
A+Guide

Faculty of Computers
and Artificial Intelligence

A^+ Guide

Ahmed Hany Hamdy

Hossam Khaled Fawzy

Mario Emad Naguib

Supervised By

Prof. Dr. Ihab El Khodary

Dr. Marwa Mostafa Sabry

CAIRO UNIVERSITY

T 2023

ABSTRACT

Every year, the sophomore student in FCAI faces the decision of choosing which major to specialize in. For some students it is not a big deal, while for others this is a great problem. Selecting the appropriate department is considered the first biggest decision the student makes. That why it is hard to decide. Many students take into consideration the wrong aspects while choosing, hence they suffer the rest of their academic years in college.

As a response, there must be another option in which a system analyzes the collage data and provide the managers with what is going on in the courses and how students are performing. There must also be a method to give recommendations to students on which department is better for them to join according to their grades.

Data analysis tools will be used along this project and also some machine learning techniques such as clustering. This document will provide further information about how these methods will be applied regarding this specific case. Then comes the project implementation.

DECLARATION

We hereby declare that our dissertation is entirely our work and genuine / original. We understand that in case of discovery of any PLAGIARISM at any stage, our group will be assigned an F (FAIL) grade and it may result in withdrawal of our bachelor's degree.

Group members:

Name	**Signature**
Ahmed Hany Hamdy	_____
Hossam Khaled Fawzy	_____
Mario Emad Naguib	_____

PLAIGRISM CERTIFICATE

This is to certify that the project entitled "**Analyzing performance for students' grades and predicting their suitable departments**," which is being submitted here with for the award of the "**Bachelor of Computer and Artificial Intelligence Degree" in "Operations Research and Decision Support** ". This is the result of the original work by **Ahmed Hany Hamdy, Hossam Khaled Fawzy, and Mario Emad Naguib** under my supervision and guidance. The work embodied in this project has not been done earlier for the basis of award of any degree or compatible certificate or similar title of this for any other diploma/examining body or university to the best of my knowledge and belief.

Turnitin Originality Report

Processed on 31-May-2017 00:14 PKT

ID: 300502964

Word Count: 12948

Similarity Index

10%

Similarity by Source

Internet Sources:	06%
Publications:	0 %
Student Papers:	08%

Date: 30/05/2017 Prof. Dr. Ihab El Khodary & Dr. Marwa Sabry

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF Tables

CHAPTER 1

Project Introduction and Objectives:

Introducing the Problem and Goals.

Introduction

Problem Statement

Due to the rising numbers of students joining FCAI. The faculty is facing a problem of how to accommodate these students evenly among all majors. Students choose their departments haphazardly without having enough information about what they are choosing whether or not they have the skills that are needed for joining this department, or even know what courses are being taught in that department.

The pie chart below shows a survey that was done on students in FCAI today, and the question was it easy to choose your major:

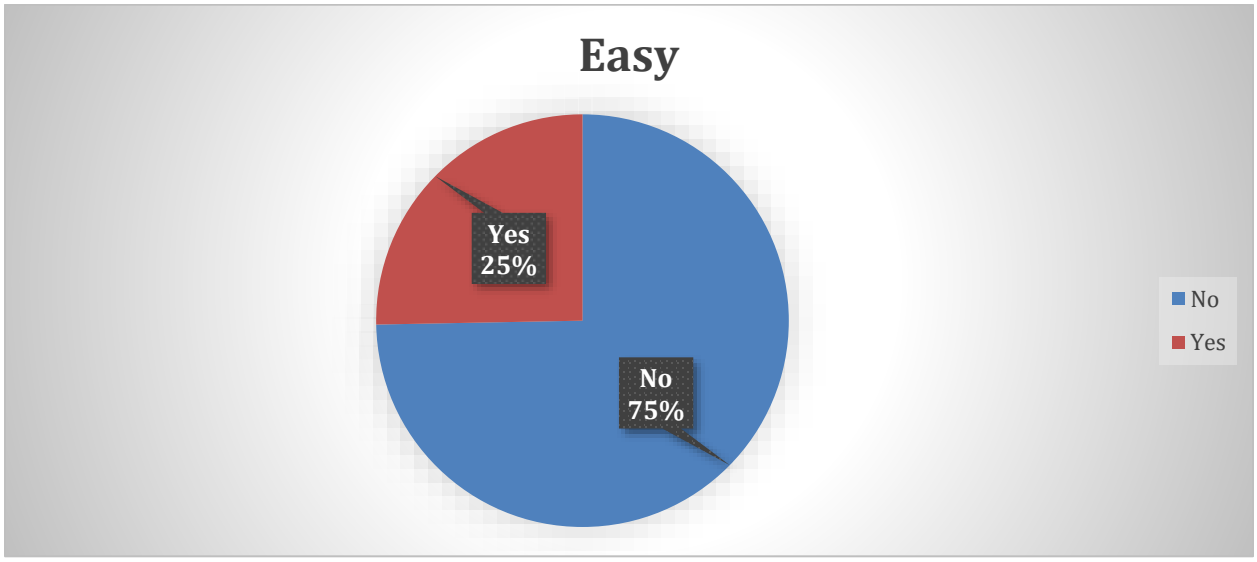

Fig 1

According to the collage, number of students that join the collage every year is increasing. With that the number of students that will face a difficulty in choosing their major will only increase.

With the proposed idea the students will have the proper guidance on what to do based on their skills that was evaluated in their first two years in collage through final exam of the courses.

Objectives

This project objectives are relieving the students from the burden of choosing the department on their own without any guidance, helping them identifying their point of strengths and which field they are capable of achieving higher grades in.

We wanted to offer the collage a system that:

- Assess each student performance.
- Provide recommendations to all students.
- Avoid managerial requests of students wanting to change their department after the junior year.
- Avoid dropouts from certain courses.

Also offer the collage a series of analysis to their data using dashboards, to provide them with recommendations on how to manage certain issues and help them make data-driven decisions.

Motivation

Our Thought During Idea Searching Phase we thought about a project that meets our jobs requirements and meets our team's intended learning objective through the graduation project and we chose an idea which assembles different skills in our team. In addition, this project will help in solving academic problems for future students and help them achieve their dreams.

Project description (The proposed model)

Data in its raw form is ambiguous and misleading. But the power of statistics come from using analyzing this data using mathematical techniques, and from these techniques that we are able to comprehend this data and take decisions upon it.

We intend to use excel to analyze the data, through applying some data preparation methods such as cleaning, labeling, validating etc.…

We are also going to introduce some dashboards in which the data will be much less complicated and can be interpreted easily. By doing that we make sure that taking a decision based on this clean, easy to understand data will be a sound decision.

Once we are finished with this step, then comes the second step.

In this step we are going to use a machine learning method (clustering), to try and find similarities in the given data and try to produce some relations and correlations between the data.

At the end, we will have a model that is able to cluster the data and give us insights about the relations. That model will also be able to make a customized proper recommendation, based on statistical calculations, on which major will be suitable for each student.

CHAPTER 2

Data Cleaning and Transformation:

Preparing the Data for Analysis.

Introduction

After identifying the problem that we are facing and knowing the nature of the data we may need in our model.

We began to search for ways to obtain this data, and we found that we could obtain it from the affairs of the students of the Faculty of Computing and Artificial Intelligence at Cairo University, after the approval of Dean Prof. Dr. Reda El-Khoriby and Prof. Dr. Ehab El-Khodary.

After submitting the required papers, we were able to obtain this data, but after modifications occurred by the college administration, these modifications did not affect the quality of the data that we identified before. Thus, we have reached the first stage in preparing the data.

Data preparation

Data Collecting

The data came to us in some way encrypted, so we resorted to using Excel, as we explained before, to help us understand the data and determine the features we need.

We were able to summarize this data into Three parts:

1. Grades of students who have previously joined the College of Computing and Artificial Intelligence and who fell below the GPA system in all courses.
2. Their GPAs up to level two
3. And the departments in which they have enrolled.

Data cleaning

In this phase, data cleaning usually begins by trying to discover errors and try to correct them, if possible, as patterns and links that link the data together are discovered.

In general, the data is simplified in order to be more readable and understandable to start analyzing it.

We were able to identify the redundant parts of the data that we will not need and exclude them from the files.

We were able to identify what was lacking in the data and try to reformulate it using Excel, such as that the data was lacking in the departments that each student enrolled in, so we began to define the compulsory courses in each department that all students enrolled in this department must register, then we linked the students

who had previously registered these courses, so we got students of each section separately.

The data also were missing the GPAs up to the second level, where the data contained the GPAs up to the current level that the students are.

So, we calculated the points achieved by each student in both the first and second levels, and calculated the number of hours that he registered, and thus we were able to calculate the GPA of each student until the second level only.

Data labeling

In the data labeling phase, we started gaining more information from the data and identifying the relations among it. We have also given it labels that are more informative and meaningful. We also determined the actual data we need and separated it.

Data validation

After data has been cleaned enough and labeled. It's time to make sure that it is correct and ready to use it in our model, by checking data different types, applying format check, consistency check, it is a type of logical check that confirms the data's been entered in a logically consistent way, and uniqueness check, it is to determine whether combinations of variables (usually key variables) uniquely identify a record e.g. the IDs of each student can't be repeated. In general, we were ensuring that issues caused due to the decay of data have been resolved.

Data visualization

In this phase, we have applied some visuals on the data to measure the range of some features.

