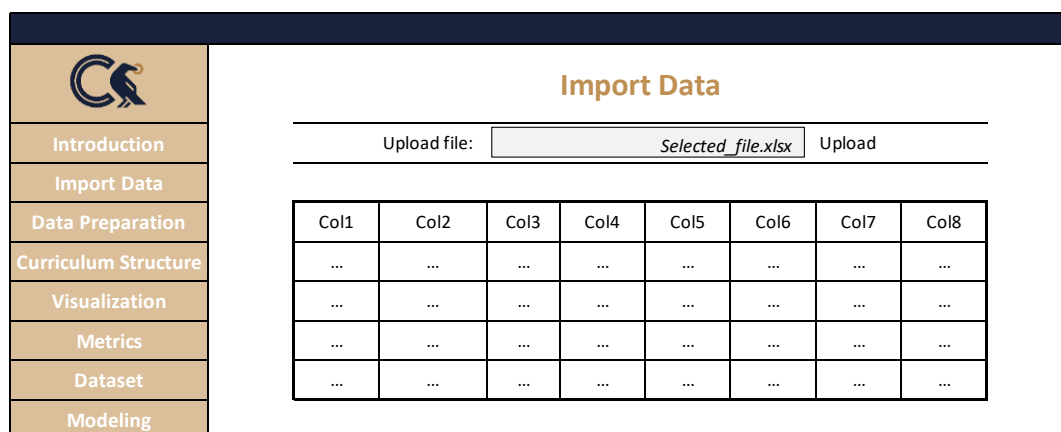


Figure 6 - Import data
(Source: own source)

Despite thorough planning, I felt it was necessary to track the development progress and have the ability to revert to a previous state at any time. I believed that this would help me with debugging because comparing changes in the code from a previous state would help identify the source of errors more easily. Additionally, I wanted the flexibility to experiment with any additional ideas that might arise during development without risking the functionality of

previously completed features. For these reasons, I decided to use version control for development. As a first step, I installed Git on my computer and then created a repository on GitHub to store the project files. Next, I set up integration between RStudio and GitHub, resulting in a local repository. Since I was working alone on the project, I only created a Development branch. I committed all changes to this branch during development, and when a larger feature was completed and successfully tested, I merged it back into the Main branch. Here can be seen some commits that were made during the development:

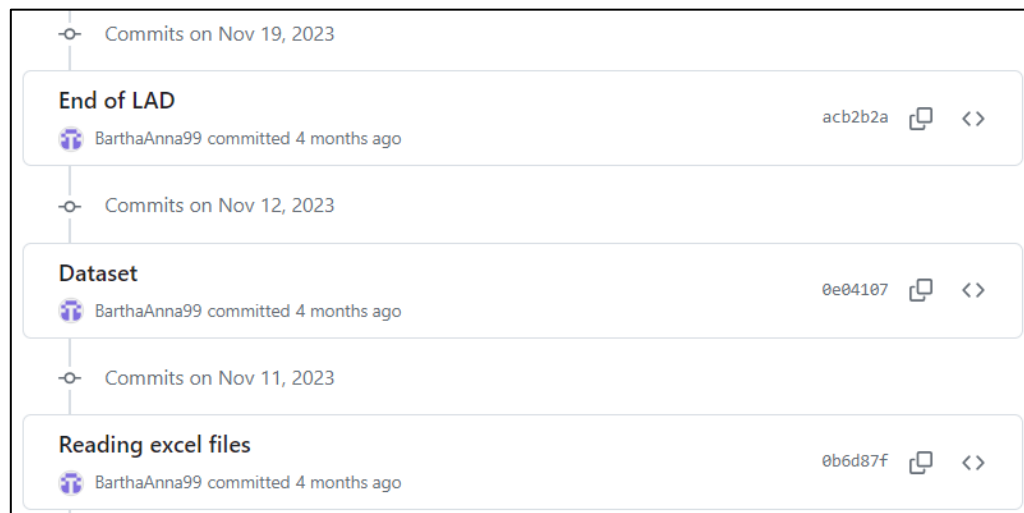


Figure 7 - Commits in GitHub
(Source: github.com)

At this point of the project, all preliminary plans and information were available, so I could start the actual development of the application.

6.2. Development

In the following chapters, I not only aim to explain the development process but to also present the finished application. I will introduce in detail each menu item, explaining the primary goal of the steps and what problem it automates or solves. Additionally, I will introduce the UI elements and their functions, as well as highlight the requirements users need to be mindful of when using the app. Furthermore, I would like to illustrate the description of the application with examples. For this purpose, I will use the data and results of the Applied Economics bachelor program.

6.2.1. Introduction

When the application is launched, the *Introduction* page appears. Its purpose is to provide the user with a very brief summary of what the Grading Analytics is all about. In terms of user interface elements, it only contains text and a flowchart, with no associated functionality. The

following text can be read under the title: The purpose of the Grading Analytics framework is to analyze the data stored in the Neptun system and uncover the relationships between courses, thereby aiding data-driven decision-making. With the help of the website, all steps from the initial analysis to the final model building can be performed.

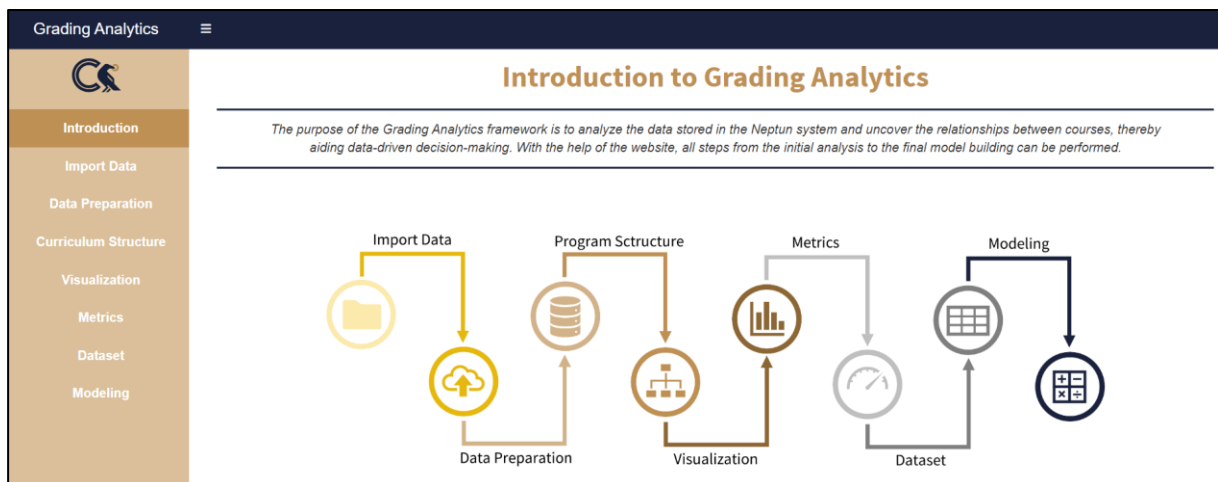


Figure 8 - Introduction page
(Source: own source)

6.2.2. Import data

The next menu item is *Import Data*. Its purpose is to allow the user to upload the data they want to analyze. I also designed an option to have a data view after uploading it. I implemented this using a table.

Program	NeptunID	Nem	Evfolyam	Tantargy	Jegy	Tipus	Datum	Felev
1 alkalmazotti kozgazdasagtan - 4BNAKGT17MBP	bb333fae4b367697eb684c8e3c78588074464515	Férő	2017/18/1	Matematikai analízis	Közepes	Vizsgajegy	2017-12-20T12:32:53Z	2017/18/1
2 alkalmazotti kozgazdasagtan - 4BNAKGT17MBP	bb333fae4b367697eb684c8e3c78588074464515	Férő	2017/18/1	Vállalatgazdaságtan	Közepes	Vizsgajegy	2017-12-13T15:55:16Z	2017/18/1
3 alkalmazotti kozgazdasagtan - 4BNAKGT17MBP	d17825821351b4f32148cd3ab1d85fd91723G	Férő	2017/18/1	Matematikai analízis	Közepes	Vizsgajegy	2017-12-20T12:32:53Z	2017/18/1
4 alkalmazotti kozgazdasagtan - 4BNAKGT17MBP	d959ae0d968b1c940da52fe5b75b3cc2c368cbe4	Nő	2017/18/1	Esszéírás, kutatósmódszertan	Jó	Oyakorlati jegy	2017-12-15T12:31:38Z	2017/18/1
5 alkalmazotti kozgazdasagtan - 4BNAKGT17MBP	670bb66fa14cc34b599e7a24380c5e6a5993a5fe	Nő	2017/18/1	Informatika I.	Jó	Vizsgajegy	2017-12-15T22:57:45Z	2017/18/1
6 alkalmazotti kozgazdasagtan - 4BNAKGT17MBP	6582d525646a3aaf4d62b5f0ad9a7bd942075fe	Férő	2017/18/1	Esszéírás, kutatósmódszertan	Alálrva	Alálrás	2017-12-10T00:00:00Z	2017/18/1
7 alkalmazotti kozgazdasagtan - 4BNAKGT17MBP	2a2f98d057dbc700cbfc8b725df798ab924249c	Nő	2017/18/1	Esszéírás, kutatósmódszertan	Alálrva	Alálrás	2017-12-15T00:00:00Z	2017/18/1

Figure 9 - Import Data page
(Source: own source)

At this point, I had to consider several questions and had to define certain constraints. Although my goal was to create a dynamic application, I had to predefine the structure of the input data. The Corvinus University of Budapest helped me with this. Since neither I nor the application had direct access to the source data, we had to figure out a format that could be generated by the university on one side and handled by the application on the other side. The final decision was to use the Excel format, with a maximum size limit of 30 MB. The file name is irrelevant to the application since we do not use this information later. However, the structure of the table is crucial for the application. I had to choose between two options: strictly defined field names or column order. In the end, I chose the latter because I saw a greater risk in the possibility of changes in attribute names, data being retrieved in a different language, or accents being missed. Therefore, for the application's further functionality to be usable, the variables must appear in the uploaded file in the following order:

Order	Name - Hungarian	Name - English	Description
1	Modulkód	Code of modul	Unique identifier of the programs
2	Modulnév	Name of modul	Name of the programs
3	Neptun kód	Neptun code	A hash code generated from the Neptun code of the students
4	Nem	Gender	Gender
5	Felvétel féléve	Year of admission	The semester when the students started his/her studies
6	Tárgy neve	Name of course	Name of the course
7	Tárgykód	Code of course	Code of the course
8	Bejegyzés értéke	Value of entry	A result that a student received from a course
9	Bejegyzés típusa	Type of entry	The type of the result
10	Bejegyzés dátuma	Date of entry	Date of the result
11	Felvétel típusa	Type of admission	Refers to the type of the course (elective, compulsory etc)
12	Félév	Semester	The semester when the given course was taken by a student

Table 5 - Input file - Order of the columns
(Source: own source)

After uploading the file, I only wanted to display the raw data, but after all, I performed a transformation at this step. I concatenated the name and code of the program and stored it in a variable. Later on, I will give an explanation why this step was necessary.

I would also like to demonstrate how the data is structured and what the individual variables mean through an example. As shown in the Table 5, I received the data in Hungarian, so I selected a case whose translation is shown in the table below.

