Co-funded by the European Union

Optimizing Distance Learning Educational Programs

Erasmus+ Programme Cooperation Partnership Key Action 2

Project reference No: 2022-1-EL01-KA220-HED-000089152

The ODLEP Platform

Deliverable Form	
Project Reference No.	2022-1-EL01-KA220-HED-000089152
Document Title	Deliverable 3.2: The ODLEP Platform
Relevant Workpackage:	WP3
Nature:	Report
Dissemination Level:	Public
Document version:	2.0

Date:	30/06/2024
Authors:	Authors per Participating Institute (as appearing in the agreement Annex) **ARISTOTLE UNIVERSITY OF THESSALONIKI (EL)** Dr. Georgios Aretoulis Dr. Dimitra Manou PhD Candidate Nikolaos Tegos **UNIVERSITY OF MACEDONIA (EL)** Dr. Jason Papathanasiou Dr. Georgios Tsaples PhD Candidate Giota Digkoglou **UNIVERSITATEA LUCIAN BLAGA DIN SIBIU (RO)** Dr. Dan Miricescu Dr. Claudiu Kifor **UNIVERSITA TELEMATICA DEGLI STUDI IUL (IT)** Dr. Stefano Armenia Dr. Thomas De Angelis Dr. Lara Balleri Dr. Immacolata Messuri Dr. Giorgio Cecchi **UNIVERZITET U NISU (RS)** Dr. Jelena Stankovic Dr. Sandra Milanovic Dr. Ivana Veselinovic Marjanovic Dr. Vesna Jankovic Milic **UNIVERSITY OF IOANNINA (EL)** Dr. Eleni Aretouli Dr. Eleni Konstantinopoulou
Document description:	This deliverable presents the development of a lightweight, functional Learning Management System (LMS) designed to facilitate onboarding, content delivery, and group formation within the ODLEP project. The system enables students and professors to participate in a tailored educational process that includes survey-based group matching and structured access to course materials.

Table of Contents

List of Figures

Executive summary

The current deliverable presents the development and implementation of the ODLEP Learning Management System (LMS), which integrates a student-instructor matching algorithm with a lightweight platform for accessing asynchronous learning materials. The LMS is designed to support content delivery, and group matching in a flexible and accessible manner, using widely adopted tools such as Google Colab, Google Forms, Google Drive, and Google Sites.

The platform consists of two main components. The first is the ODLEP matching module that contains the core ODLEP algorithm and assigns students to learning groups and instructors based on their responses to structured surveys. The second is a content repository that provides each group with specific educational materials through restricted-access cloud storage.

To ensure wide compatibility and low implementation costs, the system was built using existing tools familiar to most academic users. The use of Google Colab for the matching algorithm ensures that instructors can run the code directly from a web browser without any software installation. Similarly, the use of Google Forms for survey distribution and Google Drive for content storage allows for seamless integration into existing academic workflows.

The platform is fully operational, adaptable to other courses and institutions, and complies with data protection requirements. The ODLEP LMS can be scaled, updated, and reused, supporting both current project needs and future applications.

Introduction

The objective of WP3 is to develop a Learning Management System with matching capabilities. It will consist of 2 modules. The first one will incorporate the mathematical framework that was will be developed and is responsible for the matching of students and/or educators. As a result, the module will allow the input of data by users (to identify their personality traits) and next will match them to an online classroom with students of similar characteristics and finally with an appropriate educator.

The second module will focus on providing students with material that can be accessed asynchronously. The ODLEP platform will be developed in such a way that it can be used with the existing infrastructure of the universities.

Beyond this compatibility requirement, several additional constraints were considered during the design phase. First and foremost, the ODLEP matching algorithm lies at the core of the system and must be usable by individuals with no background in programming or Operational Research. For this reason, the tool had to be intuitive and accessible without requiring the installation of complex software or the development of a standalone platform — which would fall outside the scope and resource limitations of the project.

Furthermore, the ODLEP matching algorithm relies on the answering of two separate surveys, one for students and one for professors/instructors. To support this workflow, a lightweight and universally accessible method for survey distribution and data collection was necessary. The selected tools had to be easily adoptable by users across various European institutions.

Finally, the choice of the platform for the development of the Learning Management System (LMS) had to be lightweight in order for its development to be feasible in the timeline of the project, adaptable so that it can be expanded with further students, departments and universities and sustainable to host educational content with minimal long-term cost and high availability.

These constraints were identified by the partners during internal meetings and drove the development of the LMS. The design choices and overview of this system will be described in the next section, while its technical implementation with step-by-step instructions will be the focus of the section after that. Conclusions, including limitations and future updates to the ODLEP LMS will be discussed in the final section of the present document.

Co-funded by the European Union

Optimizing Distance Learning Educational Programs

System Description

Addressing the design constraints

This section describes the technical components of the ODLEP LMS platform and how each design constraint was addressed through strategic choices in technology.

Firstly, to address the constraints of the ODLEP matching algorithm the tool of Google Colab was chosen. Its advantages are that it requires no installation, while any code can be run in a browser free-of-use. As a result, non-experts can execute the matching algorithm by following the simple instructions that are embedded within the overall flow of the code as a markdown text.

Moreover, Google Forms were used for the distribution of the surveys and collection of the responses from both students and instructors. It is a widely familiar tool, with a simple interface that connects automatically to Google Sheets and allows the collection of the results. Finally, it can support mass participation across different universities with minimal friction.

Finally, Google Drive was used from the beginning of the project to host all the material that were developed in the ODLEP project. Hence, it came as a natural continuation to add a section where all the material of the ODLEP LMS platform would be gathered, ensuring that all safety and privacy concerns are addressed. Moreover, Google Drive is compatible with both colab and Forms and allows folder-based access control with the use of either a gmail or institutional account.