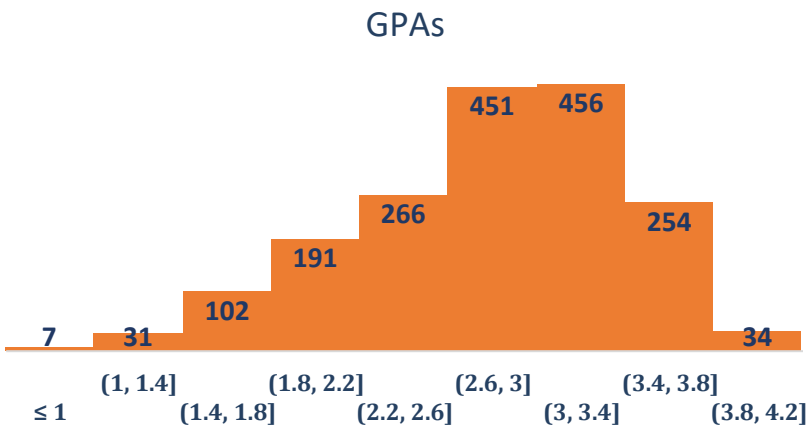

Fig 2

As shown in this histogram chart, we find that the average GPA up to the second level is 2.7.

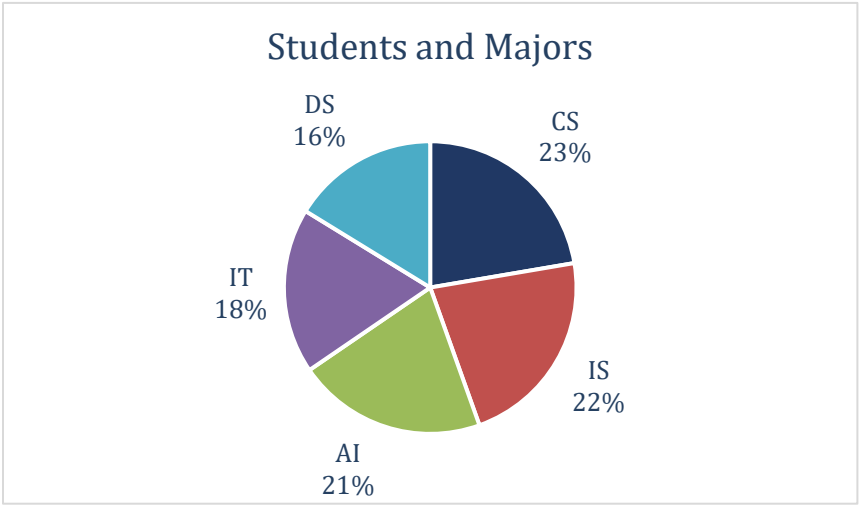

Fig 2.1

The second pie-chart shows the percentage of students that enrolled in each department, which shows the most wanted majors, and least favored ones.

What's Next

Now, after the data has been cleaned enough, and the grades of prerequisite courses for each major and the GPAs till second level have been prepared.

We put students' grades of prerequisite courses in the form of vectors.

Which means that each student will appear as vector contains his grades in all departments' prerequisites as features.

Vectors Representation

Let us say we have 3 students A, B and C

A (89, 70, 76,……, 3, 2.98)

B (89, 70, 76,……, 2, 2.78)

C (70, 80, 56,…, 4, 2.38)

Each Vector contains a student's prerequisite grades, this means that we change student to vectors to can manage clustering with cluster model to be easy to use.

Each vector also contains the GPA and the department code (1, 2, 3,…)

Firstly, we want the cluster model to focus on the grades only , so we will give the GPA and the department code weight tends to zero .

Clustering Phase

As explained, we seek to group the vectors that are quite similar in features into clusters using clustering models e.g., K-means model.

First determine prerequisite courses and apply the following steps:

1. Comparing students' grades in prerequisite courses.
2. Then clustering the students that have similar grades in prerequisite course of each major.
3. Now we have divided data into clusters each one containing the students with similar grades.
4. Then determine the dominant major in each cluster.

Fig 3

5. When a new student wants to use this model, his grades will be formed in a new vector.

   N (69, 81, 60,…….,X, 2.4)

   The model will assign the new student to a cluster having similar grades to him.

Fig 3.1

Note: X in N's Vector→ refers to The Department that

The student will choose.

6. So, the dominant major will be the recommended to the new vector.

Our Aim

We are seeking to find if there is a relationship between grades of prerequisite courses and choosing the major,

Just the GPA IS ENOUGH

CHAPTER 3

Data Visualization and Dashboard Insights:

Visualizing Key Performance Metrics

Intro to dashboards:

After thoroughly preparing the data, we were able to capture the core of it and turn it into valuable insights. We set out to design aesthetically appealing and informative dashboards using this insightful data. By clearly and concisely presenting the data, these dashboards function as an overview of the information. Our dashboards uncover a wealth of knowledge with each click and interaction, enabling decision-makers to make knowledgeable decisions and take smart actions based on the full picture of the preprocessed data.

Excel is a powerful tool that can be used to make insightful dashboards that are visually appealing. Excel provides a wide range of chart formats, such as scatter plots, pie charts, line graphs, and bar graphs, allowing users to select the most appropriate visualization for their data. With the use of Excel dashboards, users can monitor, analyze, and make decisions based on a consolidated view of important data and KPIs. With its adaptable capabilities, Excel dashboards enable you to combine numerous data sources, like financial data, sales data, marketing analytics, and more, into a single interface.

An Excel dashboard's fundamental objective is to show complicated data succinctly and clearly so that users may easily recognize trends, patterns, and important insights. Once the data has been gathered and organized, users may utilize Excel's charting and graphing features to produce engaging visualizations. Excel provides a variety of chart formats, including scatter plots, line graphs, pie charts, and bar graphs, allowing users to select the visualization method that is

most suited for their data. Colors, labels, and formatting choices can be changed on these visualizations to improve clarity and efficiently convey information.

Excel dashboards have completely changed how data is visualized and examined. With its strong features and functions, Excel gives users the ability to build dynamic, interactive dashboards that offer real-time information and encourage thoughtful decision-making. A dashboard is a graphic depiction of data that compiles data from several sources into a single, user-friendly interface. Excel dashboards offer a great deal of flexibility and may be modified to satisfy certain needs and goals. They let customers keep track of their goals by providing a thorough overview of measurements, trends, and key performance indicators (KPIs).

Dashboards can benefit from a wide range of Excel features and functionalities that can improve their usability and interaction. Users can build interactive dashboards that allow filtering, sorting, and diving down into data for more in-depth analysis by incorporating dynamic formulas, conditional formatting, slicers, and pivot tables.

Clarity and usefulness are improved by an Excel dashboard's structure, formatting, and organization. Design decisions like font styles, color palettes, and grid alignments may have an enormous influence on how well the user understands the data. The information is also displayed logically and intuitively thanks to the dashboard's thoughtful layout of charts, tables, and text components.

Excel dashboards may successfully communicate insights and support data-driven decision-making by carefully addressing these design factors.

Users may efficiently share data-driven insights with a variety of stakeholders by using dashboards. Dashboards make it simpler for technical and non-technical consumers to comprehend complicated datasets by graphically engaging information. Users may instantly recognize crucial results and make educated decisions thanks to the clear and straightforward representation of data offered by visual components like charts and graphs. Additionally, dashboards include interactive exploration, which lets users engage with the visualizations to dive further into the data and get a more thorough knowledge.

Additionally, Excel dashboards allow for real-time or scheduled updates, guaranteeing that the information being shown is always up to date and pertinent. Tracking key performance indicators (KPIs) and tracking goal-related progress are two uses for this functionality.

In conclusion, Excel dashboards offer a thorough and convenient approach to see and analyze data. For companies, groups, and people who want to use the well-known and flexible Excel program to get insights, monitor performance, and make data-driven decisions more successfully, they provide a potent option.

Creating a dashboard can be done by following these six simple steps:

1. **Define the purpose and audience:**

In order to create a dashboard, we first set out to determine its goal and target audience. We started by carefully examining the project's needs and goals. The primary objectives and KPIs that needed to be addressed were determined after reviewing the project documents and having discussions with stakeholders. This made it easier for us to comprehend the unique requirements and demands for the dashboard. In order to comprehend the tastes and needs of potential consumers, we also solicited input from them. Project managers, team members, and senior stakeholders who needed real-time insights and a clear knowledge of the project's progress and critical KPIs were designated as the dashboard's target audience.

2. **Gather and organize data:**

We started by identifying the pertinent data sources in order to collect and arrange the data for a dashboard. To decide which data points were necessary for the dashboard, we examined the spreadsheets and datasets we had obtained from the collage. In order to make sure that all pertinent information was included, we collaborated with stakeholders. After locating the data sources, we concentrated on cleaning and confirming the data. This required eliminating any duplicates, fixing mistakes, and guaranteeing consistency across various datasets. Through this method, I was able to obtain a clear and trustworthy dataset that I could utilize to efficiently produce useful visualizations for the dashboard.

3. **Select appropriate visualizations:**

We thoroughly examined the statistics and considered the information I intended to convey. We determined the best chart types for each piece of data based on its characteristics, such as whether it was numerical or categorical. Depending on the precise analysis needed, I thought about using choices like bar charts or line charts for numerical data. Pie charts were used to illustrate percentages or comparisons for categorical data. I also looked about using tables or matrices to display tabular data. I made sure the selected visualizations successfully communicated the important ideas and made the data simple to grasp for the target audience. To make sure they improved the dashboard's overall user experience, I also focused on the visual attractiveness and clarity of the graphics.

4. **Design the layout:**

We began by considering the organization and general layout of the visualizations. Our goal was to design a straightforward interface that would be simple for consumers to utilize and understand the facts. The order of the material was considered, and the most significant and powerful visualizations were given prominent placement. To improve readability, we employed uniform color schemes, typefaces, and spacing. In order to eliminate clutter and give the images room to breathe, we also made care to provide ample white space. To help users navigate the dashboard and offer context, we also included the proper names, labels, and captions. The overall objective was to design an intuitive layout that would enable efficient and speedy data processing.