Program	NeptunID	Gender	Year	Course	Grade	Type	Date	Semester
alkalmazott közgazdaságtan 4BNAKGT17MBP	20060903843f853e6492d834 b4c295290ab0bd37	Male	2017/18/1	Probability calculation	Fail	Exam grade	2018-05-23T14:05:33Z	2017/18/2
alkalmazott közgazdaságtan 4BNAKGT17MBP	20060903843f853e6492d834 b4c295290ab0bd37	Male	2017/18/1	Probability calculation	Fail	Exam grade	2018-06-12T11:19:28Z	2017/18/2
alkalmazott közgazdaságtan 4BNAKGT17MBP	20060903843f853e6492d834 b4c295290ab0bd37	Male	2017/18/1	Probability calculation	Satisfactory	Exam grade	2020-05-18T16:19:04Z	2019/20/2

Table 6 - Translated example of raw data
(Source: own source)

Each modification in the Neptun system represents a row in the table for each student and course. Therefore, the example above should be interpreted as follows: the given male student started his studies in the first semester of the academic year 2017/18, where he first took the Probability calculation course in the second semester of 2017/18, which is a graded subject. He first took the exam in May 2018, where he failed, and then again in June, also without success. Following this, in the second semester of the academic year 2019/20, he took the course again, which he successfully completed. In May, he achieved a satisfactory grade. Altogether, he had two course and three exam registrations.

6.2.3. Data preparation

After uploading the data, the users can compile the dataset they want to work with. For this, they can use three filters: they can only select one program, as the purpose of the application is to reveal the relationships between subjects within a program. They can do this using a drop-down menu. Concatenating the names and codes of the courses was important for this step. Since a course, such as Business Informatics, can be taken in both full-time and part-time formats, at bachelor's and master's levels, the name of the program does not determine exactly what the analysis will cover. The code of the program contains this information, so it is also necessary to determine that. Then additional options appear for selecting the years and subjects. Both are dynamic solutions, so for example if a year is deselected then all the subjects disappear that were taught only in that year. Another case can be if users change the program. In this case all the filters are refreshed and only the relevant data is presented. Furthermore, I

added Select and Deselect All buttons for both to make this step easier. This looks like the following:

Select a Program

alkalmazott közgazdaságtan - 4BNAKGT17MBP

Select Years

☒ 2017/18/1 ☒ 2018/19/1

Select All Deselect All

Select Subjects

☐ Alapszigorlat (7)

☐ Alkalmazott szigorlat (8)

☒ Esszéírás, kutatómódszertan (165)

Figure 10 - Prepare data - filtering options
(Source: own source)

When selecting subjects, it was important to display how many students took them in the selected program and years. LAD regression cannot be run on a dataset with null values, so only those student outcomes remain in the analysis where each outcome and explanatory variable has a result for the selected subject for each student. Therefore, if we select a course that, for example, only 7 students have enrolled in, then for certain analyses, we limit the dataset to only these 7 cases. I tried to create a solution that supports the selection of courses accordingly. When loading the courses, rows are automatically selected where at least half of the total headcount have enrolled in the subject. Of course, this does not guarantee a 100% solution, so the user needs to review this. However, I believe that as a starting point, this can be very useful, as in most cases only a few additional modifications need to be made.

After all filter settings have been adjusted, users need to click on the "Process data" button located at the bottom of the page. At this point, a lot of operations run in the background. In the first step, we determine the subset based on the filters that the user has set. Following this, the data cleaning and transformation steps described in section 2.1.1 are carried out. It is important to emphasize that all of this happens automatically, so there is no need for any manual intervention or work. The code performs all necessary steps in a few seconds. Although I have not worked with a relational database, I thought it would be very helpful to predefine the data structure and all variables using an Entity Relationship (ER) model. This plan is shown on Figure 11. below.

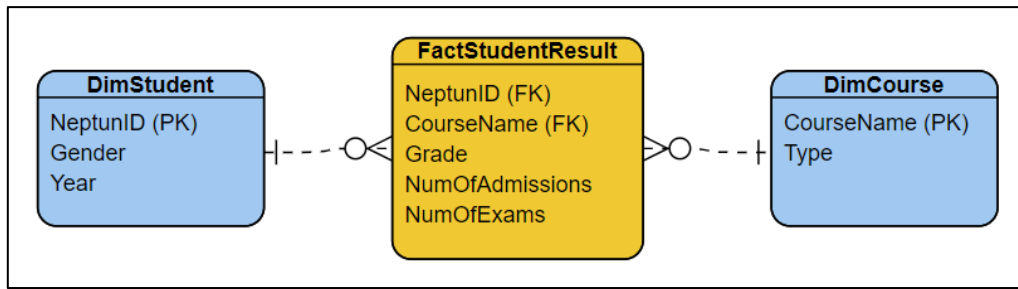


Figure 11 - Dataset ERD
(Source: own source)

I wanted to create these objects first, so I identified the students. I filtered out all unique NeptunIDs by academic year and then checked if any IDs appeared multiple times. These cases indicate that someone has applied to the same program multiple times. I removed these entries from the dataset. Next, I collected the courses and determined whether they were practical or examination-graded subjects. The challenge here was that, for example, if a student only obtained a signature for an exam-based course, in this case, it would appear as "Signed" under the course type. Therefore, if I only selected unique course names and types using `DISTINCT`, it would result in duplicate values for courses. In this step, I wrote a query that provided a result where courses were unique, and the type was either Exam or Practical grade for each of them.

Once the dimensions were available, I started working on the fact table, which contains three calculated values: final grade, number of course registrations, and number of exams taken. I obtained the final grade by calculating the date of the last entry for each student and course and then determining the grade the student received at that time. I have converted textual results into numerical values, such as 'good' becoming 4, to facilitate their use in subsequent analysis. Next, I counted, for each course, how many distinct semesters a student had entries in, which determined the number of course registrations. Then, using a similar method, I checked how many records belonged to a student-course pair when a grade was issued. This provided the number of exams taken. Through these steps, I managed to determine all the necessary variables, which I stored in a data frame called `GA_dataset` and displayed to the user in a table format as can be seen in the 12. figure. The column names can be interpreted this way: Year, NeptunID, Gender, Name of the Course, Type of the Course, Final Grade, Number of Examination, Number of Course Registration.

Show15▼entries

Search:20060903843f853e6492d834b4c295290ab0bd37

Evfolyam	NeptunID	Nem	Tantargy	Tipus	Jegy	Vizsgafelv	Targyfelv
2017/18/1	20060903843f853e6492d834b4c295290ab0bd37	Férfi	Valószínűségi számítás	Vizsgajegy	3	3	2
2017/18/1	20060903843f853e6492d834b4c295290ab0bd37	Férfi	Vállalati pénzügyek	Vizsgajegy	1	5	3
2017/18/1	20060903843f853e6492d834b4c295290ab0bd37	Férfi	Vállalatgazdaságtan	Vizsgajegy	2	1	1
2017/18/1	20060903843f853e6492d834b4c295290ab0bd37	Férfi	Számvitel alapjai	Vizsgajegy	3	6	4
2017/18/1	20060903843f853e6492d834b4c295290ab0bd37	Férfi	Statisztika II.	Vizsgajegy	2	6	3
2017/18/1	20060903843f853e6492d834b4c295290ab0bd37	Férfi	Statisztika	Vizsgajegy	2	3	1
2017/18/1	20060903843f853e6492d834b4c295290ab0bd37	Férfi	Pénzügytan	Vizsgajegy	5	6	4
2017/18/1	20060903843f853e6492d834b4c295290ab0bd37	Férfi	Ökonometria II.	Vizsgajegy	2	1	1
2017/18/1	20060903843f853e6492d834b4c295290ab0bd37	Férfi	Ökonometria I.	Vizsgajegy	3	1	1
2017/18/1	20060903843f853e6492d834b4c295290ab0bd37	Férfi	Nemzetközi gazdaságtan	Gyakorlati jegy	2	3	3

Figure 12 - Data preparation result
(Source: own source)

For testing purposes, it is worth examining the case presented as an example in section 6.2.2. Based on the raw data, we found that the student took the Probability Theory course twice and achieved a grade of 3 after three exam attempts. As shown in Figure 12, there is also a Search feature associated with the result table. Using this, I was able to filter for the relevant student (hash code: 20060903843f853e6492d834b4c295290ab0bd37). The table contains the results for Probability Theory in the first row. We can see that the application returned the same result as previously determined: Final Grade/Jegy – 3; Exams taken/Vizsgafelv – 3; Course registration/Targyfelv – 2.

6.2.4. Curriculum structure

At this point, the initial prepared dataset is already available, but a particularly important piece of information is still missing: to perform the LAD regression, it is necessary to determine in which semester each course can be taken. This data is not stored in Neptun, so it cannot be found in the source Excel file. The curriculum has to be examined in order to determine which semester each of these courses belong to within a program. This is a step that could not be fully automated because there is no available data source, so this information must be provided by the user each time. The goal in this case was to make this process as simple as possible and to enable the user to verify it within the application, to minimize the possibility of errors.

Program Structure

Specify the Semesters

Esszéírás, kutatómódszertan	1 ▾
Informatika I.	1 ▾
Makroökonómia	2 ▾
Matematikai analízis	1 ▾
Nemzetközi gazdaságtan	3 ▾
Pénzügytan	2 ▾
Statisztika	2 ▾
Statisztika II.	3 ▾
Számvitel alapjai	3 ▾
Valószínűségszámítás	2 ▾

>

The Structure

1 . semester	Esszéírás, kutatómódszertan, Informatika I., Matematikai analízis, Vállalatgazdaságtan, Mikroökonómia I.
2 . semester	Makroökonómia, Pénzügytan, Statisztika, Valószínűségszámítás, Mikroökonómia II.
3 . semester	Nemzetközi gazdaságtan, Statisztika II., Számvitel alapjai, Vállalati pénzügyek
4 . semester	Ökonometria I.
5 . semester	Ökonometria II.

Figure 13 - Curriculum Structure page
(Source: own source)

This page is empty when the application is loaded. The user interface elements only appear when the GA_dataset data frame is created and contains data. This way I made sure that this step can only be executed when the users have finished with the data preparation steps. If they did so, then on the left side of the page, all the courses are listed, each with its drop-down menu. These menus only contain numbers starting from 1, which determine the semester. With this solution, I managed to minimize potential data entry errors, so only content-related issues may arise if the user fills them out incorrectly. I tried to assist with this by visualizing the inputs in a table after clicking on the arrow. The application lists the semesters in ascending order in a table, along with the courses assigned to each semester. I tried to enhance this with conditional formatting to better distinguish between different semesters. This way, it is easy and quick to review how the selected program is structured based on the provided data.

Behind the scenes, in addition to visualization, another step occurs here. I associate semester data with the course names in the previously mentioned GA_dataset data frame. This means that I added a new column to it, called Semester.

6.2.5. Visualization

I considered it important that after data preparation, users have the opportunity to better understand the data. Although the primary purpose of the application is to uncover relationships between courses and help in program development, I believe that it can be very useful to see aggregated results for individual courses and programs. Accordingly, a *Visualization* page was developed, where we can gain a comprehensive view of the data through diagrams. These diagrams only appear when the underlying data is ready. So not only the data preparation step

has been executed but the curriculum structure has also been set. Otherwise, only the title can be seen here.