Core components of the ODLEP LMS

The process that is at the core of the ODLEP project also dictated the design of the core components of the ODLEP LMS.

Firstly, there is the input collection, where students and instructors are directed to the relevant Google Forms to answer the respective surveys. The instructor overseeing the course either asks for the results of the survey for their students from the LMS's administrator (Georgios Aretoulis, PM of the ODLEP project) or he notifies him to use them for the next part of the process which is the matching.

A Google colab notebook contains the code with the matching algorithm and the instructor can either request from the LMS administrator to execute the process or can request access to the notebook and perform the task themselves. The output of the matching algorithm is a grouping of the students and their matching to the most compatible instructor.

Finally, each group has a dedicated space on Google Drive that they can access through their email accounts that contains course-specific presentations or resources. The access to the drive folder is provided by the LMS administrator after the groups' formation to ensure privacy and safety. Professors may upload any material at any time during their use of the platform.

Consequently, the addressing of the constraints and the workflow of the ODLEP algorithm revealed Google sites as the central platform that would combine all components in a user-friendly interface. It has a seamless integration with colab, Forms and Drive thus, any friction from the use of different tools is minimized. In addition, the platform supports updates and expansion with no additional infrastructure, aligning with the university constraints and the project's lightweight development goals.

Technical Implementation

The technical implementation of the ODLEP LMS platform followed the logic described in the previous sections, combining simplicity, accessibility, and modularity. The aim was to create a system that can be easily used and maintained, even by individuals with no background in programming or LMS development.

The link to the ODLEP platform is the following: https://sites.google.com/view/odleplms/home

The user that enters the platform is greeted and informed about the aim of the project and its reference number (Figure 1).

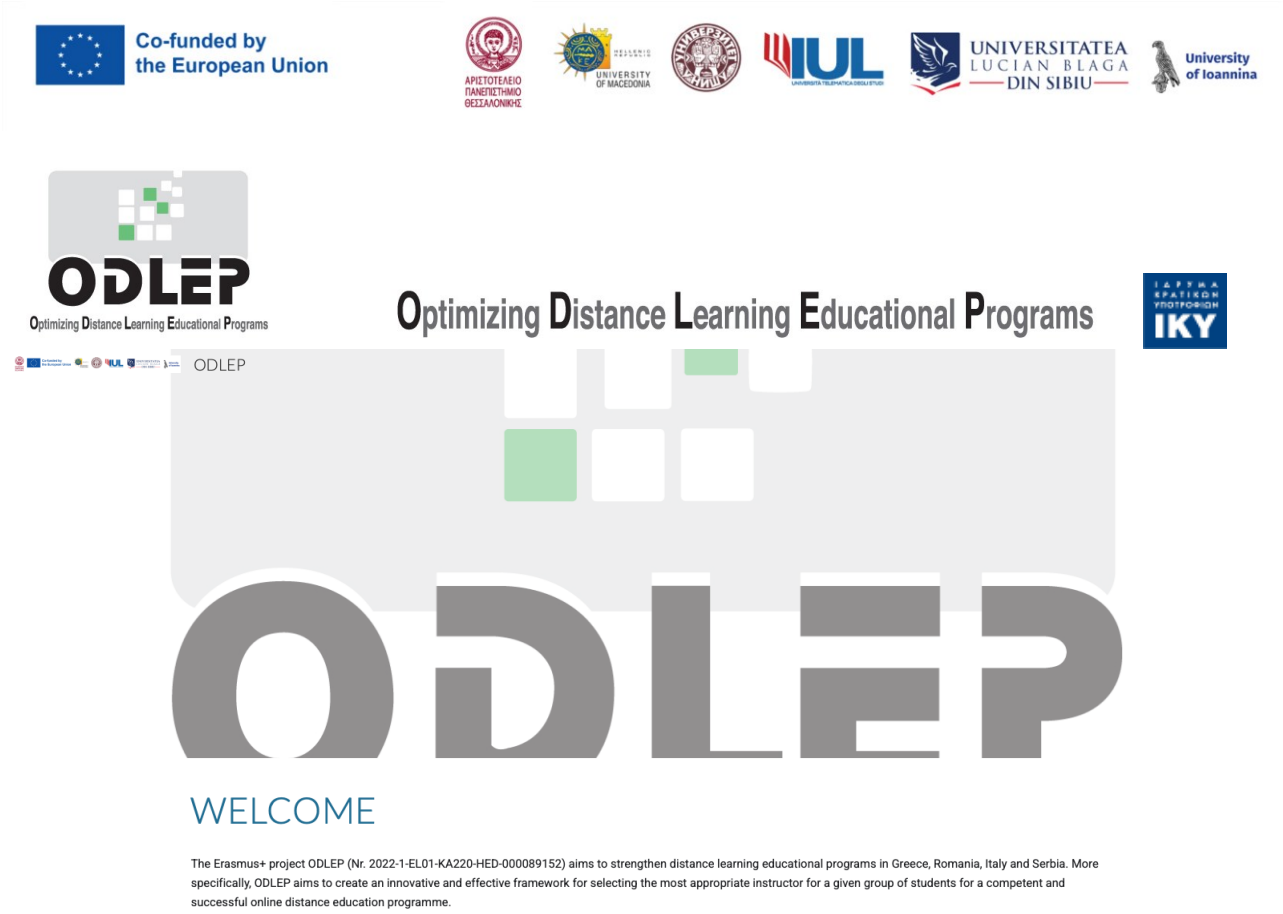


Figure 1 Landing section of the ODLEP LMS platform

The two surveys (student and instructor) are implemented through Google Forms and can be found in these links:

- Students' survey
- Instructors' survey

Each form includes a number of questions selected by the partners during the early design phase of the project. These questions aim to capture key characteristics relevant to the matching algorithm, such as learning preferences, availability, or instructional approach. The responses are automatically stored in Google Sheets, which are linked directly to the Colab environment, ensuring a smooth flow of information from input to processing.