5. **Add interactivity and functionality:**

We put in place a number of features that would improve user experience and offer more information. To enable users to interactively examine the data according to their particular needs, we included filters and slicers. They were able to get deeper insights by drilling down into particular dimensions or time periods as a result. In order to enable users to navigate between various perspectives or scenarios, I also incorporated interactive components like buttons or dropdown menus. We sought to give consumers a dynamic and engaging dashboard experience that would enable them to explore and analyze the data efficiently by incorporating these interactive and functional features.

6. **Test and refine:**

To guarantee its reliability, usefulness, and usability, we used a methodical process. To make sure the visualizations were faithful to the underlying data, we began by thoroughly validating the data. We double-checked the data, compared it to the original data sources, and made sure the summaries and computations were accurate. After that, we conducted user testing with a wide range of people who reflected the intended population. We watched how they interacted with the dashboard, solicited comments, and took note of any usability problems or areas of misunderstanding. We iteratively improved the dashboard based on feedback by fixing any problems found, tweaking the design, and making the visualizations more understandable. We also paid close attention to how well the dashboard worked, making sure that it loaded swiftly and reacted to user input without any hiccups. We kept open lines of contact with stakeholders throughout the testing and refining process and solicited their feedback to match the dashboard with their

expectations and objectives. We were able to steadily improve the dashboard's functionality, usability, and overall user experience thanks to this iterative process.

Our Dashboard

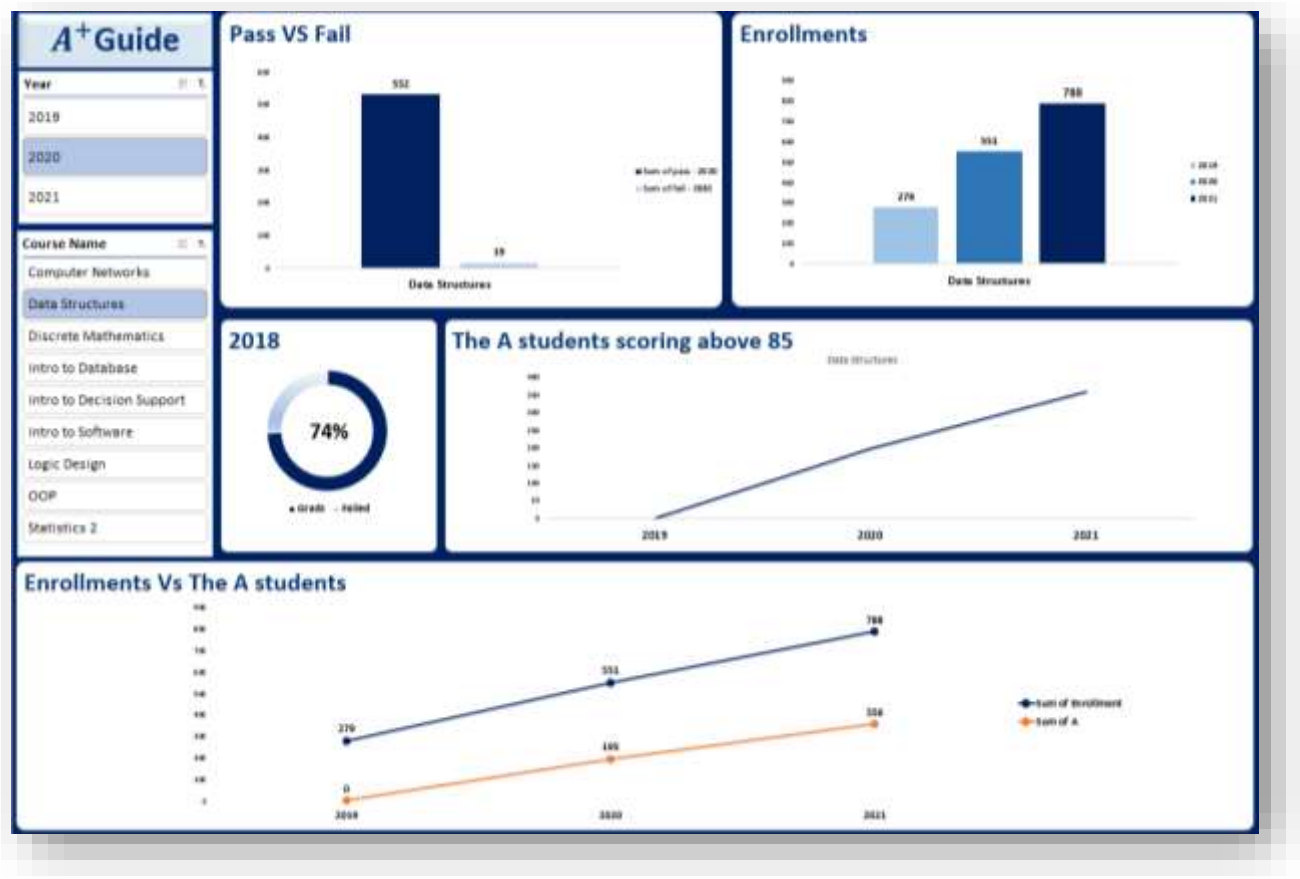

Fig 4

Our dashboards have been created to offer useful information across a range of categories, including average and beyond, as required by stakeholders. These analyses seek to provide a thorough knowledge of the data and to highlight important performance indicators for the stakeholders.

After understanding what the faculty board wanted and discussing with them what sort of KPIs they wanted to get out these data. They informed us with the needed courses to analyze its performance over the past 3 years 2019, 2020, and 2021.

These observations that we concluded from this dashboard are crucial for detecting problem areas or difficulties that may call for more attention and intervention.

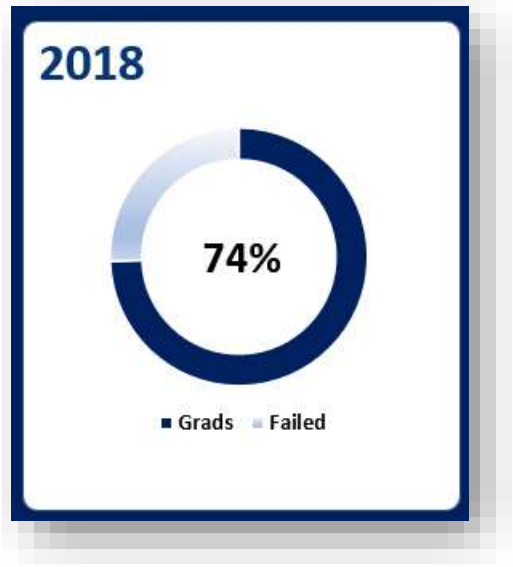

Fig 4.1

First of all, we used the data we have to calculate the graduation rates in our collage. In other words, we tried to figure out the number of students who successfully complete their educational program within a the four-years specified period.

We did not have any reference by which we could compare these results. So, we started searching for collages similar to ours who have similar programs and started to compare our results.

We found out that our collage lies between the above average and the outstanding category. This combined with the fact that Cairo University has just reached the rank of 371 worldwide between the world elite universities shows that the college is really excelling in delivering the best education to its graduates. Also, this shows that the college aid its students to help them overcome any limitations or barriers they face.

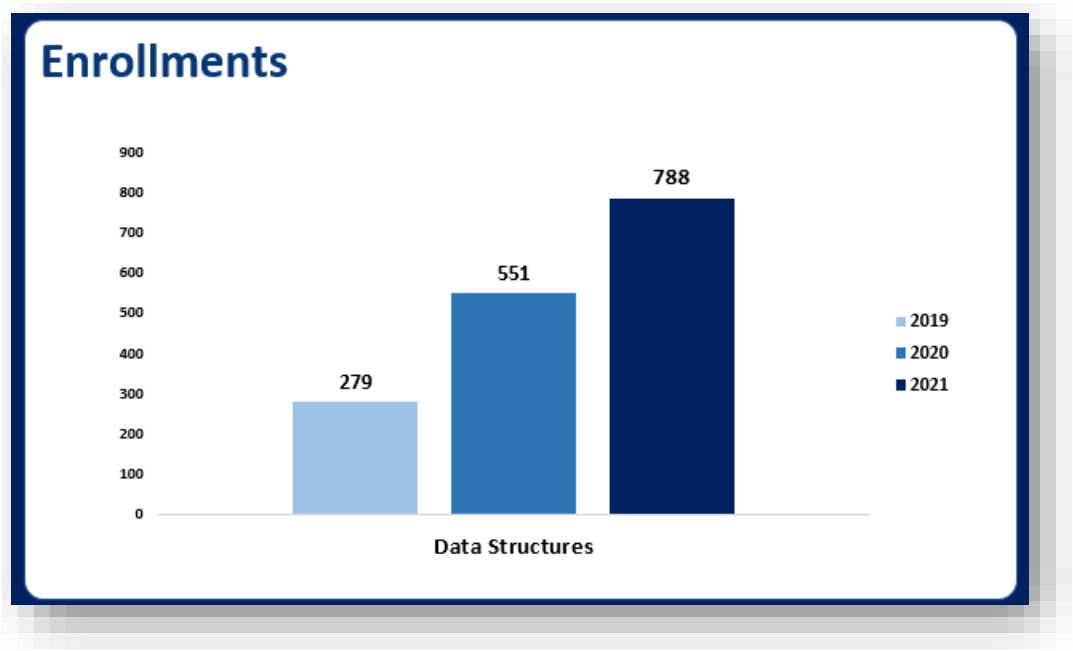

Fig 4.2

This bar chart shows the numbers of enrollments over the course of 3 years. This indicate that over the course of two years the amount of enrollment in this course in particular and all the other course that falls under our analysis, as they show the same result, have nearly tripled in size.