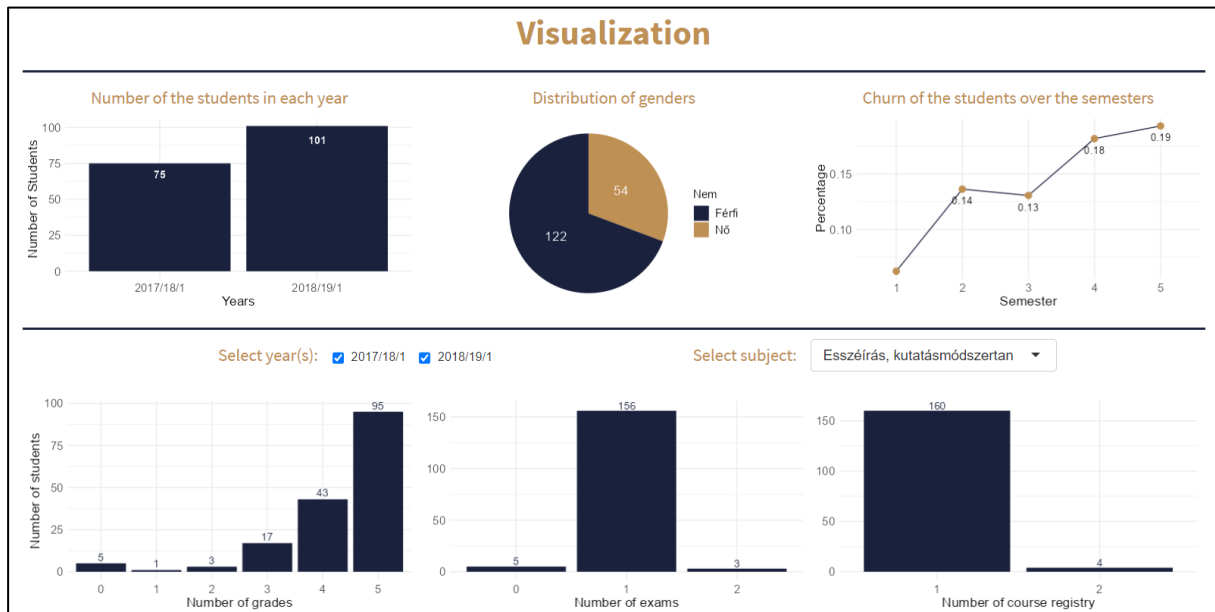


Figure 14 - Visualization page
(Source: own source)

In the first row, static charts are available, which relate to the entire program. The column chart shows how many students started in each year, while the pie chart describes the distribution of male and female students in the program. As it can be seen on Figure 14., in 2017 75 and 2028 101 students started the Applied Economics program. The "Churn of the students over the semesters" diagram illustrates how many students did not start each semester. Therefore, in the Applied economics program shown in the above picture, only 81% of students took courses in the 5th semester. These values were determined by calculating how many students started each course, and then determining the maximum for each semester from those. This maximum value was then compared to the total number of students. I used the maximum value because if a student registers in even one course in a given semester, they have actually started that semester.

In the second row, dynamic charts can be found that change based on the filters set (year and course). At this point, users have the opportunity to view grades, course registrations, and examinations for each course using column charts. The above example shows that almost all students passed the Essay Writing and Research Methodology course on their first attempt. Furthermore, we can also see that there is a heavily left-skewed distribution of grades obtained, indicating that the majority of students received a grade of 4 or 5.

6.2.6. Metrics

The next page aims to calculate the metric values described in paragraph 2.1.2. In this case, my goal was to create an interface that, on one hand, briefly summarizes the method of calculation, and on the other hand, provides a comprehensive overview of the results. As can be seen below, there is a short description under the title: Grades alone provide little information about student performance. The formula defined below, however, allows the calculation of new metrics that can be used to measure complex student performance in courses. It takes into account the number of exams and courses taken, the gender of the students, and their year. Furthermore, I also attached the formula of the metrics and a table about the index values. Under that, a button can be found. This element only appears if, the proper dataset has been created. If the user has completed both the data preparation and the setup of the curriculum structure, then the metric values can be calculated.

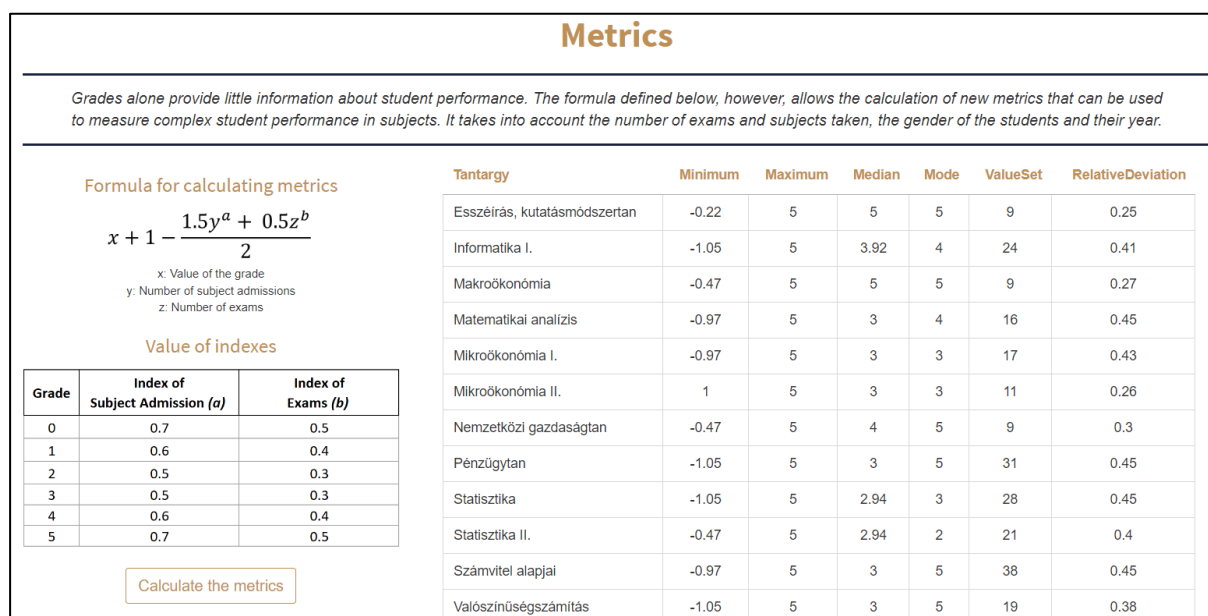


Figure 15 - Metrics page
(Source: own source)

To be able to calculate the metric values in the application, I had to translate the above formula into dplyr code. This required the use of many conditional expressions with "case_when". First, I stored which factors belong to each row based on the grades. Then, I examined the type of each course. If it was a practical course, I added an additional 1.3 multiplier to the factor for exam enrollments. Using these values, I expanded the GA_dataset with the Metrics variable, where I calculated the values.

I felt necessary not to end the process at this point, as it is important for the user to receive some feedback on what happened in the background. I believe that listing the metric values would not have been very helpful in this regard, as it would not have been possible to draw any conclusions due to the large amount of data. Therefore, I decided to calculate descriptive statistical indicators for each course. The following values were calculated: minimum, maximum, median, mode, value set, and relative standard deviation.

In this example, the second row represents the Informatics I course. The lowest-performing student achieved a metric value of -1.05. This means that the student failed to complete the course even after multiple attempts. The maximum value for the metric, according to the formula, is 5 (1 course enrollment, 1 exam attempt, grade of 5), which at least one student managed to achieve. The mode and median values are almost the same, indicating that most students obtained a grade of 4, while the median was slightly below this. There are a total of 24 different metric values for this course, significantly spreading the original scale of 5. This means that many students used multiple exam or course enrollment opportunities. The relative standard deviation for the specific course is 0.41. This can be interpreted as the grades being clustered around the mean. In other words, most students have similar performance in Informatics, with only small variations. This suggests that students' performance in Informatics is relatively consistent, with few outliers or extreme grades.

6.2.7. Dataset

Although in the previous paragraph, I mentioned that listing the metric values would be unnecessary, I still found it important to display the complete final dataset. This is the purpose of the *Dataset* page.

Dataset

Show10▼entries

Search:

	Evfolyam	Semester	NeptunID	Nem	Tantargy	Tipus	Jegy	Vizsgafelv	Targyfelv	Metrics
1	2018/19/1	1	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Esszéírás, kutatósmódszertan	Vizsgajegy	4	1	1	4
2	2018/19/1	1	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Informatika I.	Vizsgajegy	4	1	1	4
3	2018/19/1	2	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Makroökönómia	Gyakorlati jegy	5	1	1	5
4	2018/19/1	1	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Matematikai analízis	Vizsgajegy	5	1	1	5
5	2018/19/1	3	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Nemzetközi gazdaságtan	Gyakorlati jegy	5	1	1	5
6	2018/19/1	2	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Pénzügytan	Vizsgajegy	4	2	1	3.9201
7	2018/19/1	2	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Statisztika	Vizsgajegy	3	1	1	3
8	2018/19/1	3	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Statisztika II.	Vizsgajegy	4	2	1	3.9201
9	2018/19/1	3	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Számvitel alapjai	Vizsgajegy	5	1	1	5
10	2018/19/1	2	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Valószínűségszámítás	Vizsgajegy	5	1	1	5

Showing 1 to 10 of 2,355 entries

Previous

1

2

3

4

5

...

236

Next

Figure 16 - Dataset page
(Source: own source)

On this page, besides the title, there is only one table, which is displayed when the metric values are calculated, and the new variable is added to the GA_dataset data frame. The table allows searching between rows, and the records can be sorted according to any selected column. If the user wants to search between rows or only view certain values, they can do so here.

6.2.8. Modeling

Once the final dataset is available, the LAD regression analysis can be performed. This is found on the *Modeling* page, which is divided into three sections: control, result matrix, and course comparison. The control essentially means starting the modeling process. This is made possible by a "Calculate" button, as seen in Figure 17. It is important to note that this element appears only when the GA_dataset data frame, expanded with the Metrics variable, is available. As we have seen in previous cases, this ensures that steps can only be taken sequentially, preventing the code from encountering errors due to, for example, certain columns not yet existing.

