ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ARTIFICIAL INTELLIGENCE

Master Thesis

in

Languages and Algorithms for Artificial Intelligence

Exploring students' learning state through clustering and knowledge graphs

CANDIDATE
Gaia Ebli

SUPERVISOR
Prof. Maurizio Gabbrielli

CO-SUPERVISORS
Dott. Stefano Pio Zingaro
Dott. Andrea Zanellati
Dott. Francesco Balzan

Session III
Academic year 2022-2023

Introduction

In today's educational landscape, we face a number of significant challenges in analyzing, understanding, assessing and intervening in the student learning process [1][2][3][4]. One of the main dilemmas is the difficulty in recognizing and addressing problems that students may encounter in the learning environment, in the learning path and in their individual development [5][6].

The first obstacle is the lack of clarity and methodologies on how to detect early signs of problems in student learning. It is not always immediate to recognize when a student is experiencing difficulties or dissatisfaction with the learning environment or the learning path they are following [7]. Moreover, even if we can identify these challenges, it can be equally complex to determine where and how to intervene to effectively support the student [6].

However, if we could profile the learning paths or even just the current learning state of students and understand what factors they are characterized by, but more importantly influenced by, we might be able to intervene in a preventive manner to avoid phenomena such as low achievement or student dropout in the future [8][9][10].

Benefiting, of course, would not only be the students, who would be supported and helped even before any problems and difficulties arise during the learning path, but also the teachers who could, for example, personalize curricula for students [11]. In addition to these two main actors in the school system, families [12], as well as school principals and staff [13], but also those who design educational pathways and assessment tools for learning paths could benefit enormously from any opportunity for greater understanding of learning states and paths [14][15].

This thesis work aims to make an exploratory analysis of the responses of fifth-grade students of the province of Forlì-Cesena to the INVALSI test in math-

ematics, a national standardized test used in Italy to assess students' proficiency in basic subjects, particularly mathematics, Italian and English, with the goal of identifying possible common factors in the performance of the test that may be useful to a domain expert for a more in-depth analysis of students' learning state at the time of the test. The INVALSI test is structured to include multiple-choice questions, open-ended questions and problems that assess students' understanding, application and analysis of knowledge and skills. Test results are used to monitor student progress over time, identify learning trends and provide useful information for developing educational policies [16].

Each student's answers are modeled with two different types of representations: the first is a directed and weighted graph [17] that connects the different test questions based on the correctness of the student's answers and the dimensions in common between the questions, i.e. whether they are both algebra or geometry questions for example; the second representation is a spanning tree [17], i.e. a version of the graph just described with the minimum number of edges and without cycles [17], which was developed with two variants of the depth-first strategy [18].

In addition, some global metrics were calculated on these three representations, the graphs and the two versions of spanning trees, to get a general idea of some characteristics of students' performance on the test, such as understanding whether students are more proficient in some dimensions than others.

The three representations and the metrics computed on them were then analyzed with three different clustering algorithms [19], k-Means [20], DBSCAN [21], and Gaussian Mixture Model [22], to try to partition the students in order to identify which factors characterize or influence each cluster.

What emerges from this work is that there is no direct correlation between the clustering results and possible student dropout in the future, however, from the clustering results on the metrics of the various representations, metrics can be selected that can efficiently partition the data and that could be a useful tool for any domain experts to understand what factors characterize the performance of the test and consequently the student's current learning state.

This paper is organized as follows: chapter 1 proposes an extensive literature review on the effectiveness of clustering and non-clustering techniques in edu-

cation, chapter 2 describes the proposed method, including the various ways in which the data were represented and the clustering techniques used for analysis, chapter 3 reports the results, and chapter 4 discusses the validity and limitations of the results.

Contents

List of Figures

List of Tables

Chapter 1

Literature review

Extensive research has been carried out over the years in an attempt to profile the learning path of students with the dual purpose of identifying where students may encounter difficulties and using this information to predict their academic future [23].

Students' performance and learning paths over time have been analysed with a lot of different machine learning and deep learning techniques. For example, lot of research works tested different decision tree algorithms to predict students' performance. Decision tree algorithms are a family of machine learning algorithms, whose main goal is to create a predictive model that can be represented in the form of a decision tree structure, where each internal node represents a question about an attribute of the data, each branch output from that node represents a possible answer to that question, and each leaf represents a class or output value. Through an iterative process, the algorithm tries to divide the dataset into increasingly homogeneous subsets until it reaches a stopping condition, such as a maximum depth of the tree or sufficient purity of the subsets. During the tree creation phase, pruning techniques could be adopted to simplify the structure and avoid overfitting, which occurs when the tree fits too closely to the training data and does not generalize well to new data. This question-and-answer structure makes decision trees highly interpretable, allowing users to easily understand the decision process followed by the model [24]. In [25] four different decision tree algorithms (J48, NBtree, Reptree and Simple CART) were compared, in [26] and [27] C4.5, ID3 and CART

decision tree algorithms were used, in [28] decision trees were applied to predict students drop out and identify success factors, in [29] and [30] four decision tree algorithms (C4.5 Decision Tree, ID3 Decision Tree, CART Decision Tree and CHAID Decision Tree) were studied, in [31] the ID3 algorithm was tested on students' performance data, in [32] weighted ID3, a new algorithm based on ID3, is compared with J48 algorithm and Naive Bayes to predict students' performance, in [33] students' interaction data from online learning systems are analyzed by using a decision tree, generated with C4.5 algorithm, and production rules to find symptoms of low performance.

Association rules have also been experimented. Association rules are a data mining technique used to discover interesting relationships between variables within large datasets. Specifically, they identify frequent associations between elements of a set of transactions or events. An association rule is expressed as "if A then B," where A, the antecedent, and B, the consequent, are sets of elements or attributes and the rule indicates that the presence of A is associated with the presence of B with some probability or frequency. The support of an association rule indicates how frequently the association occurs, while the confidence indicates the conditional probability of the presence of B given that A is present [34]. In [35] the Apriori algorithm is used to extract association rules from each class and subsequently analyze the given data to classify the students' performance, while in [36] the Apriori algorithm is used to mine association rules to find out the correlation between courses and the factors that lead to the high or low grades. In [37] an improved version of an existing mining algorithm based on MapReduce is proposed to analyze students' behavioral data and academic performance.