This considerable gain in size suggests that the number of students joining the faculty has grown substantially. It draws attention to a quick and noticeable expansion or improvement within the Computer and Artificial Intelligence major. This highlights the faculty's impressive growth trajectory between the other faculties.

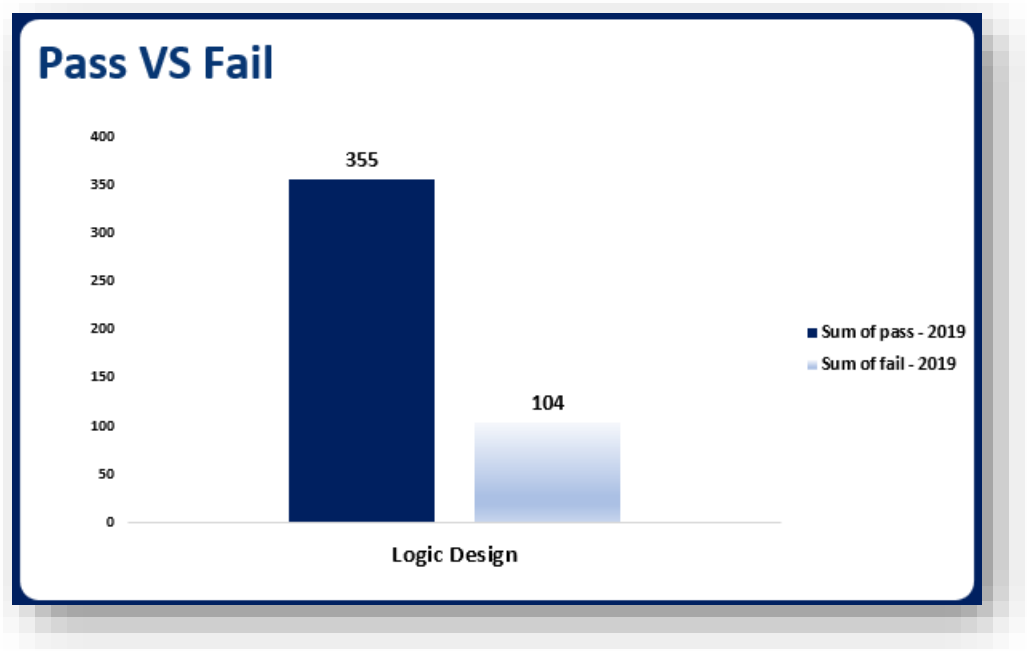

Fig 4.3

This chart shows the number of students who passed and failed the Logic Design course in 2019(which is the year we joined the faculty). This compared to the years 2020 and 2021, which have lower fail numbers, shows that this particular year the students faced great challenge in this course.

One explanation for such high failed students is that in 2019 the Corona Virus hit us. So, this could explain why so many students failed as they did not get to study in the faculty campus, and we were introduced with the online learning for the first time. This could serve as a good interpretation for such numbers.

CHAPTER 4

Methodology and Algorithm:

Implementing the Solution Approach.

Now as expected we want to explain what we did and how the work was done.

As we talked in chapter 2 about how we collected the data and grouped the student together (The same attitude the same grades in the same group), we are going to illustrate some concepts before starting in how the work had done:

1. The comparison between supervised and unsupervised algorithms
2. What is clustering?
3. The most suitable algorithm:
4. The details of work by clustering and after clustering
5. The recommendations system and why we use it and how it works.
6. The dashboards:
7. Conclusion and some helpful solution to the problem
8. Future work user interface and how we can improve this project.

Tools used:

1. Excel and spreadsheets
2. Pycharm the python Integrated Development Environment.

Unsupervised and Supervised learning:

1. Unsupervised machine learning:

Unsupervised machine learning, a subset of machine learning, is the process by which an algorithm learns to recognize correlations and patterns in data without being explicitly trained on labelled instances. Finding intriguing or practical data structures, such as clusters, is the aim of unsupervised learning.

Unsupervised learning does not have a specified output variable to predict, in contrast to supervised learning, where the algorithm is trained on a labelled dataset to learn to predict an output variable given an input. As an alternative, the program looks for patterns and connections in the dataset.

There are several types of unsupervised learning algorithms, including:

1. Clustering Algorithms: These algorithms put similar data points in groups based on how similar or far apart they are. K-means, hierarchical clustering, and density-based clustering are a few examples of clustering algorithms.
2. Dimensionality reduction methods: These algorithms convert the input data into a lower-dimensional space while retaining its fundamental structure, hence reducing the dimensionality of the data. Principal component analysis (PCA), t-SNE, and autoencoders are a few examples of dimensionality reduction algorithms.
3. Association rule mining: This involves discovering interesting relationships between variables in a dataset, such as identifying commonly co-occurring items in a transaction dataset.

And we will Focus on Clustering algorithms, which is most suitable for our data.

2. Supervised Machine Learning:

In supervised machine learning, an algorithm learns to make predictions or decisions based on labelled training data. In supervised learning, the algorithm is trained on a dataset that contains input features and the output labels or target values that correspond to those characteristics.

The goal of supervised learning is to learn a mapping function from the input features to the output labels, so that the algorithm can generalize and make accurate predictions on new, unseen data.

Problems involving classification and regression can both be solved using supervised learning. While the output labels in classification issues are discrete classes or categories, those in regression problems are continuous numerical values.

There are several types of supervised learning algorithms, including:

1. Linear regression: This algorithm models the relationship between the input features and the output labels as a linear function. It is commonly used for regression tasks where the output labels are continuous numerical values.
2. Logistic regression: This approach represents the relationship between the input features and the output labels. In binary classification problems, where the output labels can only be 0 or 1, it is frequently employed.
3. Decision trees: this method creates a tree-like model of decisions to forecast the output labels. Both classification and regression tasks frequently involve its application.
4. Support vector machines (SVM), Neural networks, etc.…

What is clustering?:

Using pairwise distances or similarities between data points, clustering algorithms are unsupervised machine learning techniques used to group related data points together. Clustering is a technique used to locate organic groupings or patterns in data without using labels or classifications beforehand.

We frequently group instances in machine learning as a first step to comprehend a subject (data set) in a machine learning system. Clustering is the process of collecting unlabeled samples.

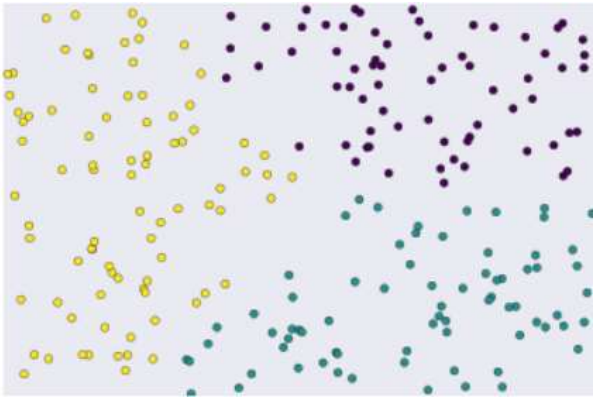

Fig 5

You can use a clustering method like K-means or hierarchical clustering and so many methods to do this.

Hierarchical clustering:

A well-liked unsupervised machine learning method called hierarchical clustering is used to put comparable data points together based on their pairwise distances or similarities. The method produces a hierarchy of nested clusters, where each cluster can either be a single data point or a collection of clusters that have already been produced.

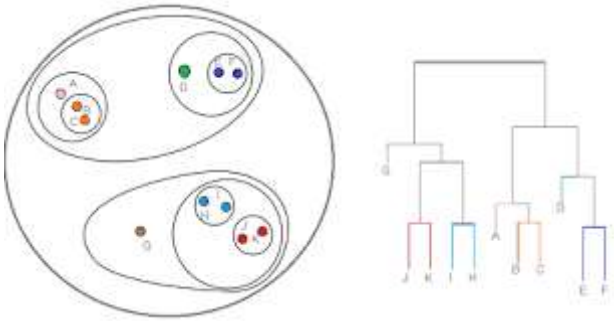

Fig 5.1

But we do not want to do that exactly, we want each cluster to represent a major (Department).

As mentioned before the most suitable approach for our data distribution is the k-means algorithm.

K-Means Algorithm:

Popular unsupervised machine learning algorithm K-means clustering is used to group related data points based on how similar or distant they are to one another. The algorithm is simple yet effective and is widely used in many fields.

The K-means algorithm works as follows:

1. Choose the number of clusters K that you want to create.
2. Initialize K cluster centroids randomly in the feature space.
3. Assign each data point to the nearest centroid.
4. Update the centroids by computing the means of all data points assigned to each centroid.
5. Repeat steps 3 and 4 until convergence, where convergence is defined as no further changes in the assignments of data points to clusters.

The within-cluster sum of squares (WCSS), also known as the sum of squared distances between each data point and its assigned centroid, is what the method seeks to minimize.

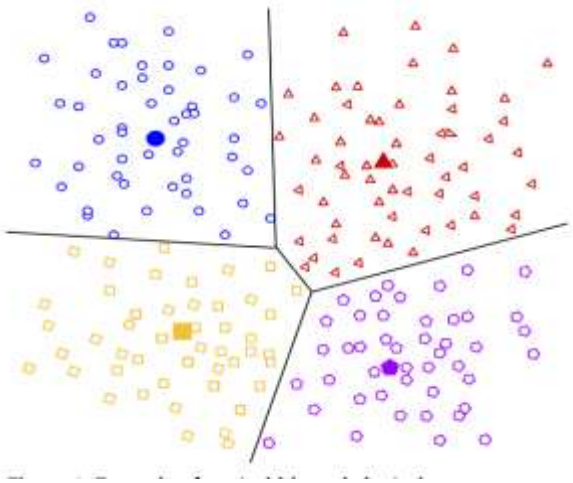

Fig6

Assigning data points to the nearest centroids:

We can use a distance metric, such as Euclidean distance, to calculate the distance between each data point and each centroid to assign each data point to the closest centroid in K-means clustering. The data point is then assigned to the centroid with the nearest distance.