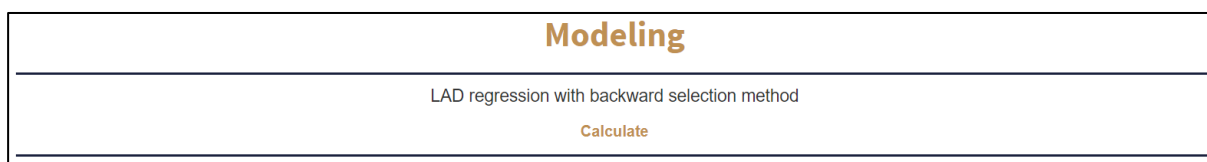


Figure 17 - Modeling page - Control
(Source: own source)

After pressing the button, the LAD regression analysis runs. My goal was to fully automate the steps outlined in section 2.1.3 and be able to represent the results in matrix format. I separated the two problems, and initially focused on ensuring that the regression analysis could run. As a result of our previous research, I had the R code that determines an optimal LAD regression model using the Backward selection algorithm, containing only significant explanatory variables. For this code to run successfully, we need to provide the course (dependent variable) we want to examine, and the dataset containing not only its results but also the data from previous semester courses (explanatory variables) in the following format:

valszt_model	lin_model	szamt_model	anal_model	infa_model	statl_model	Nem	Évfolyam	Évfolyam12_13	Évfolyam13_14
3,00	3,00	3,00	2,00	2,00	4,00	1	1	0	0
3,00	1,94	3,00	1,90	2,00	2,00	0	1	0	0
3,00	1,94	2,63	2,59	3,00	2,00	0	1	0	0
3,00	3,00	3,00	2,00	2,00	2,00	1	1	0	0
3,00	3,00	2,00	1,94	4,00	3,00	1	1	0	0
3,00	3,00	3,00	2,00	3,00	4,00	1	1	0	0
1,45	2,00	1,39	2,00	3,00	2,00	1	1	0	0
2,00	3,00	3,00	3,00	4,00	5,00	1	1	0	0

Figure 18 - Modeling dataset
(Source: own source)

There are significant differences in this structure compared to what is currently available. In the above case, the rows still represent the results of individual students, but now they only have one row associated with them. The results achieved in different courses appear not in new rows, but in columns. Therefore, the "Course" variable disappears, and instead, as many new columns are added to the dataset as explanatory variables we have, and the column names will indicate the courses. Additionally, it is crucial to note that although the "Gender" and "Year" columns remain, they must be treated as dummy variables, so we need to assign 1 and 0 values to the original records.

The further difficulty in creating this structure is that we cannot know in advance which courses will be included in the dataset, so we cannot work with fixed variable names. The problem is caused not only by the fact that different courses are taught in different programs but also within a program, the data structure on which the analysis runs constantly changes. For example, by selecting a course from the 2nd semester, all courses from the 1st semester will be included in the dataset. However, if we select a course from the 4th semester, we need to exclude all previously taught courses. Therefore, the problem was given: how can I iterate through all courses in a way that, by selecting each of them as a dependent variable, I generate the appropriate dataset, with the previously taught courses for the LAD regression analysis to run on.

In the zero step, I needed to modify the names of the courses. Since later on these will become column names and serve as variables in the analysis, they cannot contain any special characters, spaces, or accents. Therefore, I used a function I wrote to transform them. Following this, I filtered out all the courses that would become outcome variables and sorted them by semester. In practice, this meant filtering out those values where the semester is greater than 1, as we cannot identify explanatory variables for them. I stored the obtained result in a data frame. Using a for loop, I iterated through these values, thus going through all the courses starting from the second semester. Within the loop, I always selected the course corresponding to the given step from the data frame, so in the second iteration, for example, the second course became the outcome variable. Then, I filtered out the rows from the base dataset where the semester is less than the semester belongs to the outcome variable. I pivoted the result using the `pivot_wider()` function so that the results achieved from subjects appear not in rows but in columns. Additionally, I converted the Gender and Year variables into factors, so that they can be treated as dummy variables. With the result of these steps, the R code that determines the optimal

model could be executed. The loop terminates when there are no more courses left to select as outcome variables.

The next major challenge was that the original code printed the results to the console interface, so I needed to find a solution to display these results to the user in an easily interpretable format. The resulting matrix presented in section 2.1.3 adequately summarizes the obtained results, as it not only displays the beta values but also the corresponding significance levels. Therefore, my goal was to extend the already written code in a way that I obtain a similar result to the aforementioned matrix.

To achieve this, I needed a data frame whose columns always match the subjects of the given program, extended with β_0 , Gender, and Year variables. I created this empty data frame named `GA_result` before the start of the for loop. Then, I expanded the loop so that if the optimal model is determined, I store the obtained result: β_0 , course names, their corresponding beta values, and the significance level. Subsequently, I translated the significance level into * values using a case when statement and concatenated it with the beta values. This result was then added as a new row to the `GA_result` data frame. If the optimal model does not contain all the courses listed in `GA_result`, I left those columns empty. I named the rows after the outcome variable so that it can be tracked easily. As a result, in each iteration, a new row is created in the `GA_result` data frame, where the cells contain the beta values and significance levels for those courses that were included in the optimal model. The row names always represent the name of the course chosen as the outcome variable.

Once the loop has gotten to an end, I only needed to ensure that the `GA_result` data frame is properly represented. I felt it necessary to not only simply display the results but also to highlight those values that are worth paying more attention to. I implemented all this using a `DataTable`, which I managed to format correctly with JavaScript code. I used a darkening color scale to differentiate the significance levels, and I highlighted beta values greater than 0.3 with bold font. Additionally, I fixed the relative size of the table, resulting in some columns not being visible for many programs. I addressed this by adding a horizontal scroll bar and fixing the position of the first column that contains the output variables. This allowed any number of columns to be displayed.

	Beta	Nem	Evfolyam_2017/18/1	Eszeiras_kutatasmod	Informatika_I.	Matematikai_analizi	Mikrookonomia_I.	Vallalatgazdasagtan	Makrookonomia	Mikrookonomia_II.	Penzugytan	Statisztika	Valogasz
Makrookonomia	2.008****				0.197*			0.402***					
Mikrookonomia_II.	2****					0.151	0.311**						
Penzugytan	-0.236				0.557****		0.156	0.313*					
Statisztika	0.74					0.304**		0.185					
Valoszinusegsszamitas	0.201					0.539****	0.141	0.301**					
Nemzetkozi_gazdasagt	1.524****				0.333**						0.333***		
Statisztika_II.	1.337**	-0.596*					0.202*	0.292*	-0.367*			0.545****	
Szamvitel_alapjai	0.765			-0.209	0.387**	0.235*					0.347**		
Vallalti_penzugyek	1.366*			-0.209	0.308*				0.244				0.38
Okonometri_a_I.	0.897*									0.2**			
Okonometri_a_II.	1.669****						-0.139						

Figure 19 - Modeling page - LAD results
(Source: own source)

The application displays the results of the LAD regression analysis in the format shown in Figure 19. Let's take the course Microeconomics II, which appears in the second row, as an example. The β_0 value is 2 at a significance level of 0.01% (****). This indicates that if all other variables are zero, then students are expected to obtain a metric value of 2 from this course. This suggests that this is a passable course, which is also confirmed by the distribution of grades. As seen below, all students managed to obtain a grade, with only one failing Microeconomics II.

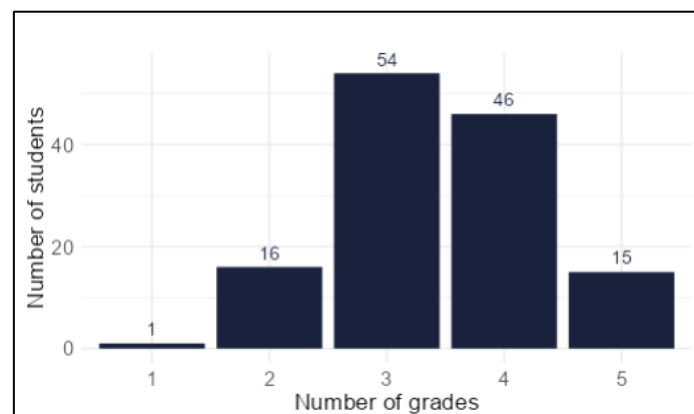


Figure 20 - Microeconomics II - Distribution of grades
(Source: own source)

Furthermore, we can see that among the explanatory variables, Analysis and Microeconomics are included in the optimal model. The beta value for Analysis is not significant even at 5%, indicating a higher chance of Type II error in this case. In contrast, for

Microeconomics, this value is significant at 1% (**). The corresponding beta value is 0.311. This means that if two students achieve the same result in all other course, but one of them performs 1 metric value better in Microeconomics, then he or she is expected to achieve a 0.311 metric value better result in Microeconomics II.

After examining the results, I concluded that it would be very useful to see the results of two courses at the same time. There were cases where I expected to see high beta values between certain courses, or conversely, there was a significant relationship between courses that I did not anticipate. Finally, I decided to enhance the Modeling page with a scatter plot. I placed it below the result matrix, extended with two dropdown menus where users can select the two courses they want to examine. The first value is placed on the X-axis, while the second is on the Y-axis.

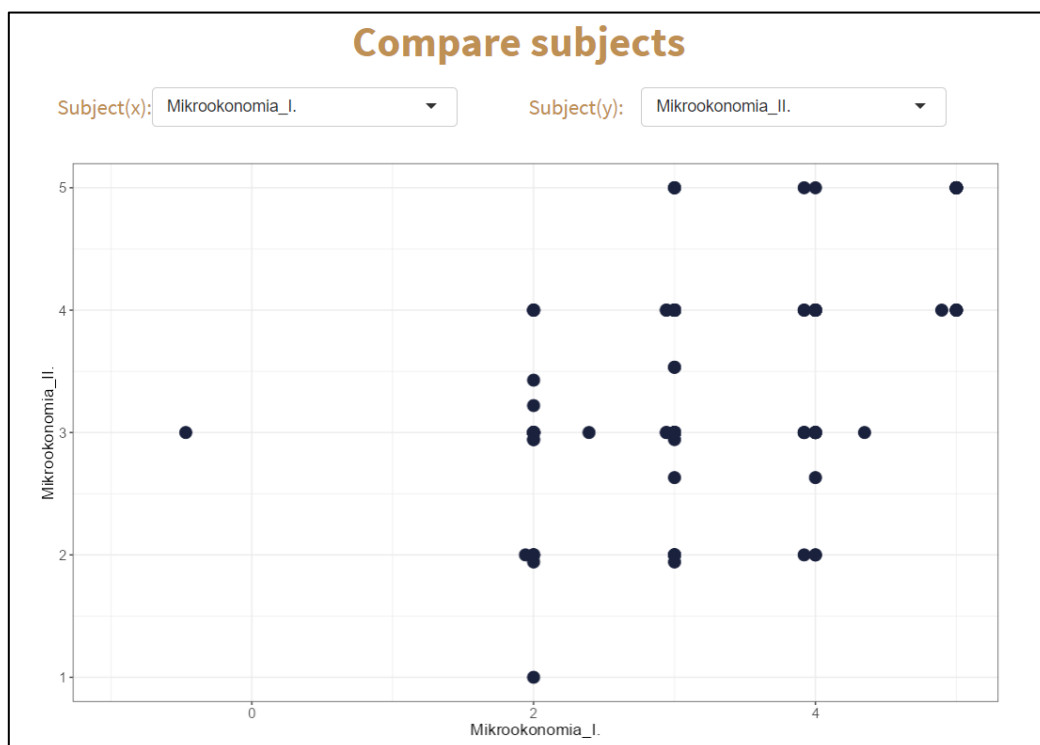


Figure 21 - Modeling page - Compare subjects
(Source: own source)

Returning to the previous example, I examined how the results of the Microeconomics II course change compared to Microeconomics. The scatter plot above clearly shows what also came out in the beta value: the tendency for students to achieve similar results in both courses.