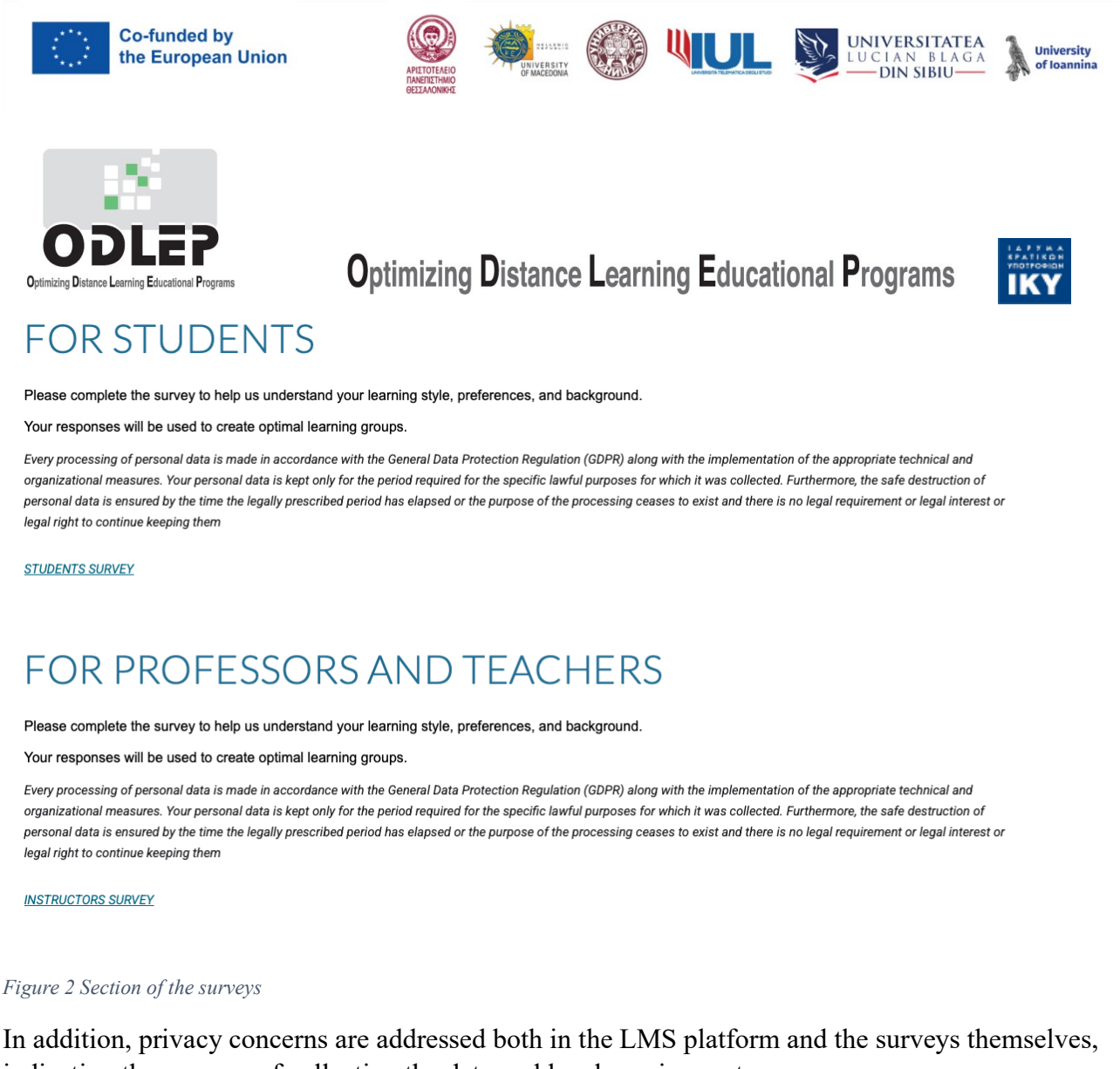


Figure 2 Section of the surveys

In addition, privacy concerns are addressed both in the LMS platform and the surveys themselves, indicating the purpose of collecting the data and legal requirements.

The core element of the implementation is the Google Colab notebook, which includes the matching algorithm. The notebook is publicly accessible and can be executed through any browser without the need for software installation.

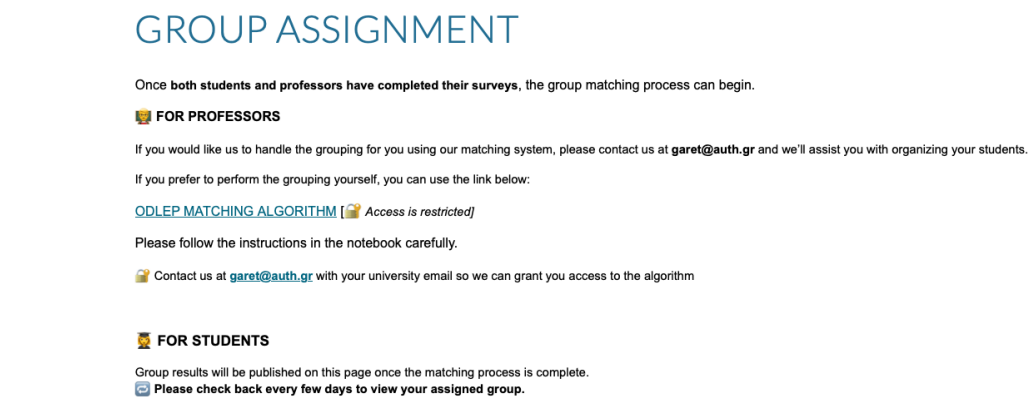


Figure 3 The ODLEP Algorithm section of the LMS

The structure of the notebook follows a simple logic with markdown cells that provide step-by-step instructions for the user, and code cells that perform the necessary tasks. The required input is the Excel file with the answers from the student and instructor surveys, which can be downloaded directly from the connected Google Sheets. The user is also asked to define a small number of parameters such as the number of instructors or the desired number of groups. The output is an Excel file with the final grouping, which can be uploaded back to the LMS or distributed to the instructors directly.