The steps to assign each data point to the nearest centroid are as follows:

1.Calculate the distance between each data point and each centroid using the selected distance metric. One can calculate the distance between a data point x and a centroid c as follows:

distance(x, c) = sqrt($(x1 - c1)^2 + (x2 - c2)^2 + ... + (xn - cn)^2$)

where the coordinates of the data point x are x1, x2, xn, and the coordinates of the centroid care c1, c2, cn.

$$\text{Euclidean distance} = \sqrt{(2-5)^2 + (2--2)^2} = 5$$

Note:

Preprocess the data:

1. We Convert the grade and GPA. It is important to normalize the data to Make sure that the data is in a format that can be used by the clustering.
2. Put the data point on the centroid that is closest to it. If the distances between the data point x and the centroids c1, c2, ..., ck are d1, d2, ..., dk, respectively, then assign x to the centroid ci, where i is the index of the centroid that minimizes the distance di.
3. For each data point in the dataset, repeat steps 1 and 2

Why do we use K-Means?:

1. Simplicity: When compared to other clustering techniques, K-means clustering is more straightforward and simpler to comprehend. A simple iterative process underlies the algorithm, which only needs a few hyperparameters, including the number of clusters and the distance measure.
2. Speed: K-means clustering can manage huge datasets with numerous features and data points and is computationally efficient. The approach can be parallelized to increase speed even further and grows well with the size of the dataset.
3. Effectiveness: For datasets with well-separated clusters and comparable cluster sizes and shapes, K-means clustering can be quite effective. A tight and well-separated cluster can result from the algorithm's attempt to reduce the sum of squared distances between each data point and its designated centroid.
4. Flexibility: K-means clustering is flexible in that it can be employed with various distance metrics and initiation techniques. Through tweaks or adjustments, the algorithm can also be made to manage more intricate data formats, such category, or binary data.
5. Interpretability: Because each data point is assigned to a single cluster and each cluster is represented by its centroid, K-means clustering generates results that are simple to read. This may make it easier to see the underlying trends and connections in the data.

CHAPTER 5

Project Execution and Implementation:

Detailed Work Overview.

K-Means implementation:

After preparing the data to be analyzed and loaded it into the program. And save the data into "Students Grades.csv"

Students IDs	CS213	CS214	IS211	CS251	MA112	IT212	IT221	DS211	ST222	GPAs	Dep
500	67	0.0001	56	50	57	57	50	64	65	1.56	DS
505	79	0	0	0	64	50	50	53	52	1.7	DS
507	91	0	0	0	78	77	86	95	78	3.56	CS
515	70	0	0	0	62	54	71	75	80	3.1	AI
516	73	0	0	0	54	0	67	73	68	2.25	IT
520	64	0.0001	50	41	50	64	0	56	36	1.42	IS

Fig 7

We set out to use a k-means clustering algorithm to find if there was a relationship between students' scores on prerequisite courses and their choice of departments. The prerequisite courses are the courses that the student needs to pass in order to be able to register for other courses that are related to the first.

At the beginning of the implementation, the two most effective and influential courses on the rest of the courses of each major (prerequisite courses) from 1st and 2nd levels were determined for each major in the 3rd and 4th years. We have settled the K to five that describes the number of clusters. Where each cluster should contain at least one major that is the most dominant major in each cluster.

After reviewing the bylaw of FCAI the prerequisites of each department have been determined as follows:

Department	**First prerequisite**	**Second prerequisite**
Computer Science	Object Oriented Programming	Data Structures
Information System	Introduction to Software Engineering	Introduction to Database Systems
Artificial Intelligence	Logic design	Discrete Math
Information Technology	Computer Networks Technology	Logic design
Decision Support	Introduction to Operations Research and Decision Support	Probability and Statistics-2

Table 1

Data dimensions:

K-means clustering can suffer from the "curse of dimensionality", where the distance between points becomes less meaningful as the number of dimensions increases.

Usually, PCA (Principal Component Analysis) can help to address these issues by reducing the dimensionality of the data and by transforming it into a new coordinate system, where the variables are linearly uncorrelated and ordered by the amount of variance they explain in the original data. PCA can be used in clustering when dealing with high-dimensional data, where the number of variables is much larger than the number of observations, and the variables are correlated. By selecting only, the most important principal components, the dimensionality of the data is reduced while retaining most of the information.

PCA works by finding the linear combinations of the original variables (called principal components) that maximize the amount of variance explained in the data. The first principal component is the linear combination that explains the most variance, the second principal component is the linear combination that explains the most variance after accounting for the first principal component, and so on. The number of principal components is equal to the number of variables in the data, but most of them explain little variance and can be discarded without losing much information.

Clustering using K-Means:

The k-means clustering algorithm was implemented on students' scores in these nine courses and their departments.

Using StandardScaler which is a data preprocessing technique that is used to standardize a dataset by scaling its features to have zero mean and unit variance. StandardScaler works by computing the mean and standard deviation of each feature in the dataset, and then subtracting the mean and dividing by the standard deviation for each feature.

The results of these clusters were somewhat disappointing, as we could not determine the most dominant department in each of the five clusters due to the data

available to us from the college administration, which was about 1700 students who fell under the GPA system that was approved by the college administration and was applied at the beginning From the fall of 2018, which is still being approved to the present time. These students have succeeded in passing both the first and second levels, and the number of their accumulated hours has reached a minimum of 45 hours to allow them to choose the majors they desire.

The result can be illustrated as following:

Cluster 0		**Cluster 1**		**Cluster 2**		**Cluster 3**		**Cluster 4**	
IS	160	CS	75	IT	59	IS	86	CS	93
CS	134	IS	69	DS	41	CS	77	IS	87
AI	123	AI	60	AI	22	AI	75	AI	74
DS	114	IT	39	IS	16	IT	57	DS	61
IT	111	DS	21	CS	13	DS	43	IT	61

Table 2

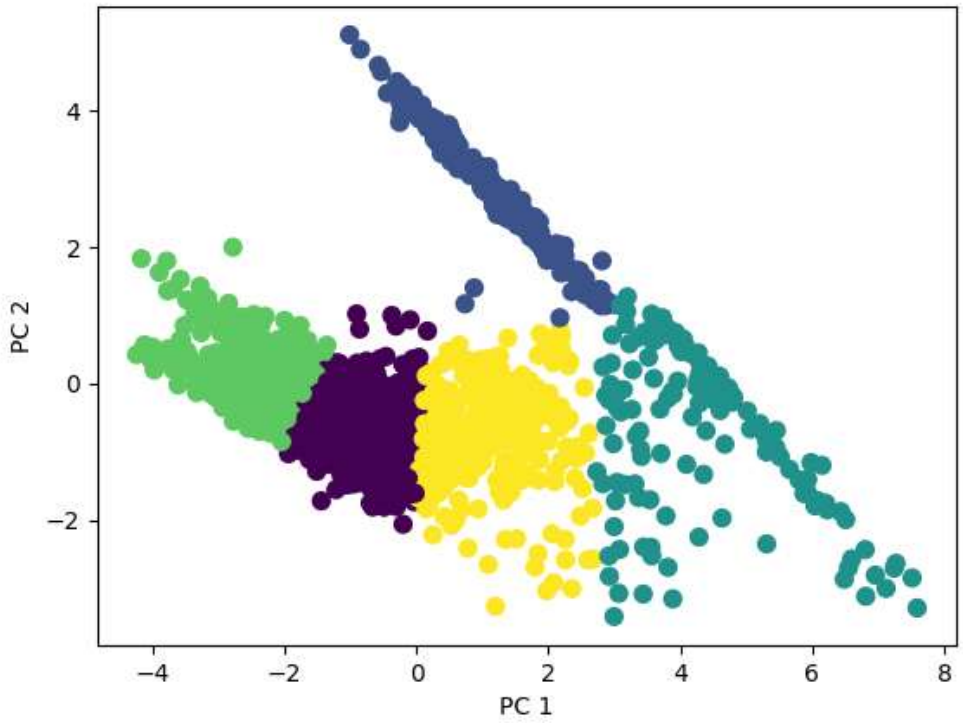

Fig 8

In this figure each student is represented by a dot, and each color represents only one cluster. The result of each student's cluster in which he has joined could be found in "Students Clusters.csv" file.

Students majors selection:

The correlation between students' scores in the prerequisite courses and selecting the major has been found around 28.2%.

This correlation represents the strength of the relation between students' scores in the prerequisite courses and selecting the majors and describes how students became irrational when it comes to selecting their majors and shows the lack of knowledge of how important the prerequisites of each major and how they affect the overall performance in both 3rd and 4th levels, and it found as follows:

Calculating the correlation between the "Dep" variable and all the other variables in the dataset.

Then drops the "Dep" variable from the correlation matrix and calculates the sum of the absolute values of the remaining correlations. This gives an indication of how strongly the prerequisite courses are correlated with the "Dep" variable. By taking the absolute value of each correlation coefficient, the analysis is focused on the strength of the correlation, regardless of its direction (positive or negative).