A lot of research works have also compared all these different methods. In [38] various ML methods, decision tree J48, Classification and Regression Tree (CART), JRIP Decision Rules, Gradient Boosting Trees (GBT) and Naive Bayes Classifier (NBC) are tested on students data from an online course. In [39] three selected classification methods, Naive Bayes, Rule Based and Decision Tree, were compared to predict students' performance from students background information. In [40] students' data is analyzed with two rule learners (OneR and JRip), a decision tree classifier (J48), two popular Bayes classifiers (Naive Bayes and BayesNet) and a Nearest Neighbour classifier (IBk) to predict their performance.

In [41] and [42] students' data are evaluated with k-Means and a decision tree to study the main attributes that may affect the performance of students. In [43] association rules, J48 decision tree and EM algorithm were tested on students' academic grades.

A few tests have also been conducted with more complex algorithms like neural networks. Neural networks are computational models inspired by the functioning of the human brain. These models consist of a set of computational units called neurons organized in layers, which are connected to each other through weighted connections. Neural networks are able to learn from data by adapting connection weights in response to the input received. Each neuron receives an input from neurons in previous layers or from external inputs, processes that input through an activation function, and transmits the output to neurons in subsequent layers. The ability of neural networks to learn from data makes them extremely flexible and suitable for a wide range of data processing and analysis problems [44]. In [45] Smooth Support Vector Machine (SSVM) was tested to predict students' performance from psychometric factors like interest and study behavior. In [46] Multilayer Perceptron (MLP), Support Vector Machines (SVM), and Extreme Learning Machine (ELM) algorithms are applied to students' data to predict their performance.

Among all the techniques used for grouping and profiling students' learning paths clustering is one of the most commonly used [47][48][49]. Clustering, the main methodology for performing unsupervised learning, aims to partition data present in an unknown area into clusters so that instances belonging to the same cluster are as similar as possible, while instances belonging to different clusters must be as dissimilar as possible from each other, according to clear and significant similarity and dissimilarity metrics [19].

The following research works used clustering to discover patterns and structures in students performances and evaluate their progresses as well. In [50] a comparison of four clustering algorithms (k-Means, k-Medoids, FCM and EM) was conducted, in [51] k-Means and FCM were employed, in [52] recursive clustering was used, in [53] k-Means combined with the elbow method was observed, also in [54], [55], [56], [57], [58], [59] and [60] k-Means was tested, in [61] k-Means and PROCLUS were performed.

Clustering techniques are also employed on aggregated data from online learning platforms to analyze students' learning behavior and eventually predict their performances. For example in [62] k-Means and Ward's clustering, a hierarchical method, were used, in [63] agglomerative hierarchical clustering was applied, in [64] EM, hierarchical clustering, k-Means and X-Means were performed, in [65] k-Means, DBSCAN and BIRCH were compared, in [66] network analysis and spectral clustering were used.

The following research works also apply clustering methods on different kinds of datasets for the same purposes of profiling students. In [67] k-Means and hierarchical clustering are combined to cluster students based on the mistakes made while using a web-based tool. In [68] spectral clustering and k-Means were performed on a dataset of students' features gathered from two tutoring systems. In [69] k-Means, EM and Farthest First were used to profile students based on competencies, affinities, and demographic attributes. In [70] a two-phase hierarchical clustering algorithm was compared with k-Means and the Farthest First Traversal algorithm on a dataset of students' learning styles. In [71] k-Means and in [72] k-Means, DBSCAN and BIRCH were tested on a dataset of students' performance and other background information. In [73] a pairwise-clustering was performed on a dataset of students' mathematical skills modelled as a dynamic Bayesian network.

Clustering algorithms are not the only method used so far for this purpose, but recently graph theory and network analysis has also begun to be exploited, as in the following works. In [74] social network analysis was employed to analyze the communication that takes place between the students of an online learning course. In [75] a student's knowledge is represented in the form of a dynamic graph of concepts connected when the student succeed in an assessment item containing both concepts and then analyzed. In [76] a Graph-based Exercise- and Knowledge-Aware Learning Network (Graph-EKLN) is proposed to model students' mastery of exercises and knowledge concepts. In [77] Graph-based Knowledge Tracing (GKT), a knowledge tracing method based on GNNs, where the knowledge is structured as a graph and the knowledge tracing task is reformulated as a time-series node-level classification problem in GNN, is proposed. In [78] learning pathways are modelled by networks constructed from

the log data of student interactions from an online learning system, which capture the sequence of reviewing the learning materials by the students enrolled in a course. In [79] a student is modelled as a Bayesian network that stores all the information about him/her so that tutoring systems can use this information to provide personalized instructions. In [80] the student's knowledge is modelled as a dynamic Bayesian network that is able to represent also skill topologies. In [81] Bayesian networks are used for modelling relationships between knowledge items, like question items, for cognitive diagnostic. In [82] Bayesian networks that model students' behavior are studied to detect the learning style of a student. In [83] a model of the students' learning pathways, as a network that captures the time dimension and sequences of the learning events is introduced. In [84] a learning path recommendation model based on a multidimensional knowledge graph framework that separately stores learning objects organized in several classes is proposed. In [85] a few descriptive statistics were computed on networks from students' engagements in two online courses.

Both clustering and graph theory have already proven to be widely useful and efficient in identifying behavioral patterns in various fields and contexts such as human mobility [86], mental disorders [87], social structures [88], human brain [89], human behavior [90][91][92], animal behavior [93], crime detection [94] and customer profiling [95].

Chapter 2

Proposed method

2.1 Representation

This section presents the dataset used and the different ways in which it was chosen to present the data in order to prepare them as input for the next stage of analysis with clustering.