More specifically, if the instructor overseeing the code wishes to run the algorithm themselves, they can follow the instructions that are in the Appendix of the current document.

The final step of the process is the upload and management of the learning material. After the groups have been formed, the LMS administrator creates individual Google Drive folders for each group and gives access to the corresponding students and instructors via their institutional or Gmail accounts. These folders contain the presentations and additional resources related to the course. The platform also allows professors to request the addition of new materials at any time, which are then uploaded by the administrator to the appropriate folder.

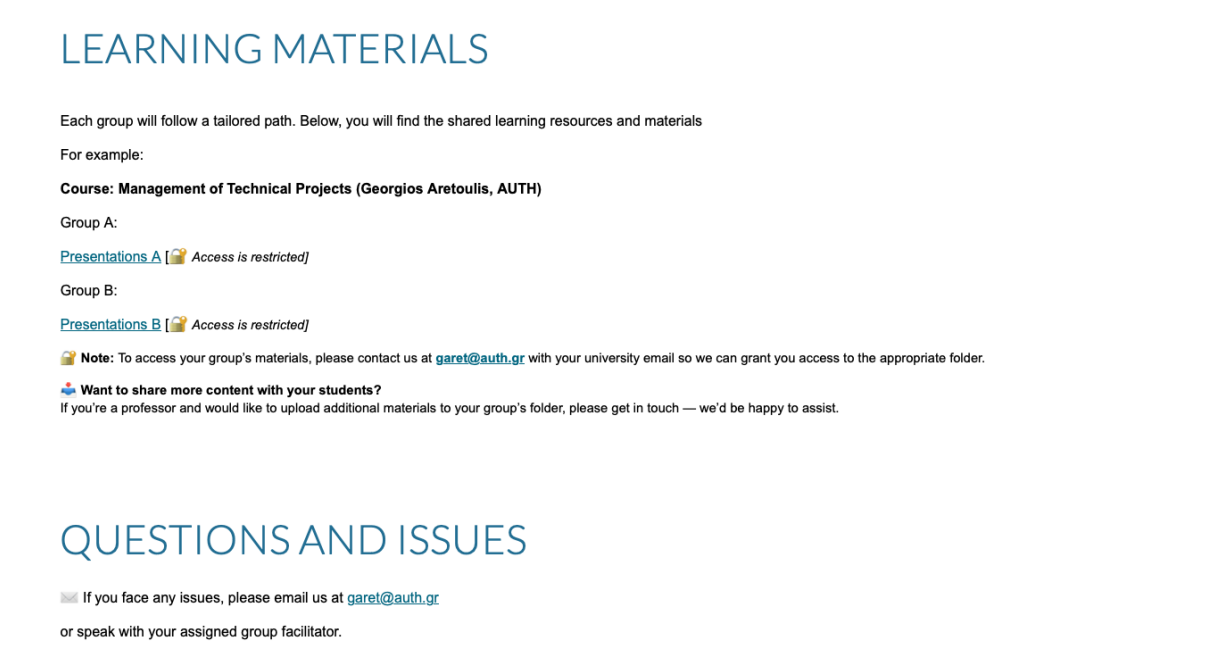


Figure 4 Section of the LMS where each group can access their respective material

All three components—Colab (algorithm), Forms (survey), and Drive (content)—are integrated into a single interface using Google Sites. The Google Site serves as the entry point for students and instructors and guides them through the process via a step-by-step structure. The published version of the site contains instructions, survey links, access to the matching algorithm, and links to the learning material (restricted by group). The administrator has the ability to update the site at any time and expand it to include more courses, students, or departments, without requiring any new infrastructure or hosting.

This lightweight technical implementation ensured that the platform could be fully operational within the timeline of the project while remaining compatible with the infrastructure and digital practices of European universities.

Conclusions

The development of the ODLEP LMS platform addressed the objectives and constraints identified by the project partners during the activities on WP3. A lightweight, modular, and fully operational system was created, allowing for survey distribution, student and instructor matching, and asynchronous access to learning materials. The choice of using widely adopted tools such as Google Colab, Forms, Drive, and Sites ensured compatibility with existing university infrastructures, minimized costs, and maximized ease of use for non-expert users.

At the same time, the modularity of the platform allows for easy updates and adaptations. The surveys can be be expanded or constracted or even new ones can be created for different courses, the matching algorithm can be applied with minor adjustments, and additional learning materials can be uploaded without any technical barriers. Moreover, the clear separation of responsibilities, with an LMS administrator handling access control and system updates, contributes to the long-term sustainability of the platform.

Despite its successful implementation, the ODLEP LMS also presents some limitations. Firstly, the platform relies on manual steps at certain points, particularly in the management of access rights to Drive folders and in the execution of the matching algorithm for users unfamiliar with Colab environments. Secondly, while the use of Google services ensures accessibility, it also implies a dependency on external platforms whose policies or availability might change over time. Finally, the current version of the ODLEP LMS does not include features such as automated notifications to students or dynamic updating of group assignments without administrator intervention, which could limit its functionality for very large-scale deployments.

Future updates to the ODLEP LMS could focus on addressing these limitations. Specifically, the automation of certain steps, such as granting access to Drive folders based on group assignments or sending automated notifications once grouping is completed, could further improve the user experience. Additionally, the Colab matching algorithm could be further developed into a simplified web interface, reducing the need for interaction with code cells or the colab environment overall. Finally, although Google Sites was selected for its simplicity and speed, future versions of the LMS could explore more specialized learning platforms if project needs and resources allow.