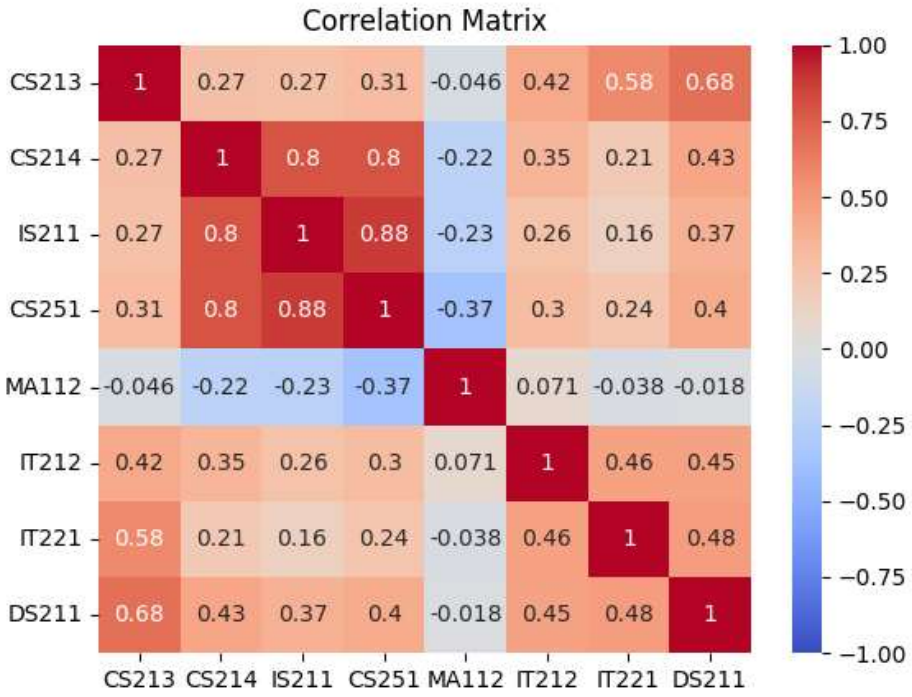

Fig 9

Overall, this analysis can help identify which prerequisite courses are most strongly related to the "Dep" variable and may provide insights into how to improve student performance in the target course.

A+ Guide Recommendation System:

For students who are indifferent between the departments and do not know the suitable department for them, A+ Guide offers them a recommendation system to helps them with an ordered departments recommendations based on their scores in each department's prerequisite courses.

For each student, the summation of each two prerequisite courses of each department had been calculated, then ordering this recommendation in descending order then saving it in "Recommended Departments.csv" file.

Now each student is listed as a vector represented by his ID, his grades in the prerequisite courses, his summation of the prerequisite courses of each department, and finally the five recommendations of the departments (in descending order).

Stude	CS213	CS214	IS211	CS251	MA112	IT212	IT221	DS211	ST222	GPAs	Dep	CS	IS	AI	IT	DS	1st re	2st re	3st re	4st re	5st re
500	67	1E-04	56	50	57	57	50	64	65	1.56	DS	67	106	114	107	129	DS	AI	IT	IS	CS
505	79	0	0	0	64	50	50	53	52	1.7	DS	79	0	114	100	105	AI	DS	IT	CS	IS
507	91	0	0	0	78	77	86	95	78	3.56	CS	91	0	155	163	173	DS	IT	AI	CS	IS
515	70	0	0	0	62	54	71	75	80	3.1	AI	70	0	116	125	155	DS	IT	AI	CS	IS
516	73	0	0	0	54	0	67	73	68	2.25	IT	73	0	54	67	141	DS	CS	IT	AI	IS
520	64	1E-04	50	41	50	64	0	56	36	1.42	IS	64	91	114	64	92	AI	DS	IS	CS	IT

Fig 10

After putting students into this representation, it is time for GPA constraint that deciding if the recommended department is eligible for each student to enroll in it or not.

The GPA distribution is not fixed for all years. It was agreed with the administration of the Faculty of Computing and Intelligence at Cairo University represented by Dr. Ihab E- Khodary and the A+ Guide team to adopt the GPA distribution for the year 2022-2023 as the initial distribution of "A+ Guide Recommendation System".

GPA distribution for year 2022-2023 had applied to 2020 fall as follows:

Department	Minimum GPA to enroll
IS	2.79
CS	2.71
AI	2.40
IT	The least GPA recorded and passed the minimum hours (45) to enroll in a department (set initially for 1.00)
DS	The least GPA recorded and passed the minimum hours (45) to enroll in a department (set initially for 1.00)

Table 3

After setting the GPA distribution the program iterates over each student and checks the five recommendation of departments -illustrated previously- if all are eligible for him to enroll and excludes the departments that fall out of the range of its GPA. Then save the result into "Final result.csv"

Students IDs	GPA	Eligible Major 1	Eligible Major 2	Eligible Major 3	Eligible Major 4	Eligible Major 5
Student15	1.98	DS	IT			
Student16	3.57	IT	AI	DS	CS	IS
Student17	1.32	DS	IT			
Student18	3.05	IT	AI	CS	IS	DS
Student19	3.01	AI	CS	IS	DS	IT
Student20	3.82	IS	CS	IT	AI	DS

Fig 10.1

The program contains a function that is responsible for randomizing a sample of students using the "random" library in python. This function randomizes a grade for each course of prerequisite courses in range of (0:100) and GPA in range of (1:4).

For those students who represents the new students to enroll in departments for the next year, A+ Guide team offers them the results of the recommendation system to guide them which department is most suitable for them to enroll in it based on their performance in the prerequisite courses he passed in the first two years and the GPA they had.

Students IDs	GPA	Eligible Major 1	Eligible Major 2	Eligible Major 3	Eligible Major 4	Eligible Major 5
500	1.56	DS	IT			
505	1.7	DS	IT			
507	3.56	DS	IT	AI	CS	IS
515	3.1	DS	IT	AI	CS	IS
516	2.25	DS	IT			
520	1.42	DS	IT			

Fig 11

Recommendation system accuracy:

Accuracy is a widely used measure of performance across many fields of life, and its method of measurement can vary depending on the specific context. It is a crucial measure of performance, as it provides insight into how well a particular task or system is performing. Accuracy can be influenced by several factors, such as the quality of data or the specific context in which the measurement is being made.

So, the data provided to us by the administration of the faculty was searched for students who happened to enter the department that was nominated for them through "A+ Guide recommendation system" which works without the GPA constraint. To make the comparison fairer, students who have completed both the third and fourth levels were selected, and the rest who are still enrolled in the college were excluded. Who are represented by approximately 330 records (about 19% of the historical data).

Their performance in both the first and second levels (pre-major), and then compared it with his performance in both the third and fourth levels (post-major) can be shown with in this graph:

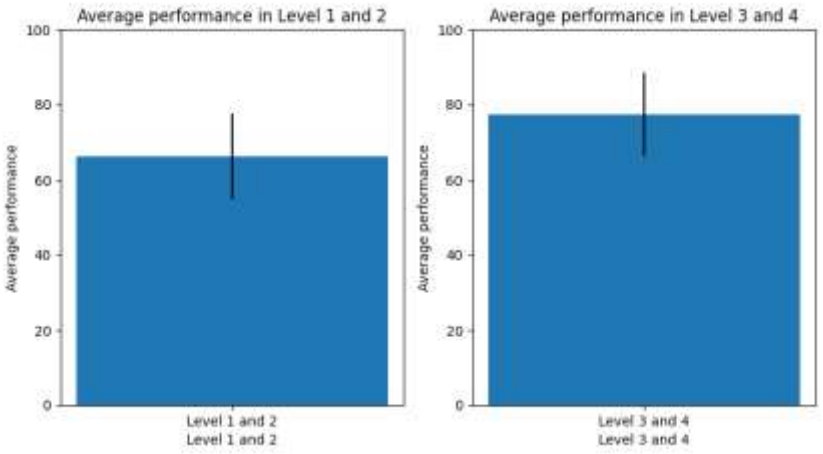

Fig 12

This analysis showed that there are 91 students from every 100 students, their performance has been improved in third and fourth levels than the first and second levels.

On the other hand, the students who happened to enter the department that was nominated for them through "A^+ Guide recommendation system" which works GPA constraint are represented by approximately 780 records (about 44% of the historical data).

Their performances in both the first and second levels (pre-major), and then compared it with their performances in both the third and fourth levels (post-major) within this graph:

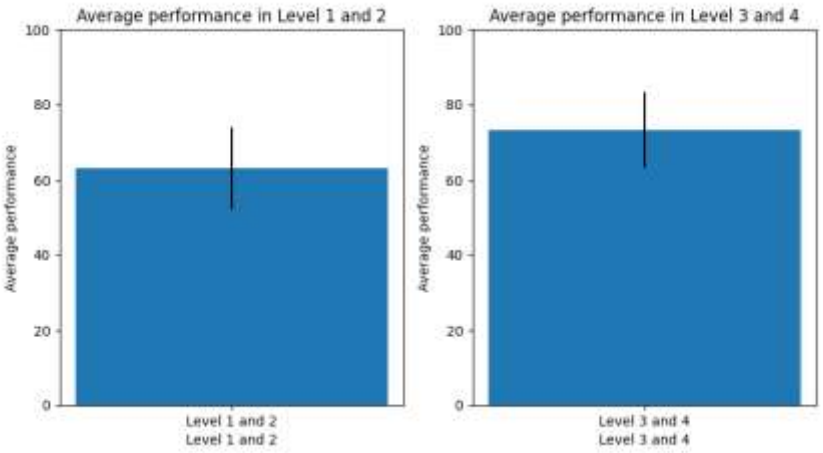

Fig 12.1

This analysis showed that there are 77 students from every 100 students, their performances have been improved in third and fourth levels than the first and second levels.

We can clearly see the drop in the percentage after adding the GPA constraint in selecting the student's major.