2.1.1 Dataset

To conduct the experiments it was used a subset of 2466 students of the province of Forlì-Cesena that includes pseudonymized information on their demographic, educational, social, economic, and cultural backgrounds, as well as whether they answered correctly to each question of a large-scale assessment test in mathematics held in the school year 2013-2014 when the students under examination were in fifth grade. The test referred to is the INVALSI math test, which that year was composed of 29 questions, some of which included a few sub-questions, for a total of 50 questions. These questions are categorized based on three dimensions, Area, Process and Macro-process, as described in [96]. In particular we can have:

- four areas:
  - (NU) numbers,
  - (SF) space and figures,

  - (RF) relations and functions,
  - (DP) data and prediction;

- seven processes or mathematical skills:
  - (P1) know and master the specific contents of mathematics,
  - (P2) know and use algorithms and procedures,
  - (P3) know different forms of representation and move from one to the other,
  - (P4) solve problems using strategies in different fields,
  - (P5) recognize the measurable nature of objects and phenomena in different contexts and measure quantities,
  - (P6) progressively acquire typical forms of mathematical thought,
  - (P7) use tools, models and representations in quantitative treatment information in the scientific, technological, economic and social fields,
  - (P8) recognize shapes in space and use them for problem solving;

- and three macro-processes:
  - (F) formulating,
  - (I) interpreting,
  - (A) applying.

For the construction of the representations and the following phase of analysis with clustering it was chosen to use only the data regarding the answers to the test, temporarily setting aside the background information of the student that could be taken into account in any future developments to look for correlations between them and the results presented in this work.

2.1.2 Graphs

The first proposed representation of each student is a graph. A graph is a data structure used to represent relationships between objects. Formally, a graph

consists of a set of nodes (or vertices) and a set of edges (or arcs) connecting pairs of nodes. Edges can be directed (with a sense) or undirected (without a sense), depending on whether they represent unidirectional or bidirectional relationships between nodes, and they can be associated with a weight to provide additional information about the relationship between nodes. More precisely, a graph can be defined as an ordered pair $G = (V,E)$, where V is a set of nodes, which can represent entities or data points, E is a set of edges, which represent relationships or connections between nodes [17].

In our domain the graph of a single student has 50 nodes, which correspond to the 50 questions of the INVALSI test, that are connected by directed and weighted edges. As for directionality, an edge always originates from a node indicating a question to which the student has answered correctly and culminates in a node with which it shares at least one of the three dimensions, area, process and macro-process, regardless of its correctness. The number of dimensions shared by a pair of connected nodes i and j is expressed by the weight w_{ij} which assumes a value of 1 to 3 according to the relation $w_{ij} = |I_i \cap I_j|$, where I_i is the set of dimensions of node i. In the figure 2.1 we can see a detail of an example of a graph, the upper right part, and in 2.2 we can see the full example.

2.1.3 Spanning trees

The second representation chosen to depict the test answers of a student is a spanning tree. The spanning tree of a connected graph is the minimal subset of edges that forms a tree that connects all the nodes of the graph, without the creation of cycles. More formally, the spanning tree of a connected graph $G = (V,E)$ is defined as a subset of edges $T \subseteq E$, such that: T is a tree, i.e. it is a connected, acyclic graph, and T contains all nodes of G, i.e. for every node v in V, there exists a path in T that connects v to all other nodes [17].

This representation is constructed from the one described in 2.1.2 to eliminate the cycles [17] in the previous one, so the same criteria of directionality and weight of the edges are maintained, as well as the same nodes. Trees are built using the depth-first strategy, i.e. you completely explore all the nodes of the current subtree before moving on to the sibling nodes [18], but in two slightly different