Overall, the ODLEP LMS platform serves as a robust foundation that meets the initial goals of the project while providing multiple pathways for future improvement and expansion.

Appendix

How to Use the Code

This section provides a step-by-step guide on how to use the provided Python code for matching students with teachers using K-Means clustering and the TOPSIS method. The code is hosted on Google Colab to ensure ease of use without requiring any local setup.

Step 1: Access the Notebook

To access the notebook, click on the following link: Open Notebook in Google Colab. This link will open the notebook directly in the Google Colab environment.

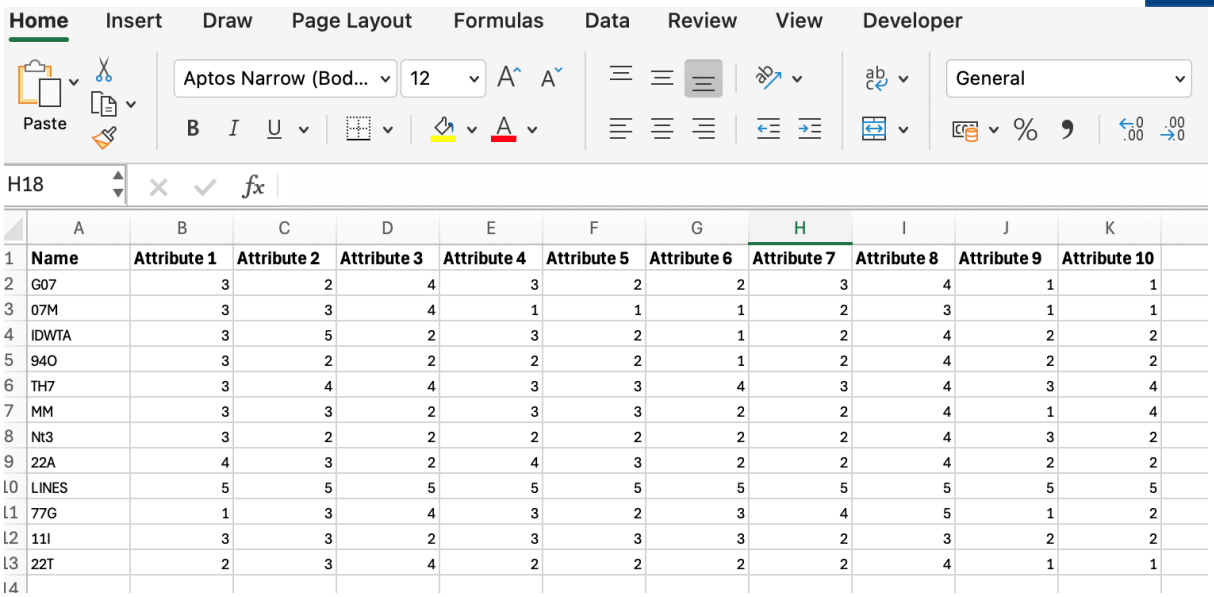
Step 2: Prepare Your Data

Ensure you have two Excel files prepared:

1. Student Attributes File:

- The first column should contain student IDs.

- The subsequent columns should contain various attributes of the students (e.g., scores, personality traits).



	A	B	C	D	E	F	G	H	I	J	K
1	**Name**	**Attribute 1**	**Attribute 2**	**Attribute 3**	**Attribute 4**	**Attribute 5**	**Attribute 6**	**Attribute 7**	**Attribute 8**	**Attribute 9**	**Attribute 10**
2	G07	3	2	4	3	2	2	3	4	1	1
3	07M	3	3	4	1	1	1	2	3	1	1
4	IDWTA	3	5	2	3	2	1	2	4	2	2
5	94O	3	2	2	2	2	1	2	4	2	2
6	TH7	3	4	4	3	3	4	3	4	3	4
7	MM	3	3	2	3	3	2	2	4	1	4
8	Nt3	3	2	2	2	2	2	2	4	3	2
9	22A	4	3	2	4	3	2	2	4	2	2
10	LINES	5	5	5	5	5	5	5	5	5	5
11	77G	1	3	4	3	2	3	4	5	1	2
12	11I	3	3	2	3	3	3	2	3	2	2
13	22T	2	3	4	2	2	2	2	4	1	1
14											

2. Teacher Attributes File:

- The first column should contain teacher IDs.

- The subsequent columns should contain various attributes of the teachers (e.g., preferences, teaching styles).