With this, I have come to the end of presenting the basic functions of the Grading Analytics application. Overall, I have succeeded in developing a solution that enables the automation of the steps in the Grading Analytics framework, allowing analyses to be run at any

time and results to be easily extracted for users. Furthermore, the application ensures that the steps can only be performed sequentially, thus assisting the user in the process. If, after analyzing a program, we want to analyze another dataset, all previous results disappear after uploading, and we can only proceed to the next step again.

6.3. Testing and error handling

After completing the first version of the application, I found it crucial to thoroughly test it. The Institute of Data Analytics and Information Systems provided me with the results of 12 Hungarian-language full-time bachelor's programs. Thus, I had access to an Excel file containing all the mandatory and elective course data from these programs. This provided me with 12 test cases, which was an excellent opportunity to review the functionalities of the application multiple times. This proved to be very useful because it uncovered cases and issues that I had not encountered during development. In the following, I would like to present the 4 cases I faced during testing and describe the solutions developed for them.

6.3.1. Changes in course names

The first and most easily recognizable problem that arose was the course name changed from one year to another. An example of this occurred in the Business Informatics program, where the 2017/2018 year had courses named Mathematics I and II, while the 2018/2019 had courses named Analysis and Probability Theory. If the curriculum remains the same and only the name changes, it is necessary to standardize these names in the initial Excel file. In this example, I used only the names Analysis and Probability Theory for the 2018/2019 year instead of Mathematics I and II. Since it is not the purpose of the application to modify the original data, I did not want to implement this step within the application. If not only the names of the courses change but also certain courses are discontinued or new courses are introduced into the program, those years cannot be analyzed together, or the affected courses need to be removed from the dataset we want to analyze. This step can be performed within the application on the Data Preparation page.

6.3.2. Error related to explanatory variables

The next challenge was caused by the error occurring in the LAD regression analysis. At this point, the major challenge was to identify the underlying reasons for the failure. Eventually, I managed to develop a solution that narrowed down the possible causes of the error to three scenarios. Firstly, I extended the code of the Modeling page with a try-catch block. This allowed the application to continue running in case of an error, and instead of the result matrix, an error

message was displayed. Since often this message alone did not contain sufficient information, I further expanded it. On one hand, I also display which outcome variable the loop is currently at, so that the user is aware of which course is causing the problem. In addition, in case of an error, two analyses are run, which serve for the evaluation and diagnosis of linear regression models. Firstly, I used the `alias()` function to show the relationships between the variables present in the problematic model. Aliases, or the linear combinations of variables, are displayed, which helps to understand the relationships between the variables in the model. With the `vif()` function, it is possible to examine how the explanatory variables correlate with each other. High VIF values indicate significant multicollinearity between variables, which can result in unstable estimates and reduce the reliability of the model.

The extended code no longer stopped in case of an error but provided sufficient guidance to identify the cause of the problems. The first such case arose in the Human Resources program. When running the LAD regression analysis, I received the following message and results instead of the matrix.

```
LAD regression cannot be calculated. Error: the leading minor of order 4 is not positive Output variable: Makrookonomia

Model :
Makrookonomia ~ Nem + `Evfolyam_2018/19/1` + `Evfolyam_2018/19/2` +
  Informatika + Matematika_I. + Mikrookonomia + Tanulas_es_kutatasmodszertan +
  Vezetes_es_szervezes + Vallalatgazdasagtan

Complete :
              (Intercept) Nem `Evfolyam_2018/19/1` Informatika Matematika_I. Mikrookonomia
`Evfolyam_2018/19/2` 0          0 0              0          0          0
              Tanulas_es_kutatasmodszertan Vezetes_es_szervezes Vallalatgazdasagtan
`Evfolyam_2018/19/2` 0                      0                      0

Aliased values the leading minor of order 4 is not positive
VIF for each predictor cannot be calculated. Error: there are aliased coefficients in the model
```

Figure 22 - Error handling 1.
(Source: own source)

As we can see in the screenshot above, the code stopped at Macroeconomics, so we got an error already at the first course. The results of `alias()` can be found here and another error message. The VIF values could not be calculated for the given dataset due to the following error: "there are aliased coefficients in the model." In this case, the alias values show exactly what is causing the problem. The obtained results indicate that there is no linear dependence for the highlighted variable, which is currently the 2018/19/2nd academic year. So, I already knew exactly which variable was causing the problem, but the reason was still unknown. I returned to the Visualization page to see what data belongs to this year. Then I saw that the

problem was caused by the fact that only 1 student started in this semester, so it is understandable why this one result is independent. Fortunately, the solution was very simple because I just had to remove this semester on the Data Preparation page. However, since this error can occur in any program, I felt it necessary for the application to alert the user before the last step to remove such cases from the dataset. I solved this on the Visualization page. If there is a year with fewer than 5 students, a warning message appears below the column diagram displaying the years. This alerts the user that there are too few students in that group, so it is necessary to remove that year from the dataset.

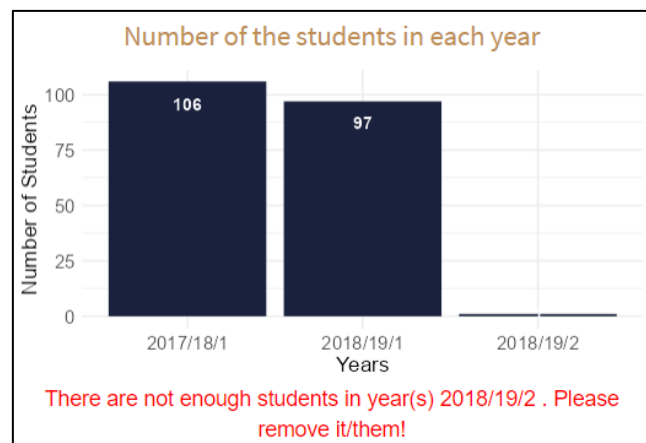


Figure 23 - Visualization page - warning message
(Source: own source)

6.3.3. Error related to outcome variables

The next case also arose in the Human Resources program. After removing the 2018/19/2 year, I reran the modeling, but the code entered an infinite loop, so I had to stop it manually. From the logs, I saw that the analysis was continuously running on a dataset where all explanatory variables had Inf/-Inf/Nan p-values. Therefore, as a first step, I extended the program with a condition that if such a case occurs, the code should stop and go to the catch branch. After that, I reran the modeling, and the results shown in Figure 24 were obtained. As we can see, in this case, the "Methods of Staff Selection" course posed a problem. However, the result of the alias() function was appropriate this time, so it could be inferred that the explanatory variables were fine. Furthermore, the VIF indicator values fell into an acceptable range. Values below 5 indicate low multicollinearity among the explanatory variables, so this could not have been the issue either.

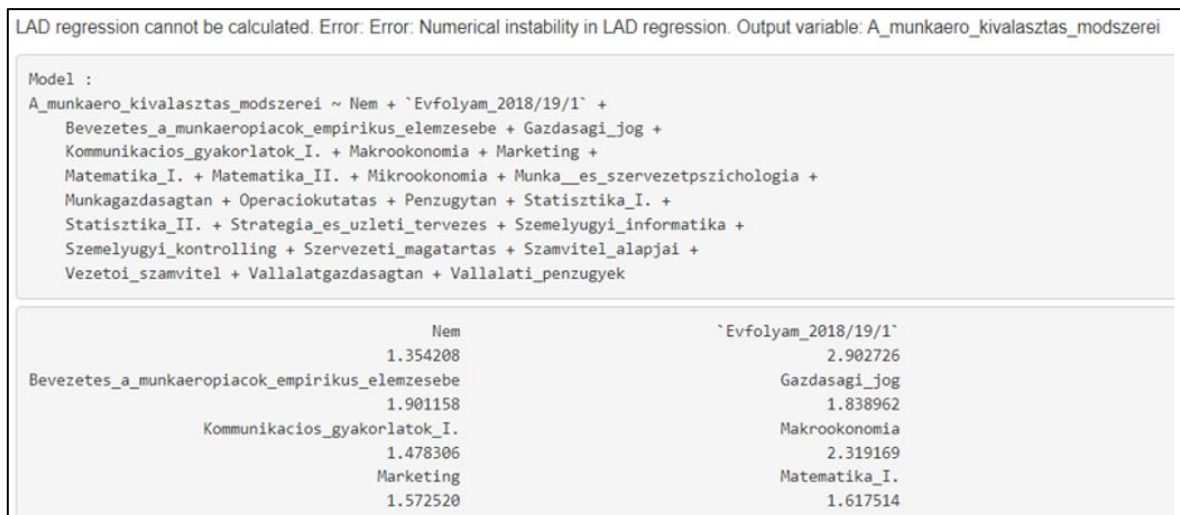


Figure 24 - Error handling 2.
(Source: own source)

Since I only received Inf/-Inf/Nan p-values during the modeling, I concluded that the cause of the error must be sought in the data of the outcome variable, supported by the results of the alias() and vif() functions as well. Just as I did in the previous case, I checked the Visualization page to see what values are associated with this course. Once again, I quickly received an explanation to why the modeling could not run. From Figure 25, it can be seen that out of 151 students, 150 finished the course with the same result. This explains the p-values because we tried to describe an almost constant value using the explanatory variables.

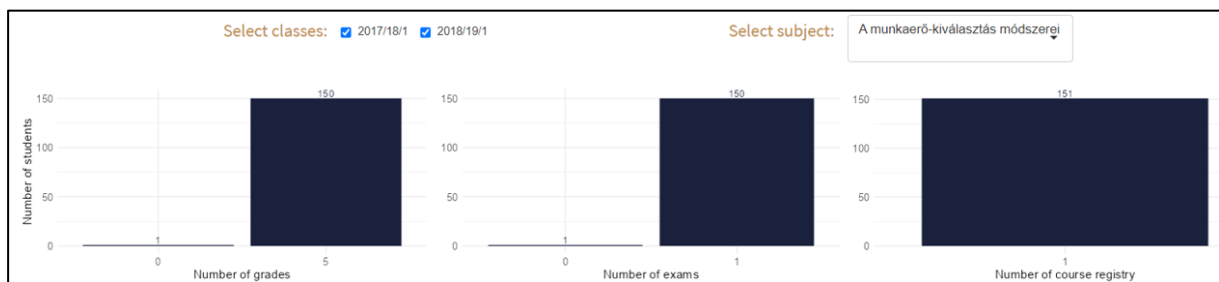


Figure 25 - Methods of Staff Selection course diagrams
(Source: own source)

The solution again meant removing this course from the data on the Data Preparation page. Once again, I concluded that it would be necessary to provide a warning to the user similar to the previous one. Although due to the modifications, the user will no longer get stuck in an infinite loop, it would still be useful for the app to draw attention at the first possible point to the fact that the particular course may cause a problem.

I decided to implement this step on the Metrics page, as this is the first point where metric values are available. I extended the code so that if there is at least one course whose value set

is less than 5 and/or the relative standard deviation is less than 0.1, then a Warning column appears in the summary table. In these cells, three different messages may appear: "The value set might be too low" if there are less than 5 different achieved results. "The relative deviation is too low" if the variance in grades is minimal. This indicates that most students likely achieved very similar results, with little deviation from the average. Additionally, if both conditions are true for a course, then both messages will appear.