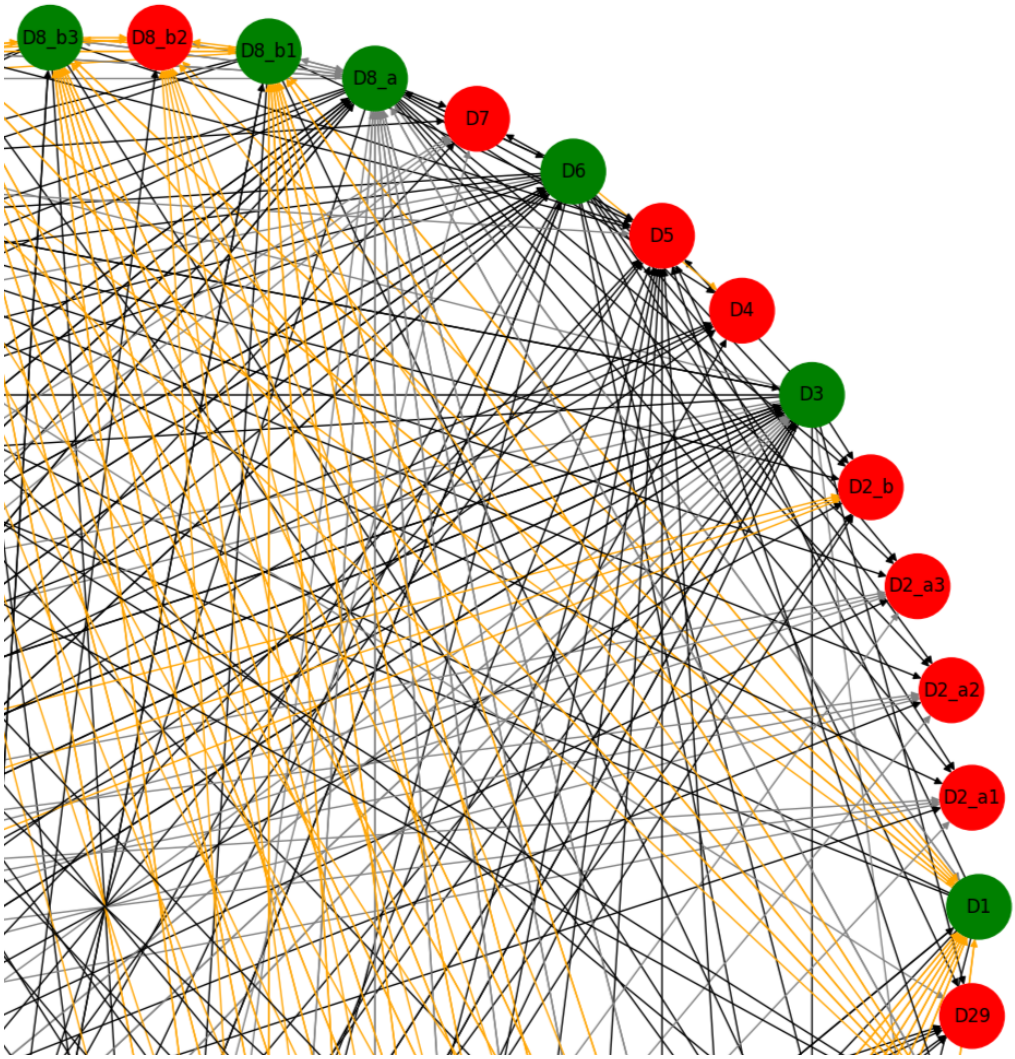

Figure 2.1: Detail of figure 2.2. Upper right part of an example of a student graph with green nodes representing correct answers and red nodes representing wrong answers. The black edges have weight equal to 1, the grey edges equal to 2, and the orange edges equal to 3.

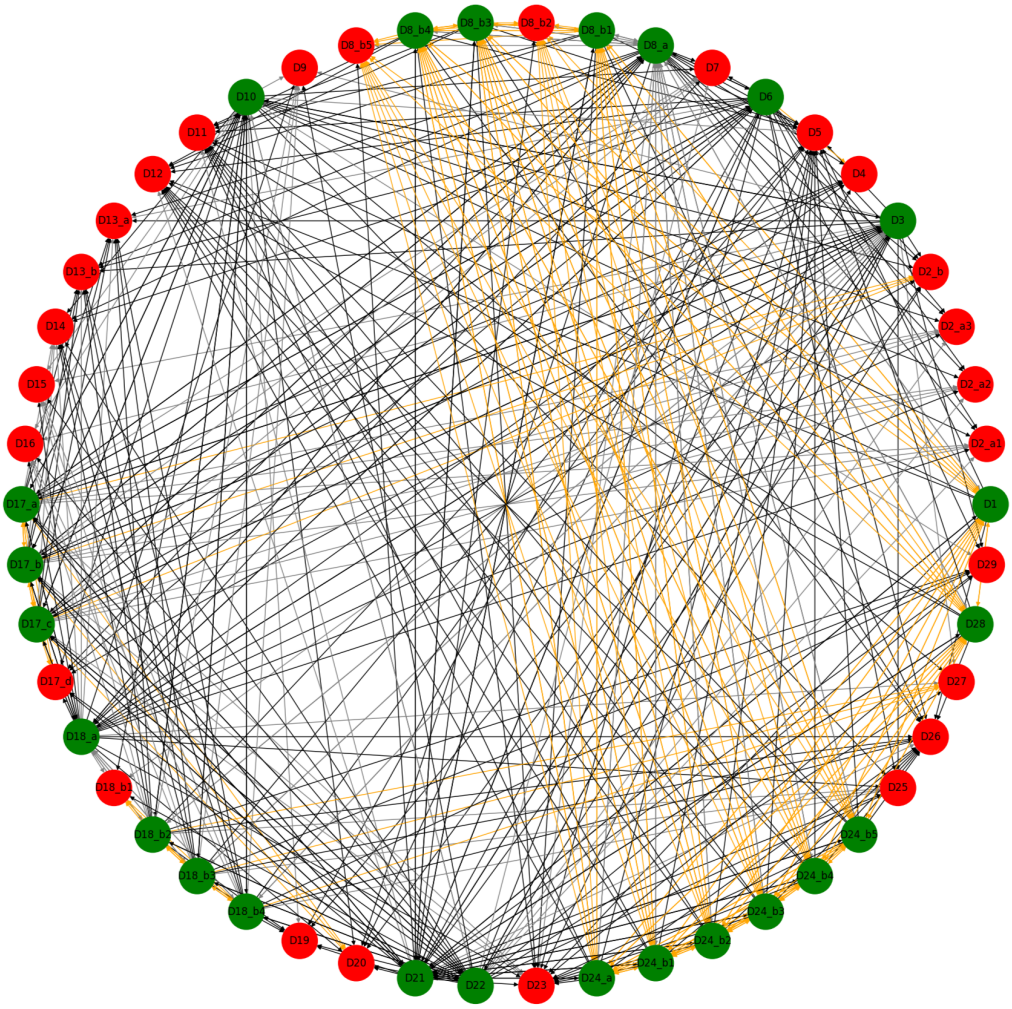

Figure 2.2: Example of a student graph with green nodes representing correct answers and red nodes representing wrong answers. The black edges have weight equal to 1, the grey edges equal to 2, and the orange edges equal to 3.

versions: in the first version of trees, called deep spanning trees, every time you explore a node, only one child node is added and you continue exploring from it, while in the second version, the shallow spanning trees, every time you explore a node, all its child nodes are added to the tree. A common element between the two versions which is fundamental for the construction of these trees is the strategy for choosing the next node to be explored, which in the case of deep spanning trees coincides with the child node to be inserted in the tree and in the case of shallow spanning trees is the first node to be explored among the newly inserted child nodes, which is based on the difficulty of the question. In fact, the easiest node is always chosen, which is the node that received the highest number of correct answers among the nodes considered with respect to the reference dataset. In this way we get two representations without cycles, but to preserve the indication of the correctness of the answer to a question, when a node that represents a correct answer but has no child nodes is added to the tree, a fictitious child node is connected to it. In figure 2.3 we find a detail, the middle part specifically, of the representation as deep spanning tree of the same student in fig 2.2 and in 2.4 the full example, while in figure 2.5 we see the details of the representation as shallow spanning tree, again the middle part, and in 2.6 the full example.