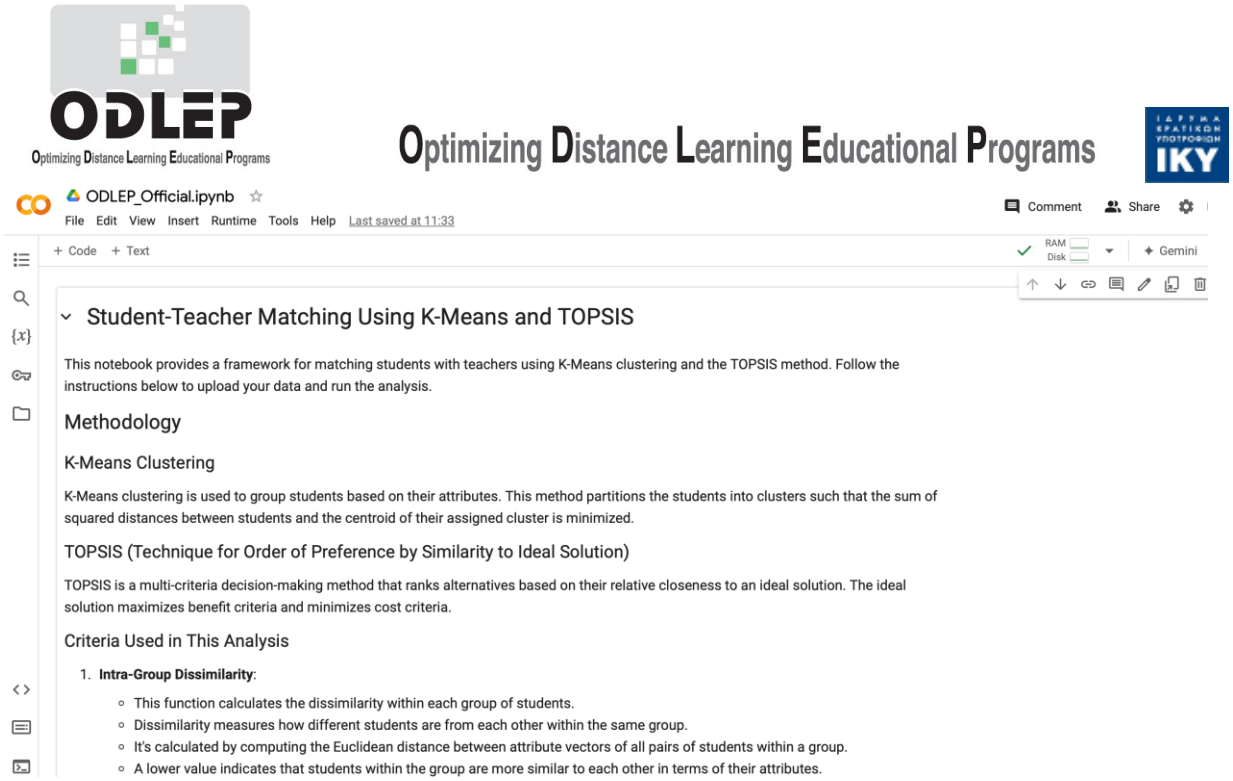
Both files should be in Excel format (.xlsx).

Step 3: Make a Copy of the Notebook

Before running any code, please make a copy of the notebook to your own Google Drive to ensure you do not modify the original version.

1. Open the Notebook in Google Colab:

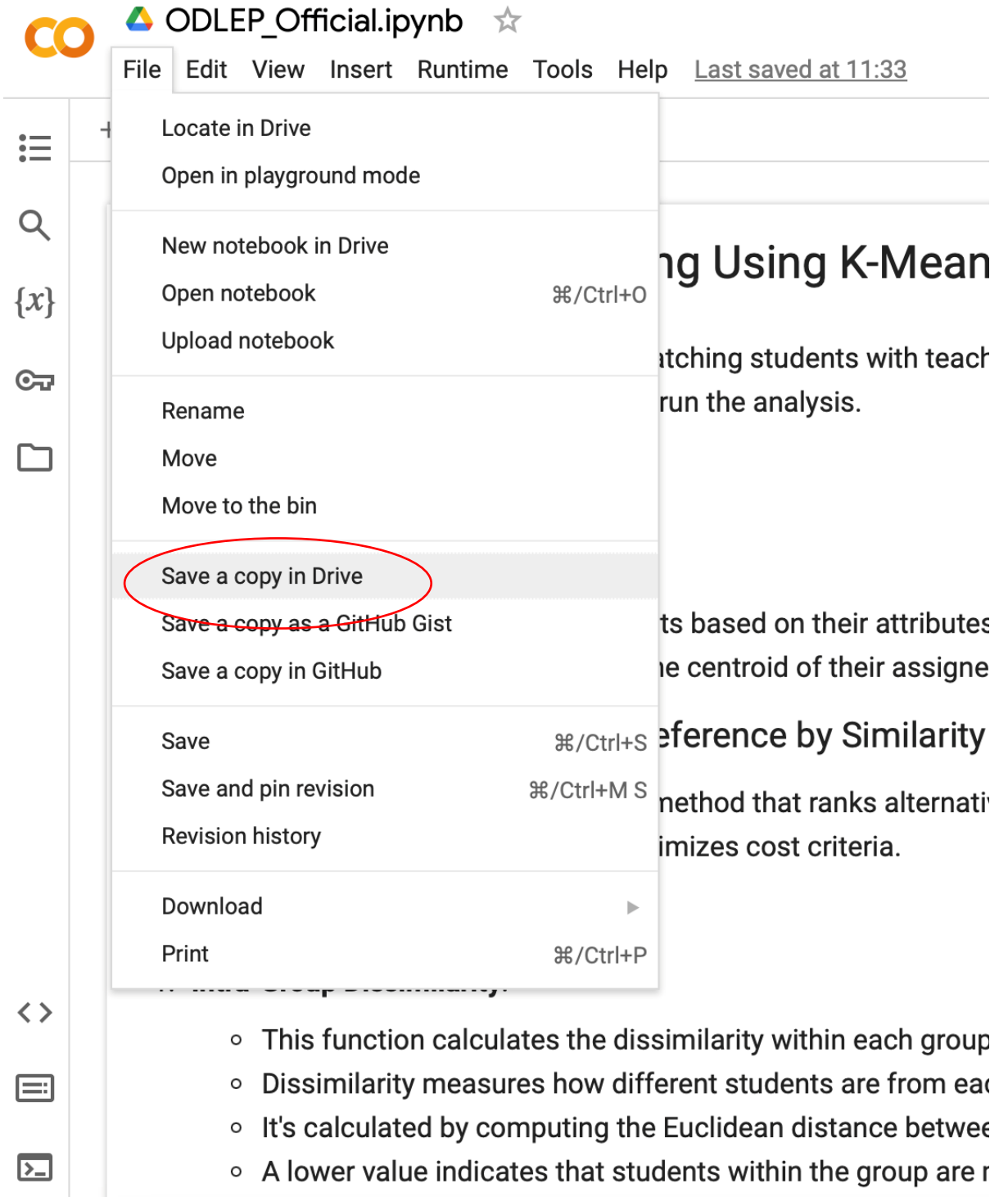
- Click on the provided link.