Where the remaining students chose their departments based on their whims and without knowledge.Their performance in both the first and second levels (pre-major), and then compared it with his performance in both the third and fourth levels (post-major) within this graph:

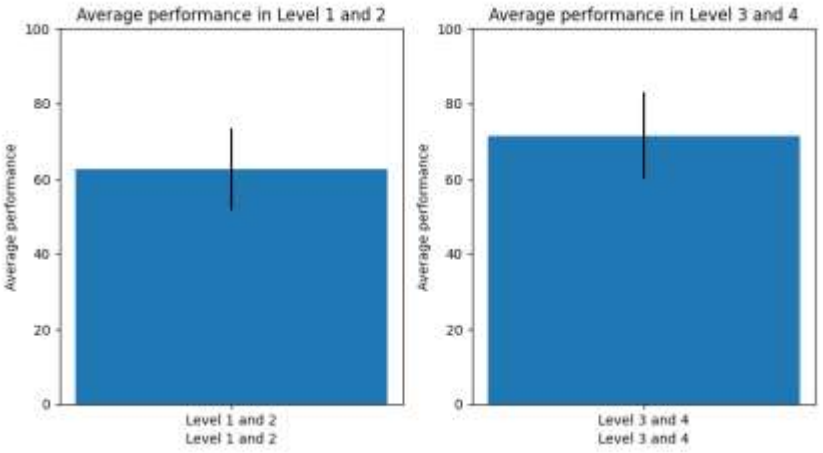

Fig 12.2

This analysis showed that there are only 51 students from every 100 students, their performances have been improved in third and fourth levels than the first and second levels.

Results of testing:

The comparison between students' performances, who we can say they have followed our recommendations of the first department showed that about 91.19% of students their performance in Level 3 and 4 was better than their performance in level 1 and 2.

These results showed how prerequisite courses play a key role in students' performance in the upcoming years, and how these courses have affected the performance of students who did not study the situation correctly and chose their departments unwisely.

There is still our question remaining are there more students who would be able to perform better if they chose the right major without GPA constraint?

Does the GPA system hinder students from enrolling in the most appropriate major for them and increasing their chances of improving their performance in the following two years?

CHAPTER 6

Interface Design and Future Scope:

Enhancing User Experience and Expanding Functionality

User interface:

We made a user interface to visualize the work we made as a web application or mobile application to easily understand the process for any student.

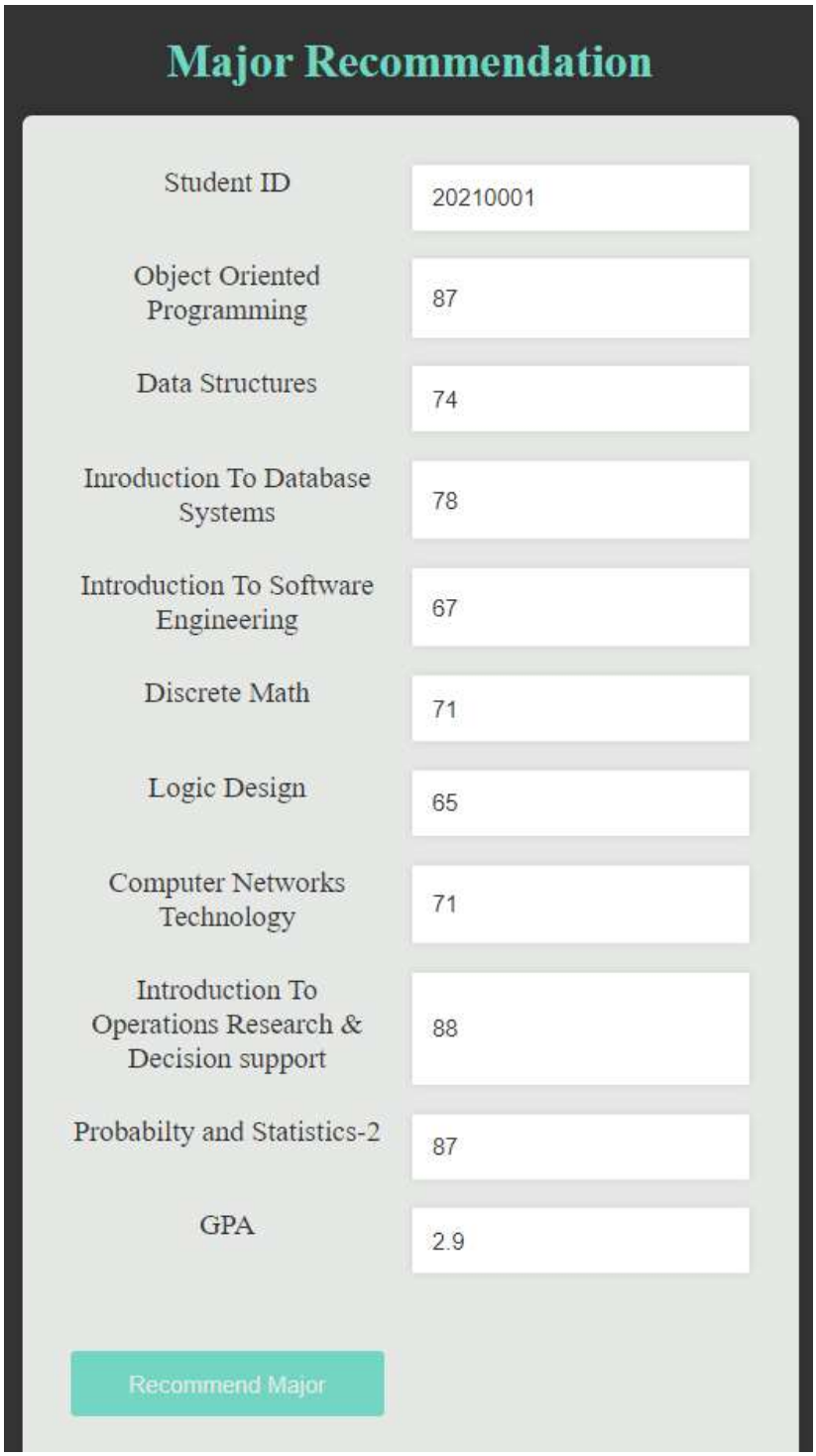

Fig 13

Steps:

1. Each student must enter their own ID.
2. The grades he got in all the mentioned courses.
3. Find the recommended major.
4. Each Student now has the recommended major based on his grades in the prerequisite course for each major.

Fig 13.1

Future work:

1. After the user interface we are seeking to do something bigger as web application or mobile application so each student enters the faculty can use it to guide him.

2. We are seeking also to make a dashboard to students' performance at the end of each year so he can improve himself by looking to the student who look like him in performance and bigger than him in level and do what he did and to compare his performance each new year with past year.

3. Verifying the project by trace the students who listen to us and enter the project we recommend to them and see if their performance is improving.

CHAPTER 7

Conclusion and Findings:

Summarizing the Results and Implications

Conclusion:

By the end of the project, we are offering a recommendation system that is able to recommend the most suitable department to students where they can perform better.

The dashboard shows the percentage of number of students graduated and number of students didn't and the percentage of people who get by luck into the major recommended to him their performance get better this may after period of time makes the percentage of students who graduated raise and the performance of students at each major raise and after graduated each student maybe more applicable for job better than before and students' reputation graduated from FCAI raise and.

There will be always a chance to student who miss the major that he most qualified for because of the GPA by:

The student who has the highest GPA (close to 4), but the X major is not the most suitable for him, he will leave an empty place to the one who did not reach the minimum GPA to the major (maybe), so each one can get better performance at the most suitable major (the department limit will always not fully complete).

At the end as we say in phase 1, we did not know what is actually better for us (Mario, Ahmed, Hossam) and there are many students like us do not know what is going on and want some help to guide them

So, here is our recommendation system:

A^+ Guide

References

What is Clustering? (n.d.). Google for Developers. https://developers.google.com/machine-learning/clustering/overview

Clustering Algorithms. (n.d.). Google for Developers. https://developers.google.com/machine-learning/clustering/clustering-algorithms

Machine Learning Glossary. (n.d.). Google for Developers. https://developers.google.com/machine-learning/glossary#k-means

Ecosystem, E. (2022, May 17). Understanding K-means Clustering in Machine Learning. Medium. https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1

Babitz, K. (2023). Introduction to k-Means Clustering with scikit-learn in Python. https://www.datacamp.com/tutorial/k-means-clustering-python

Alteryx. (2023, June 21). Supervised vs. Unsupervised Learning; Which Is Best? - Alteryx. https://www.alteryx.com/glossary/supervised-vs-unsupervised-learning#:~:text=Supervised%20and%20unsupervised%20learning%20have,unsupervised%20learning%20uses%20unlabeled%20datasets

Soni, D. (2020, July 21). Supervised vs. Unsupervised Learning - Towards Data Science. Medium. https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d