As seen in Image 26, the application provided a warning for the "Methods of Staff Selection" course. I would like to emphasize that this is just a cautionary message, so it is always the user's responsibility to assess what further steps to take. They may decide to examine the problematic courses more closely on the Visualization page and then remove them from the data to be analyzed. Of course, there is also the option to proceed with modeling, but in doing so, they take the risk that the analysis may not run successfully. However, there is always the possibility of experimenting and error, as we have seen before; the application will not stop, only inform the user that modeling is not possible.

Tantargy	Minimum	Maximum	Median	Mode	ValueSet	RelativeDeviation	Warning
A munkaeö-kiválasztás módszerei	0.25	5	5	5	2	0.08	The value set might be too low. The relative deviation is too low.
Bevezetés a munkaeröpiacok empirikus elemzésébe	-0.22	5	4	4	7	0.23	

Figure 26 - Metrics page - warning message
(Source: own source)

For this type of error, the idea of having the application automatically filter out these courses arose, thereby making it easier for the user. I dismissed this approach because I believe that although these are similar cases, there can be various variations. It is possible that the value set is above 15, but the relative deviation is still very low. An example of such a case would be if 95% of students received a grade of 4, so the remaining 14 metric values are distributed among the remaining 5% of students. I thought it is necessary to examine these cases more closely, so the user has the opportunity to understand what the problem is. In addition, although, for example, "Methods of Staff Selection" will not be included in the result matrix, conclusions can still be drawn about it with the help of the application.

6.3.4. No optimal model

With the above developments, I was able to proceed with the testing; however, I encountered another problem. Despite the code stopping in cases of Inf/-Inf/Nan p-values, there was still an occurrence where the program entered an infinite loop again. In the background, the code continued to run without any error messages, but I did not receive any results. Once again, I checked the logs and noticed that the LAD regression analysis was running repeatedly on a dataset where only the values of the dependent variable were present. Upon closer examination of the intermediate results, I concluded that during each iteration, the model never met the conditions necessary for termination. Therefore, after each step, there were variables with p-values greater than 15%, or the AIC value increased instead of decreasing. All of this resulted in the code being unable to complete because it was not prepared for this scenario.

Thus, it was necessary to extend the logic for LAD regression modeling with another branch. Firstly, I check if an optimal model has been found; if yes, the code exits the loop, and the result is added to the GA_result data frame. If there is no optimal model, the analysis does not proceed automatically; instead, I first check if there are still explanatory variables in the dataset. I designed the final code so that if there are no variables left, it also exits the modeling process and begins analyzing the next course. Therefore, the code does not stop but can run to completion, and the user receives the matrix containing the results. However, it does not include the courses for which an optimal model could not be determined. I thought it would be better to highlight these variables and output them in a separate row underneath the results, as shown in Figure 27.

An optimal model could not be calculated for the following subjects: Szamitogep_architekturak ; Informaciomenedzsment ; Bevezetes_az_E_businessbe ; Integralt_rendszerek ; Folyamat_menedzsment ;

Figure 27 - Modeling page - no optimal models
(Source: own source)

In paragraph 6.3.3, I highlighted on why I do not consider it advantageous for the application to automatically eliminate variables, but rather beneficial for the user to be able to examine individual cases and make decisions based on their own judgment. However, in the current scenario, it may seem as if the code is doing just that. Therefore, I would like to emphasize that during the analysis, these courses are not removed from the dataset. Let's consider a course from the second semester as an example. If we cannot determine an optimal model for this one, it will not be included in the matrix. This is entirely understandable, as the purpose is to summarize the results of the optimal LAD regression analysis for a specific

training program. However, in this case, we cannot speak of an optimal result. In further iterations, however, this course will still be included as an explanatory variable because there is no numerical instability in the dataset; therefore, it may still have explanatory power. As it can be seen in Figure 27 and 28 Szamitogep_architekturak course is a good example for that. No optimal model was identified for that; however, it is included in the result of Szoftver_technologia with a high beta value and high significance level.

	nalizis	Informatika _alapjai	Mikrookono mia	Vallalatgazd asagtan	Vezetes_ es_ szervezes	Makrookon omia	Statisztika_ l.	Szamitastu domany	Szamitogep _architekt
Szoftver_ technologia		0.374**						0.304***	0.339***

Figure 28 - Modeling page - example
(Source: own source)

After all these modifications, I was able to successfully test all the programs for which data was available. I believe that a 12-round test is sufficient to declare that the application can handle the most common errors and cases. Thus, I have managed to answer the first research question, as I could develop an application that automatize the Grading Analytics framework on a single platform.

6.4. Implementation

After this, I focused on the next problem, which is how to make this accessible to anyone. Since the entire development was done in the Shiny framework, the resulting program is a web application. However, currently, it only runs on my own computer, but for it to be shared, the following conditions must be met. Firstly, a server is needed that can host the application. After this, all files need to be available in a package. This must include the ui.R and server.R files, as well as all other dependencies. Additionally, it may be advantageous to assign a custom domain to the application in order to make it more easily accessible.

Shinyapps.io is a cloud-based service provided by RStudio. Its aim is to provide an easy solution for deploying and sharing Shiny apps. For testing purposes, I created an account here for myself. I wanted to see if the development was all right, if I could deploy it, and then if it was accessible over the internet. To do this, I used the documentation available on the shinyapps.io website, which detailed how to perform each step. The set up for the deployment can be seen in the below figure. On the left side the files used during development can be seen, from which I had the option to select which ones I wanted to share. Since I stored only the necessary files in this folder, I had to select all of them. On the right side, you can see the

account where these are to be uploaded, and at the "Title" section, there is also an opportunity to name the page.

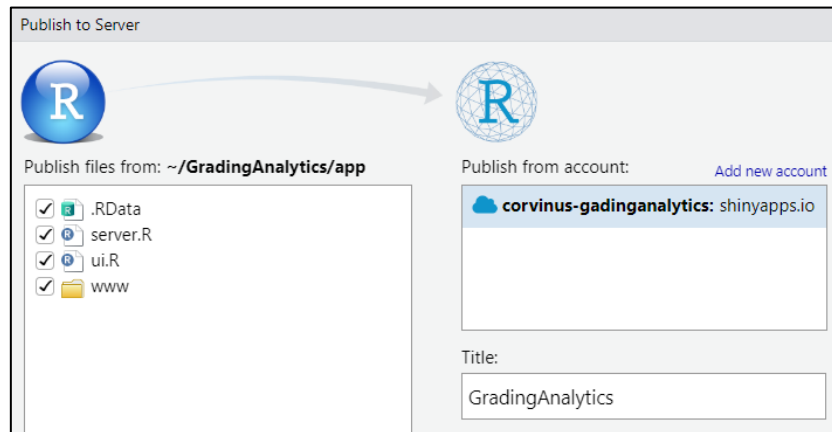


Figure 29 - RStudio - Publish to Server
(Source: own source)

As a result, using the deploy button in the RStudio interface, I was able to share the latest version of the application at any time. This can be opened from any browser anywhere, as long as the user has internet access. The result shown in Figure 29 was produced with the shared application. Everything works exactly as it was presented earlier. All steps can be executed; however, there is one limiting factor. Since I am currently using a free hosting service, only smaller-sized Excel files can be uploaded on this platform. Of course, this limitation applies only in this situation; with dedicated hosting and appropriate configuration, the 30-megabyte limit will still apply.

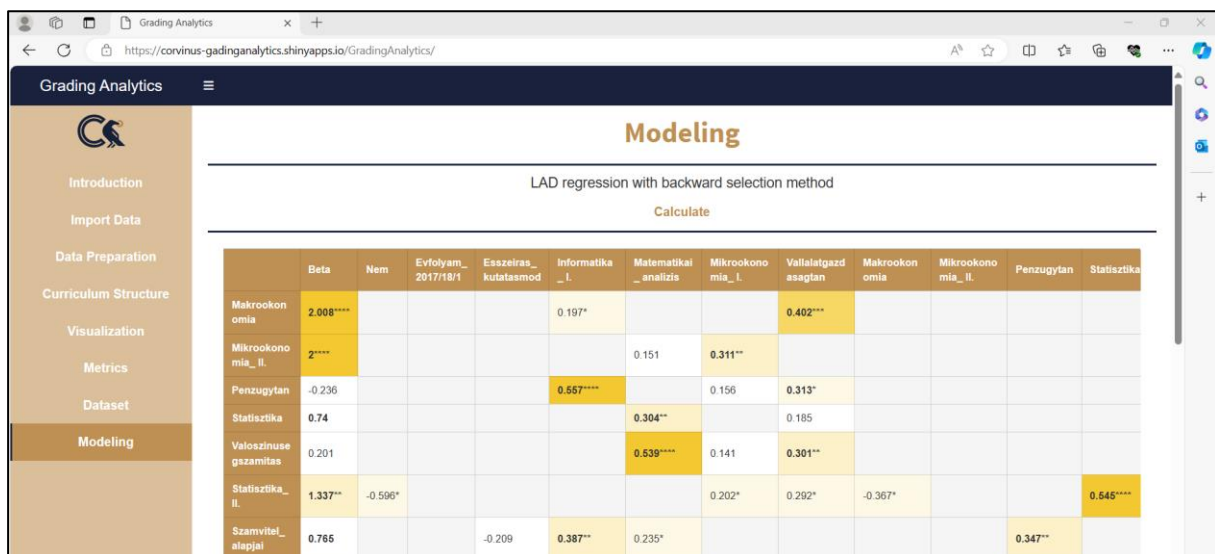


Figure 30 - Application in the browser
(Source: own source)

However, I would like to emphasize that the sharing was done only for testing purposes. I believe that this application does not belong on the Shinyapps.io server. If necessary and there is demand for it, it would be advisable for the higher education institutions to provide hosting for the application. Since this is a solution intended for internal use, they can ensure that it is accessible to the target users on the appropriate platform.

6.5. Evolution

Since it has been proven that the application can function on the internet, I focused on the last research question, which is how useful and usable the application is from a user perspective. I was curious about how actual end users perceive the development and what feedback they have. As I explained in paragraph 4.1.5, evaluation is a crucial part of the lifecycle of BI solutions. Therefore, I decided to request feedback through interviews.

6.5.1. Designing the interviews

Initially, I wanted to conduct a narrower survey. I expected that a smaller sample size of feedback could provide valuable insights and shed light on aspects that could further improve the application before a more extensive survey will be carried out. That is why I decided to conduct two interviews targeting the two main user groups: a program director and a course leader.

Dr. Ildikó Borbásné Szabó is responsible for several courses at the Budapest Corvinus University, and she is also in charge of a newly launched program. It was a great pleasure for me that she agreed to participate in the interview, as she could evaluate the application from multiple perspectives. Furthermore, most of the courses she teaches are related to the Business Informatics program, so she has expertise both in statistics and technology.

When selecting the other interviewee, it was important to choose someone with several years of experience as a program director. Additionally, I wanted to interview someone less connected to the field of business intelligence and information technology. Dr. László Kacsirek, the program director of International Business, agreed to the interview, which proved to be very useful. Overall, with these two interviews, I had the opportunity to hear perspectives from both course leaders and program directors. Additionally, I was able to examine how the attitude towards the application differs when someone has less statistical and technological background.