2.1.4 Metrics

The last representation considered provides the calculation of a set of global metrics on the three representations just proposed, the graphs and the two types of spanning trees. Below are the metrics calculated on all three representations with the definition and interpretation based on our reference domain:

- **Average out-degree** calculates the average of the edges leaving the nodes taking also into account the weights. In our domain it gives us an indication of how many dimensions on average a node representing a correct response shares with other nodes.

- **Compactness** measures the percentage of pairs of nodes that can be reached from a path of any length, but the paths connecting the nodes are weighted inversely according to their length, therefore it measures how easily things

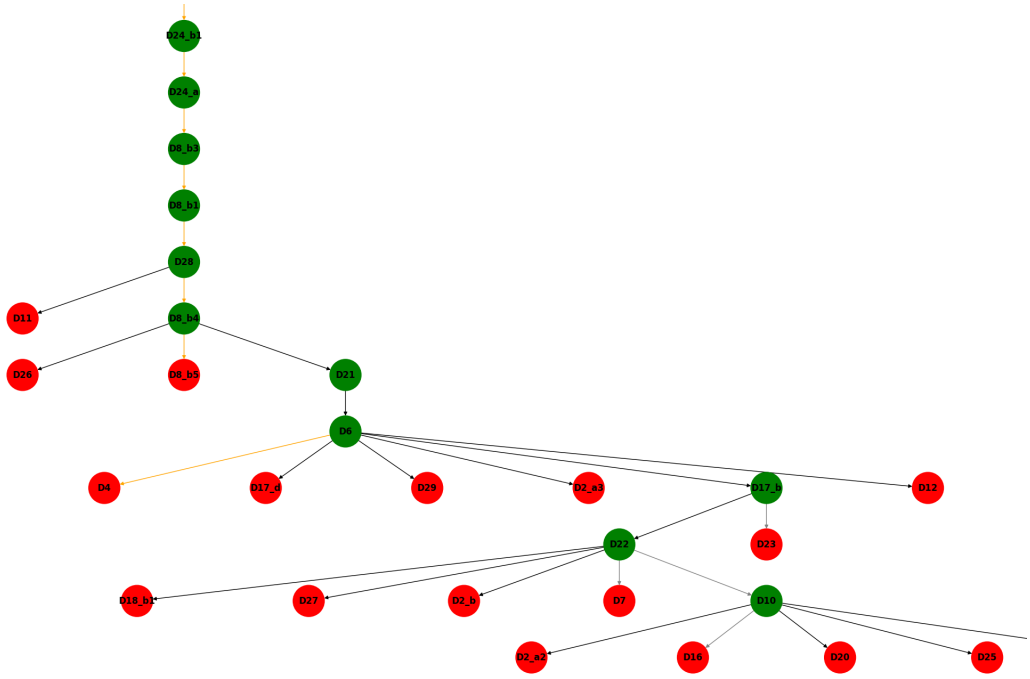

Figure 2.3: Detail of figure 2.4. Middle part of an example of a student deep spanning tree with green nodes representing correct answers and red nodes representing wrong answers. The black edges have weight equal to 1, the grey edges equal to 2, and the orange edges equal to 3.

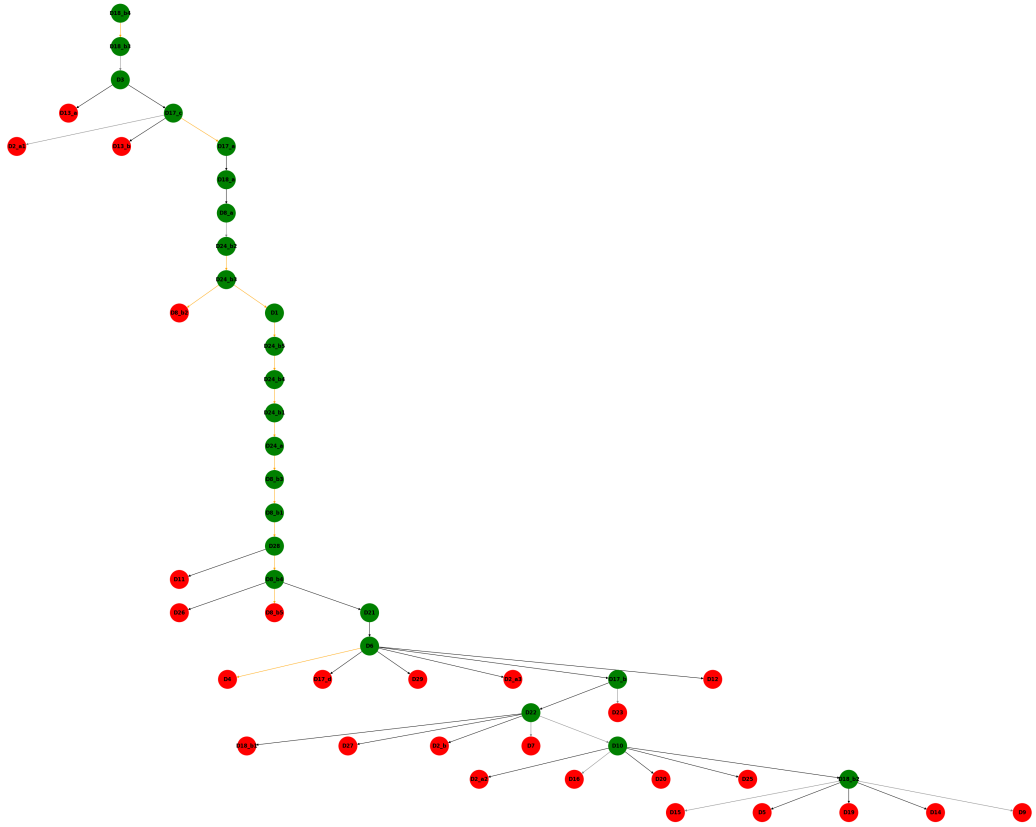

Figure 2.4: Example of a student deep spanning tree with green nodes representing correct answers and red nodes representing wrong answers. The black edges have weight equal to 1, the grey edges equal to 2, and the orange edges equal to 3.