2. Save a Copy to Your Drive:

- Once the notebook is opened in Google Colab, go to the menu bar and click on `File`.

- Select `Save a copy in Drive`.

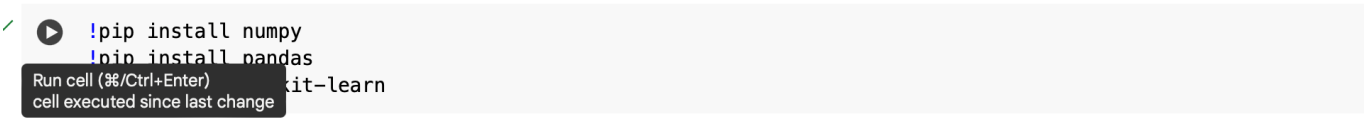


4. Upload excel files

Please upload the two excel files in the same folder as the file containing your code

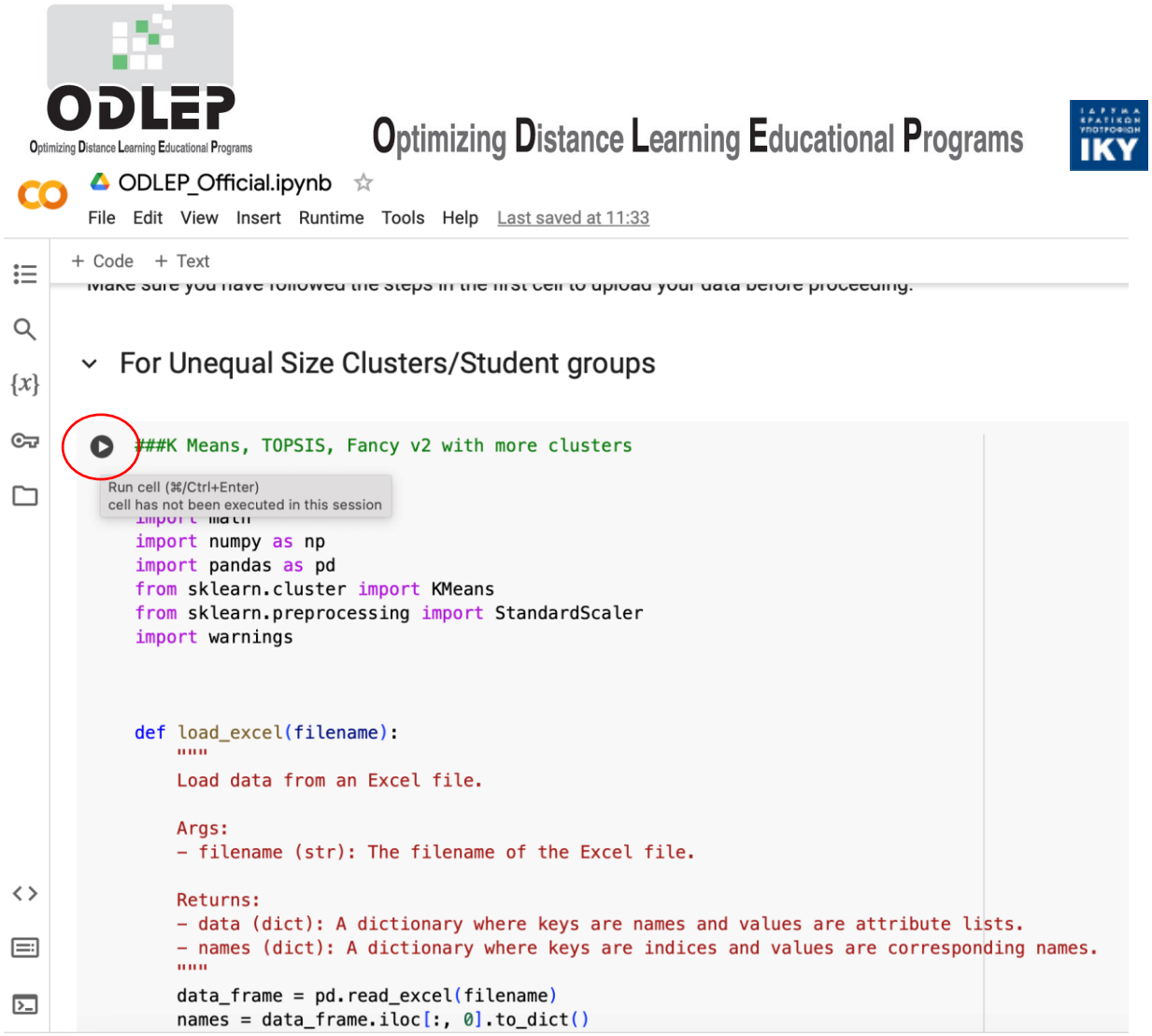
5. Instructions for running the file properly

If you have never used Google Colab or python in your laptop, please press Play on the next cell



6. Run the Copied Notebook:

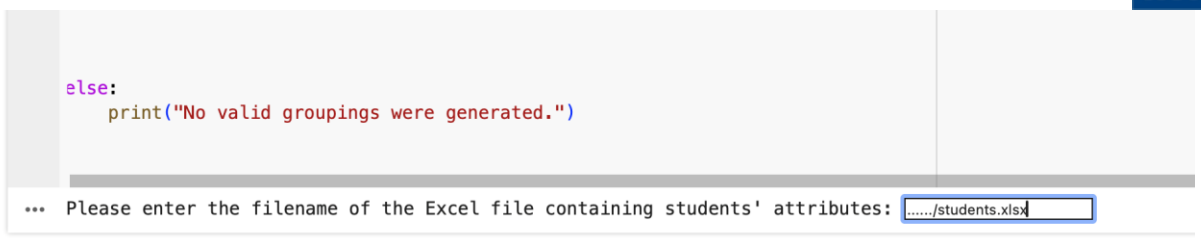
- The copied notebook will open in a new tab. You can now run the code cells in this copy without affecting the original notebook.