First, I designed the questionnaire I wanted to discuss with the interviewees. To do this, I used Figure 4 in paragraph 5.1 as a reference. This provided a suitable starting point for creating a questionnaire based on the TAM methodology. It consists of twelve statements,

where the interviewee has to decide on a scale from 1 to 7 how true each statement is. The first six questions relate to the usefulness of the application, while the second six measure usability. However, I wanted to complement this questionnaire with two open-ended questions. I thought it would provide valuable feedback if users had the opportunity to express in their own words what they think of the application. The first question also serves of the measurement of the usefulness of the development: "In your opinion, how could the application help with your work at the university?" The second one is related to the usability of the application: "How easy do you find the use of the Grading Analytics application?"

I decided to plan the interviews in advance to make the conversations as effective as possible. I decided to start by presenting the steps of the Grading Analytics framework. It was particularly important to ensure that the information related to metric values was properly conveyed. For the analysis to be usable, it is necessary for users to understand how these values are calculated and how they can be interpreted. In addition, I aimed to present the interpretation of the results of the LAD regression table in advance. At this point, I did not plan to explain the background of statistical methods, as one of the goals of the application is to enable users who do not have such knowledge to use it. However, it is very important that the results obtained are properly interpreted.

So, in both cases, I started the interviews with a theoretical summary. Therefore, we only moved on to the application when both Ildikó and László had no questions left about the framework. After that, I presented them the application itself and provided them with the opportunity to perform the analysis on the data related to their own courses. In this case, I considered it important to link the steps of the framework to the pages of the application in the first step. I always tried to explain the functions and the process in advance and interpret the results obtained. After going through all the steps together and discussing the results, I left some time for both interviewees to explore the application on their own. I tried to take notes throughout both conversations because it was already useful for me to know which points they asked about repeatedly or required longer explanations. In addition, several ideas and suggestions were raised that I thought could be very useful later.

In both cases, we only moved on to questionnaires after this. We reviewed the questions together, and if something was not clear, we discussed it. Then I sent them the questionnaire to give them time to think about the answers, and later, I could analyze the responses they sent back.

6.5.2. Results of the interviews

First, I would like to discuss the answers to the open-ended questions. The first one aimed to assess whether the interviewees believed the application could assist their work and, if so, how. From Ildikó's perspective as an instructor and course leader, she emphasized the usefulness of the solution: "this monitoring and reporting system provides a huge step forward analyzing the structure of both our courses and also my program." Additionally, she believed that the application could help her review student progress and uncover connections between courses that she had not previously considered. She also found it very useful to be able to verify these results with visualizations and compare the results of different years.

From the perspective of László, as a program coordinator, he highlighted how useful it is to see the connections between courses, which can greatly support the development of the prerequisite system within a program. Furthermore, the curriculum structures contain certain course chains. This means that courses build on each other over multiple semesters, culminating in a course at the end of the program. László thought that it could also be very useful to examine these with the help of the application. Overall, he believed that this solution could greatly support the development of curriculums.

I received different responses to the second question. According to Ildikó, using the application seemed easy, and the interface was well designed. However, she emphasized that it would be useful if the solution provided more support to the user. For example, tooltips and automatic interpretation of results could greatly enhance the user experience.

László's opinion on the usability of the application was as follows: "It seems to be very complicated and complex – probably a lot of time needs to be invested to understand the application in its complexity." However, he emphasized that it is definitely worth learning, but it is essential to have appropriate materials and support available. His suggestion for this was: "A step-by-step, on-the-screen learning aid would be very helpful – and much more practical than a long and detailed manual that is frightening to even read its table of content."

Overall, both of them believed that the application covers an area for which there is currently no solution at the university. This can support not only program coordinators but also course coordinators and teachers. Regarding usability, two opinions emerged. On one hand, they highlighted that the application has a well-designed and user-friendly interface. However, to ensure it is truly usable, it is necessary to provide additional support to users. This could be done through tooltips or short instructional materials.

These responses were also supported by the TAM questionnaire. For questions related to Perceived Usefulness, both interviewees gave a score of 7 for all the six statements. In contrast, different results were seen in the assessment of Perceived Ease of Use. According to Ildikó, there may be difficulties regarding how to interact with the application, and learning how to use it may also pose some problems.

From László's responses, it can be seen that although it is clear how to interact with the application, using it appropriately may present more serious challenges. Since this is a solution supporting more complex analysis, learning to use it can also be challenging. László believes that it is important to focus on this, otherwise it may be difficult to navigate within the application, and the user may not find what they want to do.

From these responses, it can be concluded that although the application is very useful and fundamentally well-designed, it is necessary to adequately support users. Ildikó's significant statistical and technological background, as mentioned earlier, is evident from the feedback. She had much less doubt about the usability of the application. However, László provided valuable feedback on how to support users who may find the operation of the application less transparent at first. Overall, I believe it is crucial to provide adequate support for users to learn how to use the application easily and quickly.

7. Overview

The Grading Analytics framework aims to leverage data extracted from the Neptun system. The analysis steps enable the calculation of metric values that illustrate the comprehensive student performance. These metrics encapsulate not only the final outcome but also the journey leading to it. This allows for more complex analyses, resulting in the development of an LAD regression analysis. Its outcome reveals the relationships between courses within a program and their respective strengths.

Consequently, the analysis can uncover valuable insights to support the work of program directors and course leaders in higher education institutions. However, the framework alone is not applicable. Significant technological and statistical knowledge is required for users to execute individual steps, not to mention the considerable time investment due to the many manual processes involved. Therefore, my research aimed to develop a solution that would make the Grading Analytics framework accessible to anyone. This means the automatization of all steps and integrating the entire process into one platform. The challenge here was to automatize both complex data cleaning and statistical modelling activities. It was also crucial

for this solution to be available through a user-friendly interface that is easily learnable. Furthermore, the upload and analyses of data of new training programs in the application raised new theoretical and methodological problems that all had to be solved in order to make the application truly usable for any program. Ultimately, I succeeded in resolving all issues and addressing my research questions. I developed an application that overcomes all previous limitations. With the Grading Analytics application, each step of the analysis can be fully automated, requiring users only to set the appropriate parameters. The determination of the curriculum structure is the only exception, as users need to specify which courses belong to which semesters individually. However, full automation of this process was not feasible due to a lack of suitable data sources.

Moreover, the application allows users to perform all steps within a single interface. The proper conditions ensure that the steps can only be executed sequentially, which reduces the possibility of errors. Error handling is designed not only for usability but also to address statistical errors. Since the aim of the solution was to be suitable for analyzing all courses, it was necessary to address situations where there is numerical instability in the data and analyses cannot be performed. A solution was developed that not only helps users identify the cause of the problem, but also supports its resolution. Consequently, the analysis can even be run on initially problematic datasets if the causes of the error are adequately handled.

The utility and usability of the application are also supported by feedback from end users. Through two interviews, course leaders and program directors provided feedback on the solution. In both cases, it was emphasized that there are currently few tools available for university staff to perform such statistical analyses. Therefore, the application can indeed create value by providing information about courses to users, thereby aiding in their work.

Regarding usability, positive feedback was received from interviewees. They viewed the application as having a nice, clean, and easy-to-use interface. However, they highlighted additional areas for improvement, with which I fully agree. Currently, the interface only allows for the execution of analysis steps and provides little support regarding the theoretical background of these steps. I also believe that enhancing the application with tooltips and descriptions to explain precisely what each step does would be highly beneficial. Additionally, it would be helpful if the application not only displayed the final results but also interpreted them. For example, in the case of LAD regression results, if a user selected a row, an interpretation related to the corresponding course would appear.

Overall, the most significant area for improvement lies in supporting users. This would entail not only descriptions for individual functions but also the interpretation of results. Furthermore, I believe it would be interesting to test the application in everyday use, which raises another question: how can this application be integrated into university infrastructure?

8. References

- Agiu, D. & Mateescu, V., & Muntean, I. (2014). Business Intelligence overview. Database Systems Journal, 5 (3)
- Atlassian. (n.d.). Setting up a repository. Atlassian Git tutorials. Retrieved from <https://www.atlassian.com/git/tutorials/setting-up-a-repository>
Downloaded: 2024.02.12.
- Atlassian. (n.d.). Saving Changes/Git Commit. Atlassian. Retrieved from <https://www.atlassian.com/git/tutorials/saving-changes/git-commit>
Downloaded: 2024.02.12.
- Bara, A., Botha, I., Diaconita, V., Lungu, I., Velicanu, A., & Velicanu, M. (2009). A model for business intelligence systems' development. Informatica Economica, 13(4), 99.
- Bartha, A., & Galambos, Á. (2021). Developing a Grading Analytics Framework to Improve the Structure of Higher Education (Bachelor's thesis, Corvinus University of Budapest)
- Beckman, M. D., Çetinkaya-Rundel, M., Horton, N. J., Rundel, C. W., Sullivan, A. J., & Tackett, M. (2021). Implementing version control with Git and GitHub as a learning objective in statistics and data science courses. Journal of Statistics and Data Science Education, 29(sup1), S132-S144.
- Chacon, S., & Straub, B. (2023). Pro Git (Version 2.1.413). Retrieved from <https://git-scm.com/book/en/v2>
Downloaded: 2024.01.06.
- Davenport, T. H. (2006). Competing on Analytics. Harvard Business Review 84(1), 98-107.
- Davis, F. D. (1987). User acceptance of information systems: the technology acceptance model (TAM).
- Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. MIS Quarterly. 13(3), 319-340.

- Grolemund, G. (2023). Quick list of useful R packages. Retrieved from <https://support.posit.co/hc/en-us/articles/201057987-Quick-list-of-useful-R-packages>
Downloaded: 2024.01.06.
- Maghsoudi, M., & Nezafati, N. (2023). Navigating the Acceptance of Implementing Business Intelligence in Organizations: A System Dynamics Approach. *Telematics and Informatics Reports*, 100070.
- Olszak, C. M., & Ziemba, E. (2007). Approach to building and implementing business intelligence systems. *Interdisciplinary Journal of Information, Knowledge, and Management*, 2(1), 135-148.
- Resnizky, H. G. (2015). *Learning Shiny*. Packt Publishing Ltd.
- Rouhani, S. & Ghazanfari, M. & Jafari M. (2012). Evaluation model of business intelligence for enterprise systems using fuzzy TOPSIS. *Expert Systems with Applications*, 39(3), 3764-3771.
- Sharp, J. H. (2006). Development, extension, and application: a review of the technology acceptance model. *Director*, 7(9), 3-11.
- Split Software. (n.d.). Feature branch. In *Split Glossary*. Retrieved from <https://www.split.io/glossary/feature-branch/>
Downloaded: 2024.01.28.
- Staples, T. L. (2023). Expansion and evolution of the R programming language. *Royal Society Open Science*, 10(4), 221550.
- Venkatesh, V. (2000). Determinants of perceived ease of use: Integrating control, intrinsic motivation, and emotion into the Technology Acceptance Model. *Information Systems Research*, 11(4), 342-365.
- Venkatesh, V., & Davis, F. D. (2000). A theoretical extension of the Technology Acceptance Model: Four longitudinal field studies. *Management Science*, 46(2), 186-204.