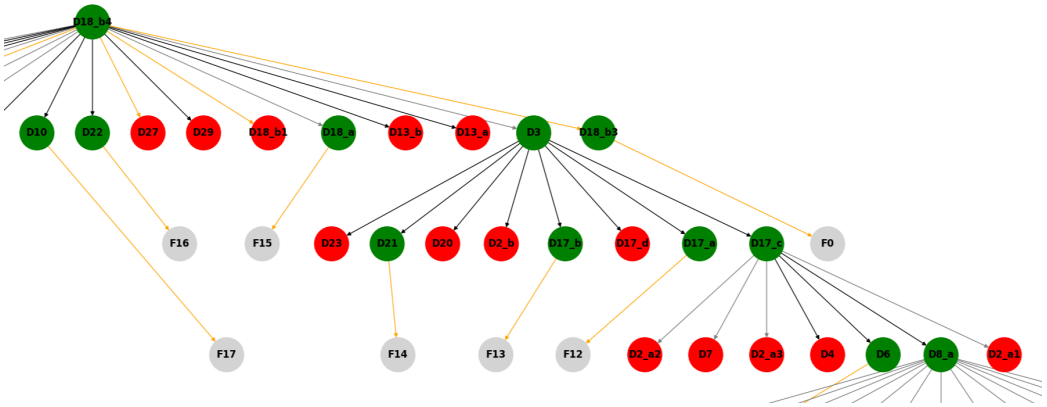

Figure 2.5: Detail of figure 2.6. Middle part of an example of a student shallow spanning tree with green nodes representing correct answers, red nodes representing wrong answers and light gray nodes representing fictitious nodes. The black edges have weight equal to 1, the grey edges equal to 2, and the orange edges equal to 3.

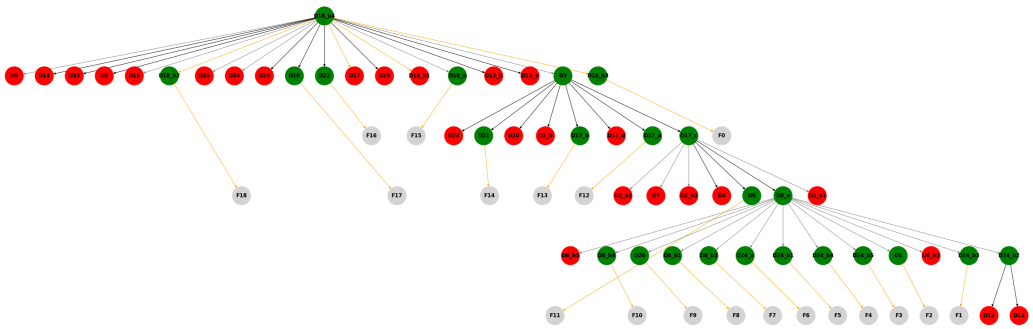

Figure 2.6: Example of a student shallow spanning tree with green nodes representing correct answers, red nodes representing wrong answers and light gray nodes representing fictitious nodes. The black edges have weight equal to 1, the grey edges equal to 2, and the orange edges equal to 3.

can cross it, including disconnected components. In our domain a very high value indicates that the nodes reach each other with short paths and consequently that the student has sufficient mastery of the various possible dimensions, except in the case of deep spanning trees where a compact graph is an index of many errors, while in shallow spanning trees it indicates that the student has good mastery of some dimensions and poor mastery of others.

- **Average closeness centrality** calculates the mean reciprocal of the mean shortest path distance to a node over all reachable nodes n-1. In our domain it indicates how much the student has mastered the required skills, except in spanning trees where it's the other way around.

- **Average betweenness centrality** calculates the average of the sums of the fraction of the shortest paths among all pairs passing through the node. In our domain a very high value indicates that there are many critical nodes that are fundamental to a proper assessment of the student's skills, in particular in spanning trees where nodes are ordered in growing order of difficulty.

- **Average edge betweenness centrality** calculates the average of the sums of the fraction of the shortest paths among all the pairs crossing each edge. In our domain, the higher the value, the more there are many critical edges which, if removed, would make the student's level of competence lower, and therefore indicates that the student's competence, even with a decent mastery, is not very consolidated, in particular in spanning trees where nodes are ordered in growing order of difficulty.

- **Density** calculates how much a graph is "filled" in relation to the maximum number of possible edges, i.e. those of a graph where all the answers are correct, and can vary from 0, "spreadly" connected, to 1, densely connected. In our domain, the higher the density, the more it means that the test contains fewer errors.

- **S-metric** calculates the sum of the degree products, calculated as the sum of in-degree and out-degree for direct graphs, of the two nodes connected by

an edge for each edge of the graph, without taking into account the weights of the edges. It gives an indication of how much the graph is scale-free, that is, how much it follows the power law distribution, that is, if there are few nodes with very high degrees and many nodes with low degrees. In our domain it indicates how robust the graph is to the removal of nodes and it gives an indication of how much the test taken by the student is able to reflect his knowledge, since even after removing some questions from the test we could still correctly verify the level of learning of the student.

- **Number of isolates** calculates the number of isolated nodes. In our domain it can only happen if a node is wrong and has no dimension in common with the others, or if all the other nodes that share one or more dimensions in common with it are also wrong, ergo the three dimensions of the node have been mistaken throughout the test.

- **Number of weakly connected components** calculates the number of weakly connected components, i.e. the maximum subsets of the nodes, where there is an indirect path between each pair of nodes. In our domain, in the case of graphs it tells us the number of separate subgraphs, while in the case of spanning trees it tells us the number of trees if there are more than one tree.