Step 7: Upload Your Data

1. Open the copied notebook in Google Colab.

2. Follow the prompts in the notebook to upload your Excel files. There will be sections where you need to upload the student and teacher attribute files.



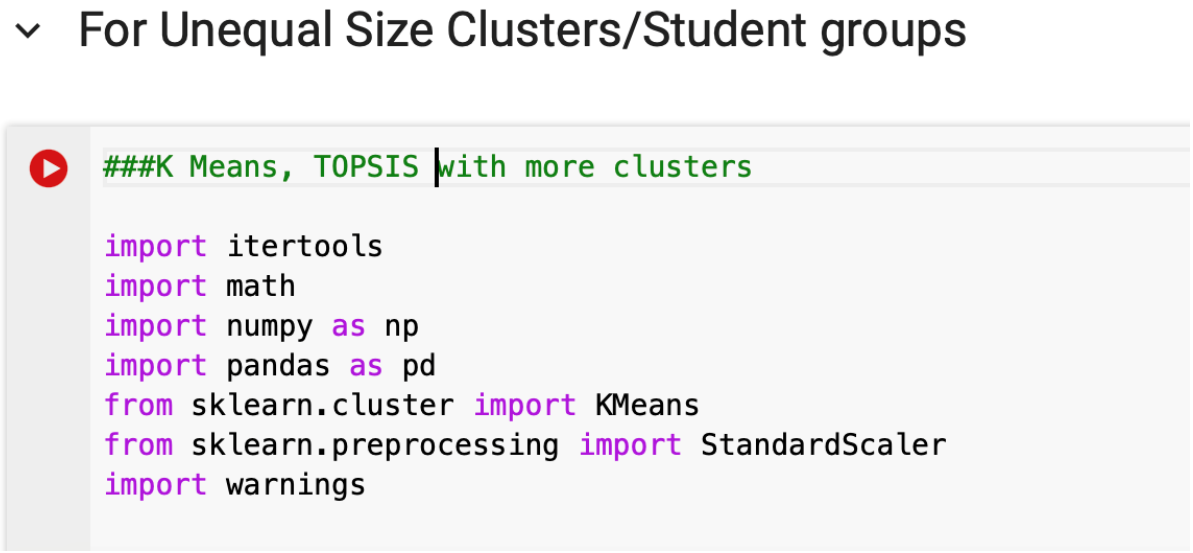
Step 8: Choose Clustering Option

The notebook provides two clustering options: unequal size clusters and equal size clusters. Follow the instructions in the notebook to choose your desired option.

1. For Unequal Size Clusters:

- Press the "Run" button on the cell labeled for unequal size clusters.

- Follow any additional prompts to input data or weights.



2. For Equal Size Clusters:

- Skip the cell for unequal size clusters.

- Press the "Run" button on the cell labeled for equal size clusters.

- Follow any additional prompts to input data or weights.

For equal Size Clusters/Student groups

```
###K Means, TOPSIS with more clusters of equal size####



from sklearn.preprocessing import StandardScaler
import numpy as np
import pandas as pd
import itertools
import math
import warnings

def load_excel(filename):
```

Step 6: Put weights for the three criteria

The weights should equal 1. The input should be in the form: **0.34**

```
Please enter the filename of the Excel file containing students' attributes:  students.xlsx
Please enter the filename of the Excel file containing teachers' attributes:  teachers.xlsx
Please provide the weights for each criterion:
Weight for intra-group dissimilarity:  0.34
Weight for group-teacher dissimilarity:  0.33
Weight for intra-group diversity:  0.33
```

Step 9: Run the Analysis

After uploading your data and selecting the clustering option, run the remaining cells in the notebook sequentially. This will execute the K-Means clustering and TOPSIS analysis to generate the optimal student-teacher groupings.

Step 10: View the Results

The results of the analysis, including the best three groupings for the selected clustering option, will be displayed at the end of the notebook. The TOPSIS scores for each grouping will help you identify the most effective student-teacher matches.

```
The best three groupings are:
Alternative 1 (TOPSIS Score: 0.7990675814003713):
Group 1: ['G07', '940', 'Nt3'] with Teacher 25A
Group 2: ['11I', 'LINES', '22A'] with Teacher 16R
Group 3: ['IDWTA', '07M', 'MM'] with Teacher T0000
Group 4: ['TH7', '22T', '77G'] with Teacher AP1

Alternative 2 (TOPSIS Score: 0.7781813440087904):
Group 1: ['11I', 'LINES', 'G07'] with Teacher 25A
Group 2: ['Nt3', '22A', '77G'] with Teacher AP1
Group 3: ['22T', '07M', 'IDWTA'] with Teacher T0000
Group 4: ['MM', 'TH7', '940'] with Teacher 16R

Alternative 3 (TOPSIS Score: 0.7666539281126158):
Group 1: ['11I', 'LINES', 'G07'] with Teacher 16R
Group 2: ['Nt3', '22A', '77G'] with Teacher AP1
Group 3: ['22T', '07M', 'IDWTA'] with Teacher T0000
Group 4: ['MM', 'TH7', '940'] with Teacher 25A
```

Additional Information

- The methodology used in this analysis combines K-Means clustering to group students based on their attributes and the TOPSIS method to rank the groupings based on criteria such as intra-group dissimilarity, group-teacher dissimilarity, and intra-group diversity.

- Detailed explanations of the criteria and the methods are provided within the notebook to ensure a clear understanding of the process.

By following these steps, you can effectively use the provided code to match students with teachers, optimizing educational outcomes based on the specified criteria.