Appendix

```
import csv
import random
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler


def add_random_students(students_num):
    course = ["CS213", "CS214", "IS211", "CS251", "MA112", "IT212", "IT221",
"DS211", "ST222"]
    with open('Students Grades.csv', mode='a', newline='') as file1, \
            open('Students Grades2.csv', mode='a', newline='') as file2:
        writer1 = csv.writer(file1)
        writer2 = csv.writer(file2)
        for i in range(students_num):
            name = "Student" + str(i + 1)
            grades = [random.randint(0, 100) for _ in range(len(course))]
            GPA = round(random.uniform(1.0, 4.0), 2)
            writer1.writerow([name] + grades + [GPA])
            writer2.writerow([name] + grades + [GPA])
            print("Grades added for", name)
    print("Done adding", students_num, "students.")


add_random_students(20)

# Load the data
df = pd.read_csv("Students Grades.csv")
# Extract the features and labels
X = df.drop(["Students IDs", "GPAs", "Dep"], axis=1)
y = df["Dep"]

# Scale the data
scale = StandardScaler()
X_scaled = scale.fit_transform(X)

# Perform PCA to reduce the dimensionality
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

# Perform K-Means clustering
kmeans = KMeans(n_clusters=5, random_state=25)
clusters = kmeans.fit_predict(X_pca)

# Visualize the clusters
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=clusters, s=50, cmap='viridis')
plt.xlabel('PC 1')
plt.ylabel('PC 2')
plt.show()

# Save the cluster labels to a CSV file
```

```
df_clusters = pd.DataFrame({"Student ID": df["Students IDs"], "Cluster
Label": clusters})

# Combine the original data with the cluster labels and save to a CSV file
df_combined = pd.concat([df, df_clusters["Cluster Label"]], axis=1)
df_combined.to_csv("Students Clusters.csv", index=False)

# Select data where "Dep" column is not null
df = df[df['Dep'].notnull()]
df = df.drop(["Students IDs", "GPAs"], axis=1)

# Calculate correlation matrix
corr_matrix = df.corr(numeric_only=True)

# Extract correlations between "Dep" and Grades
dep_corr = corr_matrix.iloc[:-1, -1]

# Calculate sum of absolute correlations
sum_abs_corr = np.sum(np.abs(corr_matrix.values.flatten()))

print("Correlation between Dep and Grades:", sum_abs_corr)

# Plot heatmap of correlation matrix
sns.heatmap(corr_matrix, cmap='coolwarm', annot=True, vmin=-1, vmax=1)
plt.title("Correlation Matrix")
plt.show()

# Illustrate each cluster
for i in range(5):
    print(f"Cluster {i}:")
    cluster_data = df_combined[df_combined["Cluster Label"] == i]
    print(cluster_data["Dep"].value_counts())

# Courses recommendation system
data = pd.read_csv("Students Grades.csv")

# define the majors and their prerequisite courses
majors = {"CS": ["CS213", "CS214"], "IS": ["CS251", "IS211"], "AI": ["IT212",
"MA112"],
          "IT": ["IT212", "IT221"], "DS": ["DS211", "ST222"]}

# calculate the sum of scores for each major
for major, courses in majors.items():
    data[major] = data[courses].sum(axis=1)

# recommend majors for each student
for i in range(1, 6):
    data[f"{i}st recommendation"] = data[majors.keys()].idxmax(axis=1)
    for major in majors:
        data = data[data[major] != data[major].max()]

# sort the recommendations in descending order
for i in range(1, 6):
    data[f"{i}st recommendation"] = data.apply(lambda x:
sorted(majors.keys(), key=lambda y: x[y], reverse=True)[i - 1],
                                               axis=1)

data.to_csv("Recommended Departments.csv", index=False)
```

```
# Eligible courses due the GPA constrain
df = pd.read_csv('Recommended Departments.csv')

# Create a copy of the dataframe without the "Dep" column
df_without_dep = df.drop(columns=['Dep'])

# Define the GPA ranges and their corresponding major eligibility
gpa_ranges = {'range1': {'min': 1, 'max': 2.40, 'majors': ['IT', 'DS']},
              'range2': {'min': 1, 'max': 2.71, 'majors': ['AI', 'IT',
'DS']},
              'range3': {'min': 1, 'max': 2.79, 'majors': ['CS', 'AI', 'IT',
'DS']},
              'range4': {'min': 1, 'max': 4.00, 'majors': ['IS', 'CS', 'AI',
'IT', 'DS']}}

# Create a new dataframe to store the output
output_df = pd.DataFrame(columns=['Student ID', 'GPA', 'Eligible Major 1',
'Eligible Major 2', 'Eligible Major 3',
                                  'Eligible Major 4', 'Eligible Major 5'])

# Initialize an empty list to store output dataframes
output_dfs = []

# Iterate over each row in the dataframe without the "Dep" column
for index, row in df_without_dep.iterrows():

    # Get the student's GPA
    gpa = row['GPAs']

    # Check which GPA range the student falls into
    for key, value in gpa_ranges.items():
        if value['min'] <= gpa <= value['max']:

            # Check if the student's department is eligible for all
recommended majors
            eligible_majors = []

            for major in row[1:]:
                if major in value['majors']:
                    eligible_majors.append(major)

            # Create a new output dataframe for the current student
            output_row = {'Students IDs': row['Students IDs'], 'GPA':
row['GPAs']}
            i = 0
            for i, major in enumerate(eligible_majors):
                output_row[f'Eligible Major {i + 1}'] = major

            while i < 4:
                i += 1
                output_row[f'Eligible Major {i + 1}'] = ''

            output_df = pd.DataFrame(output_row, index=[0])
            output_dfs.append(output_df)
            break

# Concatenate all output dataframes into a single dataframe
output_df = pd.concat(output_dfs, ignore_index=True)
```

```
output_df.to_csv('Final result.csv', index=False)

# System testing
df1 = pd.read_csv("System test lvl1,2.csv")
df1_no_ids = df1.drop("Students IDs", axis=1)
df2 = pd.read_csv("System test lvl3,4.csv")
df2_no_ids = df2.drop("Students IDs", axis=1)

# Calculate the average for each row in both files
avg1 = df1_no_ids.mean(axis=1)
avg2 = df2_no_ids.mean(axis=1)

# Combine the results into a single DataFrame
results = pd.DataFrame({"Students IDs": df1["Students IDs"], "Avg lvl1,2":
avg1, "Avg lvl3,4": avg2})

# Calculate the percentage of students with better performance in the next 2
years
num_students = len(results)
num_greater = len(results[results["Avg lvl3,4"] > results["Avg lvl1,2"]])
percentage_greater = (num_greater / num_students) * 100

# Print the percentage of students
print("The percentage of students whose performance in Level 3 and 4 is
better"
      " than their performance in  lvl 1 and 2 is:
{:.2f}%".format(percentage_greater))

# Create a figure with two subplots
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 5))

# Create a bar plot for the average performance in Level 1 and 2
ax1.bar(["Level 1 and 2"], [avg1.mean()], yerr=[avg1.std()])
ax1.set_ylim([0, 100])
ax1.set_xlabel("Level 1 and 2")
ax1.set_ylabel("Average performance")
ax1.set_title("Average performance in Level 1 and 2")

# Create a bar plot for the average performance in Level 3 and 4
ax2.bar(["Level 3 and 4"], [avg2.mean()], yerr=[avg2.std()])
ax2.set_ylim([0, 100])
ax2.set_xlabel("Level 3 and 4")
ax2.set_ylabel("Average performance")
ax2.set_title("Average performance in Level 3 and 4")
plt.show()

# System testing
df1 = pd.read_csv("System test lvl1,2 not.csv")
df1_no_ids = df1.drop("Students IDs", axis=1)
df2 = pd.read_csv("System test lvl3,4 not.csv")
df2_no_ids = df2.drop("Students IDs", axis=1)

# Calculate the average for each row in both files
avg1 = df1_no_ids.mean(axis=1)
avg2 = df2_no_ids.mean(axis=1)

# Combine the results into a single DataFrame
results = pd.DataFrame({"Students IDs": df1["Students IDs"], "Avg lvl1,2":
avg1, "Avg lvl3,4": avg2})
```

```
# Calculate the percentage of students with better performance in the next 2
years
num_students = len(results)
num_greater = len(results[results["Avg lvl3,4"] > results["Avg lvl1,2"]])
percentage_greater = (num_greater / num_students) * 100

# Print the percentage of students
print("The percentage of students whose performance in Level 3 and 4 is
better"
      " than their performance in  lvl 1 and 2 is:
{:.2f}%".format(percentage_greater))

# Create a figure with two subplots
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 5))

# Create a bar plot for the average performance in Level 1 and 2
ax1.bar(["Level 1 and 2"], [avg1.mean()], yerr=[avg1.std()])
ax1.set_ylim([0, 100])
ax1.set_xlabel("Level 1 and 2")
ax1.set_ylabel("Average performance")
ax1.set_title("Average performance in Level 1 and 2")

# Create a bar plot for the average performance in Level 3 and 4
ax2.bar(["Level 3 and 4"], [avg2.mean()], yerr=[avg2.std()])
ax2.set_ylim([0, 100])
ax2.set_xlabel("Level 3 and 4")
ax2.set_ylabel("Average performance")
ax2.set_title("Average performance in Level 3 and 4")
plt.show()

# System testing
df1 = pd.read_csv("System test lvl1,2 GPA.csv")
df1_no_ids = df1.drop("Students IDs", axis=1)
df2 = pd.read_csv("System test lvl3,4 GPA.csv")
df2_no_ids = df2.drop("Students IDs", axis=1)

# Calculate the average for each row in both files
avg1 = df1_no_ids.mean(axis=1)
avg2 = df2_no_ids.mean(axis=1)

# Combine the results into a single DataFrame
results = pd.DataFrame({"Students IDs": df1["Students IDs"], "Avg lvl1,2":
avg1, "Avg lvl3,4": avg2})

# Calculate the percentage of students with better performance in the next 2
years
num_students = len(results)
num_greater = len(results[results["Avg lvl3,4"] > results["Avg lvl1,2"]])
percentage_greater = (num_greater / num_students) * 100

# Print the percentage of students
print("The percentage of students whose performance in Level 3 and 4 is
better"
      " than their performance in  lvl 1 and 2 is:
{:.2f}%".format(percentage_greater))

# Create a figure with two subplots
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 5))
```

```
# Create a bar plot for the average performance in Level 1 and 2
ax1.bar(["Level 1 and 2"], [avg1.mean()], yerr=[avg1.std()])
ax1.set_ylim([0, 100])
ax1.set_xlabel("Level 1 and 2")
ax1.set_ylabel("Average performance")
ax1.set_title("Average performance in Level 1 and 2")

# Create a bar plot for the average performance in Level 3 and 4
ax2.bar(["Level 3 and 4"], [avg2.mean()], yerr=[avg2.std()])
ax2.set_ylim([0, 100])
ax2.set_xlabel("Level 3 and 4")
ax2.set_ylabel("Average performance")
ax2.set_title("Average performance in Level 3 and 4")
plt.show()
```