The following metrics were also calculated for the graphs only:

- **Average in-degree** calculates the average of the incoming edges from the nodes, also taking into account the weights. In our domain it gives us an indication of how many dimensions on average a node, which can be correct or not, shares with the nodes corresponding to correct answers.

- **Degree assortativity coefficient** measures the tendency of nodes with similar in-degrees to connect to each other and the same for nodes with similar out-degrees. In our domain it indicates how much answers with the same dimensions tend to be connected to each other because they are both correct and consequently how much a student actually has a good mastery of the various skills.

- **Global reaching centrality** computes the average over all nodes of the difference between the local reaching centrality of the node, which is the proportion of other nodes reachable from that node, and the largest local reaching centrality in the graph. In our domain it indicates the ability of the nodes to reach other nodes compared to the most influential node, so a high value indicates the percentage of wrong answers in the test, since the wrong nodes have a local centrality equal to 0, and the correct nodes equal to 1.

- **Flow hierarchy** calculates the fraction of edges that does not participate in a cycle. In our domain, since directionality always starts from nodes representing correct answers, a high value indicates that there is a hierarchical structure in the graph and therefore that some nodes have a more important role than others and consequently their dimensions are more positively relevant for the student.

- **Transitivity** calculates the fraction of all possible triangles in the graph, i.e. measures the probability that if node A is connected to node B and node B is connected to node C, then also node A is connected to node C. In our domain, a high value suggests the presence of cyclic structures in the graph, i.e. that the student is very good at most of the questions' dimensions.

- **Average clustering** is the average of local clusterings, that is the fraction of triangles that actually exist on all possible triangles in its neighborhood. In our domain a high value means that nodes tend to form clusters with other nodes that share at least one dimension out of three, so clusters can represent test questions that are related to each other based on the dimensions being considered.

- **Overall reciprocity** measures the tendency for pairs of nodes to be reciprocal. In our domain, since pairs of nodes are reciprocal only when both nodes are correct, it is the measure of how much two nodes that are related by some dimensions are also correct.

- **Average node connectivity** computes the average of local node connectivity between all node pairs, i.e. the average number of nodes to be removed

to disconnect the graph or reduce its connectivity. In our domain, a high value indicates that, even if we removed some questions from the test, we would still be able to adequately evaluate all the questions' dimensions.

- **Edge connectivity** calculates the minimum number of edges that must be removed to disconnect the graph or make it trivial, so the higher the value, the more resistant the graph is to removing edges and stays connected. In our domain, a high value indicates that, even if we remove some connections, for example we might assume more mistakes are made by the student, the graph remains connected, so the student doesn't have any particular difficulty in one of the three dimensions.

- **Number of strongly connected components** calculates the number of strongly connected components, i.e. subsets of nodes where there is a direct path between each of the nodes. In our domain, strongly connected components can be formed by a single node in the case of nodes that correspond to wrong answers, while all components with more than one node contain only nodes that represent correct answers, so the fewer strongly connected components there are in a graph, the fewer mistakes the student will make in the test.

- **Connectedness** measures the percentage of pairs of nodes that can be reached from a path of any length or, alternatively, the percentage of pairs of nodes that are in the same component. In our domain a high value indicates that most answers are strongly connected and can be reached by any other answer through a series of connections, indicating a significant consistency between the student's answers with respect to the three dimensions of the questions.

Only for spanning trees the following metrics have been calculated:

- **Size** is the sum of the weights of the edges in the spanning tree. In our domain it indicates that the higher the sum, the more the nodes connected to each other share many dimensions.

- **Breadth (Max node out-degree)** calculates the maximum number of outgoing edges on a single node in the spanning tree, which corresponds to the

width of the tree. In our domain in the case of shallow spanning trees a very large width, which corresponds to a node with many children, indicates that the three dimensions of the node are very common in the questions of this test, while in the case of deep spanning trees if the width is large means that the tree has not been developed in depth because of too many incorrect nodes.

- **Load balance** calculates the average difference between the number of incoming edges and the number of outgoing edges taking into account the weights of the edges. In our domain it gives interesting indications about how much parent nodes and child nodes have in common.

- **Height (DAG longest path length)** calculates the maximum distance from the tree root to the leaves. In our domain, in the case of deep spanning trees, if the height is limited, it means that there are many wrong nodes that prevent the deep development of the tree, while in the case of shallow deep trees, if the height is very high, it means that the easiest and correct nodes already inserted and then explored have the least frequent dimensions in the test.

2.2 Analysis

All the representations described in 2.1, the graphs, the two versions of spanning trees and the metrics calculated on the previous three, were then analyzed with three clustering algorithms (k-Means, DBSCAN and Gaussian Mixture) to identify elements that characterize the various clusters, so that a domain expert could then use them to extract information on the learning states of the students.

As for graphs and spanning trees, these representations were prepared for clustering by transforming them into linearized adjacency matrices. In the case of a directed and weighted graph such as ours, the adjacency matrix is a tabular representation of the graph that shows the relationship between nodes through their edges, indicating the weights of the associated edges. For a directed and weighted graph with n nodes, its adjacency matrix is an $n \times n$ square matrix, where the element in row i and column j represents the weight of the edge from node i to node

j (if any). If no edge exists between nodes i and j, the corresponding element in the matrix will usually be represented by a special value, in our case 0, to indicate the absence of the edge. More formally, given an adjacency matrix A of a directed, weighted graph, the elements a_{ij} of the matrix are defined as follows: a_{ij} represents the weight of the edge from node i to node j, if there is no edge between nodes i and j, a_{ij} can be represented by a special value such as 0 to indicate the absence of the edge. The matrices were then linearized to further simplify the clustering input, i.e. each matrix was transformed from table form $n \times n$ to linear form $1 \times n^2$ by concatenating the rows. At the end, the input dataset of these representations was composed of 2466 rows, one for each student, and the columns (or features or dimensions) represent the connections between each pair of nodes with the associated weight.