Warkentin, M., & Shropshire, J., & Johnston, A. (2007). The IT security adoption conundrum: An initial step toward validation of applicable measures. AMCIS 2007 Proceedings, 276.

9. Appendix

9.1. Appendix 1 – Grading Analytics questionnaire - Dr. Ildikó Borbásné Szabó

Details

Interviewee: Dr. Ildikó Borbásné Szabó

Program: BSc in Business Informatics (2BNGINF17MBP)

Subject: The purpose of the Grading Analytics framework is to analyze the data stored in the Neptun system and uncover the relationships between courses, thereby aiding data-driven decision-making. With the help of the website, all steps from the initial analysis to the final model building can be performed.

Questions

In your opinion, how could the application help with your work at the university?

As a lecturer, it gives me an overview of the students' progress. I can get insights about what correlations are among the different courses that are not expected to be. I can also test different scenarios to verify these correlations. In long term, I could compare the knowledge of students of different years.

As a programme director, I need to verify how the competences provided by the courses are acquired by the students. I can also test if the chains of courses – when courses are built on each other – are also in the students' thinking or just remain as planned. The input of this system highlights relationships between different courses. I can use this insight to redesign the programme. I can also compare the results of students of different years.

From both perspectives, this monitoring and reporting system provides a huge step forward analyzing the structure of both our courses and also my programme. It has a transparent layout, I only missed some tip boxes to make the concepts more clear. It collects data from the Neptun system, if it is streamed then it makes data collection more easier than it is now.

In summary, this is very useful application.

How easy do you find the use of the Grading Analytics application?

It seemed to be easy. The layout is well designed. Some tip boxes help understand the concepts. I also suggested to put emphasis on the explanations for users rather than on the technical knowledge (e.g. how the model was built). These knowledge could be stored or linked in a knowledge base.

Perceived Usefulness (PU)

The following questions relate to tasks regarding course and curriculum development, as well as program structuring. Please, answer them.

Using the Grading Analytics application in my job would enable me to accomplish tasks more quickly.

1	2	3	4	5	6	7
(Disagree)						(Agree)

Using the Grading Analytics application would improve my job performance.

1	2	3	4	5	6	7
(Disagree)						(Agree)

Using the Grading Analytics application would increase my productivity.

1	2	3	4	5	6	7
(Disagree)						(Agree)

Using the Grading Analytics application would enhance my effectiveness on the job.

1	2	3	4	5	6	7
(Disagree)						(Agree)

Using the Grading Analytics application would make it easier to do my job.

1	2	3	4	5	6	7
(Disagree)						(Agree)

I would find the Grading Analytics application useful in my job.

1	2	3	4	5	6	7
(Disagree)						(Agree)

Perceived Ease of Use (PEOU)

My interaction with the Grading Analytics application would be clear and understandable.

1 2 3 4 5 6 7
(Disagree) (Agree)

I would find the Grading Analytics application to be flexible to interact with.

1 2 3 4 5 6 7
(Disagree) (Agree)

I would find it easy to the Grading Analytics application to do what I want it to do.

1 2 3 4 5 6 7
(Disagree) (Agree)

Learning to operate the Grading Analytics application would be easy for me.

1 2 3 4 5 6 7
(Disagree) (Agree)

It would be easy for me to become skilled at using the Grading Analytics application.

1 2 3 4 5 6 7
(Disagree) (Agree)

I would find the Grading Analytics application easy to use.

1 2 3 4 5 6 7
(Disagree) (Agree)

9.2. Appendix 2 – Grading Analytics questionnaire - Dr. László Kacsirek

Details

Interviewee: Dr. László Kacsirek

Program: BSc in International Businesses (2BNNKGA17MBP)

Subject: The purpose of the Grading Analytics framework is to analyze the data stored in the Neptun system and uncover the relationships between courses, thereby aiding data-driven decision-making. With the help of the website, all steps from the initial analysis to the final model building can be performed.

Questions

In your opinion, how could the application help with your work at the university?

It could help in:

- Overviewing the relation/correlation between various courses (=subjects), e.g. a course and its prerequisite.
- Plan the prerequisites themselves.
- Assess the effectiveness of “chains” courses that build on one another, e.g. Business economics – Management – Strategic Management – Entrepreneurship, or (Macroeconomics – Global economy) + (Microeconomics – Business economics – Corporate Finance – International logistics) and International trade practices.
- Assess the effectiveness of the curriculum.
- If available: see and learn from practices of other programmes (e.g. Business Management).
- Further develop the curriculum.

How easy do you find the use of the Grading Analytics application?

- It seems to be very complicated and complex – probably a lot of time needs to be invested to understand the application in its complexity.
- For that reason, also learning to use seems to demand a lot of time.
- On its own that might not be a problem, but in times when time is a rare resource it may delay learning and thus applying the tool.
- A step-by-step, on-the-screen learning aid would be very helpful – and much more practical than a long and detailed manual that is frightening to even read its table of content.

Perceived Usefulness (PU)

The following questions relate to tasks regarding course and curriculum development, as well as program structuring. Please, answer them.

Using the Grading Analytics application in my job would enable me to accomplish tasks more quickly.

1	2	3	4	5	6	7
(Disagree)						(Agree)

Using the Grading Analytics application would improve my job performance.

1	2	3	4	5	6	7
(Disagree)						(Agree)

Using the Grading Analytics application would increase my productivity.

1	2	3	4	5	6	7
(Disagree)						(Agree)

Using the Grading Analytics application would enhance my effectiveness on the job.

1	2	3	4	5	6	7
(Disagree)						(Agree)

Using the Grading Analytics application would make it easier to do my job.

1	2	3	4	5	6	7
(Disagree)						(Agree)

I would find the Grading Analytics application useful in my job.

1	2	3	4	5	6	7
(Disagree)						(Agree)

Perceived Ease of Use (PEOU)

My interaction with the Grading Analytics application would be clear and understandable.

1 2 3 4 5 6 7
(Disagree) (Agree)

I would find the Grading Analytics application to be flexible to interact with.

1 2 3 4 5 6 7
(Disagree) (Agree)

I would find it easy to the Grading Analytics application to do what I want it to do.

1 2 3 4 5 6 7
(Disagree) (Agree)

Learning to operate the Grading Analytics application would be easy for me.

1 2 3 4 5 6 7
(Disagree) (Agree)

It would be easy for me to become skilled at using the Grading Analytics application.

1 2 3 4 5 6 7
(Disagree) (Agree)

I would find the Grading Analytics application easy to use.

1 2 3 4 5 6 7
(Disagree) (Agree)

9.3. Appendix 3 – User interface of the Grading Analytics application

Grading Analytics

Introduction

Import Data

Data Preparation

Curriculum Structure

Visualization

Metrics

Dataset

Modeling

Introduction to Grading Analytics

The purpose of the Grading Analytics framework is to analyze the data stored in the Neptun system and uncover the relationships between courses, thereby aiding data-driven decision-making. With the help of the website, all steps from the initial analysis to the final model building can be performed.

Grading Analytics

Introduction

Import Data

Data Preparation

Curriculum Structure

Visualization

Metrics

Dataset

Modeling

Import Data

Upload zip or excel file:

Program	NeptunID	Nem	Evfolyam	Tantargy	Jegy	Tipus	Datum	Felev
1 alkalmazott közgazdaságtan - 4BNAKGT17MBP	bb333fae4b367697eb684c8e3c7858807d464515	Férfi	2017/18/1	Matematikai analízis	Közepes	Vizsgajegy	2017-12-20T12:32:53Z	2017/18/1
2 alkalmazott közgazdaságtan - 4BNAKGT17MBP	bb333fae4b367697eb684c8e3c7858807d464515	Férfi	2017/18/1	Vállalatgazdaságtan	Közepes	Vizsgajegy	2017-12-13T15:55:16Z	2017/18/1
3 alkalmazott közgazdaságtan - 4BNAKGT17MBP	d17825821351b4f32148c4df3ab1d85d917233	Férfi	2017/18/1	Matematikai analízis	Közepes	Vizsgajegy	2017-12-20T12:32:53Z	2017/18/1
4 alkalmazott közgazdaságtan - 4BNAKGT17MBP	d969ae0d968b1c940da52fe5b75b3cc2c368cbe4	Nő	2017/18/1	Esszéírás, kutatómódszertan	Jó	Gyakorlati jegy	2017-12-15T12:31:38Z	2017/18/1
5 alkalmazott közgazdaságtan - 4BNAKGT17MBP	670bb66fa14cc34b599e7a24380c5e6a5993a5fe	Nő	2017/18/1	Informatika I.	Jó	Vizsgajegy	2017-12-15T22:57:45Z	2017/18/1
6 alkalmazott közgazdaságtan - 4BNAKGT17MBP	6582d525646a3aa4d82b5f0ad9a7bd94207fe	Férfi	2017/18/1	Esszéírás, kutatómódszertan	Átlárva	Átlírás	2017-12-10T00:00:00Z	2017/18/1
7 alkalmazott közgazdaságtan - 4BNAKGT17MBP	2a2f98a057dbc700cb8c8b725df798ab924249c	Nő	2017/18/1	Esszéírás, kutatómódszertan	Átlárva	Átlírás	2017-12-15T00:00:00Z	2017/18/1

Showing 1 to 7 of 3,938 entries

Previous 1 2 3 4 5 ... 563 Next

Grading Analytics

Introduction

Import Data

Data Preparation

Curriculum Structure

Visualization

Metrics

Dataset

Modeling

Data Preparation

Select a Program

alkalmazott közgazdaságtan - 4BNAKGT17MBP

Select Years

☒ 2017/18/1
 ☒ 2018/19/1

Select Subjects

☐ Alapszigorlat (7)
 ☐ Alkalmazott szigorlat (8)
 ☒ Esszéírás, kutatómódszertan (165)
 ☐ Fejezetek a matematikából II. (5)
 ☐ Fejezetek a matematikából III. (5)
 ☐ Fejezetek a matematikából IV. (IMP-SPM) (5)
 ☐ Fejezetek matematikából I. (7)
 ☒ Informatika I. (167)

Evfolyam	NeptunID	Nem	Tantargy	Tipus	Jegy	Vizsgafelv	Targyfelv
2018/19/1	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Esszéírás, kutatómódszertan	Vizsgajegy	4	1	1
2018/19/1	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Informatika I.	Vizsgajegy	4	1	1
2018/19/1	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Makroökonómia	Gyakorlati jegy	5	1	1
2018/19/1	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Matematikai analízis	Vizsgajegy	5	1	1
2018/19/1	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Nemzetközi gazdaságtan	Gyakorlati jegy	5	1	1
2018/19/1	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Pénzügytan	Vizsgajegy	4	2	1
2018/19/1	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Statistika	Vizsgajegy	3	1	1
2018/19/1	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Statistika II.	Vizsgajegy	4	2	1
2018/19/1	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Számvetel alapjai	Vizsgajegy	5	1	1
2018/19/1	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Valószínűségszámítás	Vizsgajegy	5	1	1
2018/19/1	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Vállalatgazdaságtan	Vizsgajegy	5	1	1
2018/19/1	013b3b601e4f8cc04b1c97467f5cbad9b0027548	Férfi	Ökonometria I.	Vizsgajegy	5	1	1