Before proceeding with the clustering, the Principal Component Analysis (PCA) was performed on the graphs and on the two types of spanning trees, but not on the metrics, in order to reduce the dimensionality of the input data of the clustering methods, since in the linearized adjacency matrices of these representations it was very high, while keeping as much as possible the variance between the features. PCA is a dimensionality reduction technique that is used to transform a multi-dimensional dataset with a lot of features into a smaller dataset, while retaining most of the information contained in the original data [97]. The main goal of PCA is to transform a high-dimensional dataset into a new coordinate system, known as ”principal components,” so that most of the variance in the data is explained by the first principal components. PCA identifies a set of new variables, the ”principal components” precisely, which are linear combinations of the original variables. These principal components are ordered by importance, so that the first principal component explains the maximum variance in the data, the second explains the second maximum variance, and so on. After calculating the principal components, the original data can be projected into the new principal component space in a way that reduces the dimensionality of the data while maintaining the maximum amount of variance explained. A critical step in PCA is the selection of the number of principal components to be retained. This decision depends on the goal of the analysis and the amount of variance you want to retain in the reduced data, in our case, for example, we want to keep as much variability in the data as

possible.

In the case of metrics prior to applying clustering algorithms, metrics with variance equal to 0 and metrics with correlation equal to 1 with other metrics already present have been removed, so the input dataset of metrics was composed of 2466 rows, one for each student, and the columns are the calculated metrics.

The quality of the clustering partition was then assessed using the Silhouette Score, a measure that evaluates the cohesion and separation of the clusters by providing an indication of the overall quality of the data splitting into clusters. For each sample it calculates how similar its cluster is to other clusters as the difference between the mean intra-cluster distance and the mean nearest-cluster distance normalized for the maximum of the two:

$$Silhouette\ Score = \frac{b-a}{max(a,b)},$$

where a is the mean intra-cluster distance, i.e. the average distance between a sample and the other samples in the same cluster, and b is the mean nearest-cluster distance, i.e. the average distance between a sample and samples from the nearest cluster not belonging to the same cluster. For a, a low value of this measure indicates that the sample is very close to the others in the cluster, suggesting good cohesion in the clusters, while for b, a high value of this measure indicates good separation between clusters. The silhouette score ranges from -1 to 1, with 1 representing the best possible subdivision, -1 indicating that the sample has been assigned to the wrong cluster, and 0 indicating that the sample is close to the edge between two clusters. In general, a silhouette score closer to 1 suggests that the clustering of data is appropriate, while values close to 0 or negative may indicate that clusters overlap or that samples have been misallocated.

2.2.1 k-Means

k-Means is a clustering algorithm proposed by J. MacQueen in 1967 and it is used to partition data into **k** groups efficiently in terms of intra-class variance [20]. Its goal is to minimize the sum of squares of distances between data points

and centroids, called inertia or Sum of Squared Errors (SSE):

$$SSE = \sum_{i=1}^{n} \underset{c_j \in C}{min}(||x_i - c_j||^2),$$

where n is the number of samples, C is the set of **k** clusters, c_j is the centroid of the j-th cluster, x_i is the i-th sample and $||x_i - c_j||$ is the Euclidean distance between sample x_i and centroid c_j.

This method processes the partition into **k** groups starting with **k** initial points called centroids and assigning each element of the dataset to the cluster of the nearest centroid according to the Euclidean distance. The Euclidean distance is a measure of distance between two points in a Euclidean space, which is a geometric space characterized by the properties of Euclidean geometry. In a Euclidean space, each point is represented by a coordinate vector, and the distance between two points is calculated using the Pythagorean theorem. In other words, the Euclidean distance is the length of the segment connecting the two points in their space. It is an intuitive measure of ”distance” that reflects the length of the shortest path between two points in Euclidean space. The initial centroids are determined using k-Means++, which selects initial cluster centroids using sampling based on an empirical probability distribution of the points' contribution to the overall inertia to speed up convergence. Once all the points have been assigned to a cluster, the centroids are recalculated as the average of the points belonging to each cluster and the points distributed again between the clusters based on the Euclidean distance. This optimization process is repeated until the centroids stop changing their position and therefore until a satisfactory partition is found.

The primary difficulty of this algorithm is to determine the parameter **k**, i.e. in how many clusters to partition the data [98]. A very simple but very popular technique is the elbow method [53][99], which involves locating the point on an elbow graph where the change in the SSE starts to decrease significantly more slowly as the number of clusters increases. In fact, going beyond this point, i.e. choosing larger **k** values, would not improve the model's ability to explain variance in the data. If, in addition to the elbow graph, we also take into account the changes in the silhouette score as a function of the number of clusters, we can make an even more informed choice of the **k** parameter.

2.2.2 DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm proposed by Ester et al. [21], which is able to identify arbitrarily shaped clusters in a high-dimensional data space, separated by low-density regions. Initially, for each point in the dataset, the algorithm calculates the number of points within a distance ε and if this number exceeds or equals a previously set **min_samples** threshold, the point is considered a core point. For each core point not yet assigned to a cluster, the points reachable from the core point are explored, which are then assigned to the same cluster as the core point, and if a core point is reachable from multiple clusters, it is assigned to the first cluster found. Non-core points that are not reachable from any other core point are considered ”boundary points” and are assigned to the cluster of the nearest core point connected by a core point path. Points that are neither core points nor boundary points are considered ”noise points” and are not assigned to any cluster.

The two fundamental parameters that characterize this algorithm are the already mentioned ε and **min_samples** and in [100] it is suggested to try the following approach: set the last parameter **min_samples** as the double of the features of the dataset as suggested in [101], while for the first parameter the heuristic proposed in [21] is revisited and the distance from $2 * dim - 1$ nearest neighbors is computed as suggested again in [101], where dim is the number of features in the dataset. According to this method we have to plot the k-dist graph, which is the graph of the k-nearest-neighbor distances computed for each point and ordered from largest to smallest. The optimal value, or the range of optimal values to test, will be where the graph will form a sort of elbow.

2.2.3 Gaussian Mixture Model

The Gaussian Mixture Model (GMM) is a probability clustering algorithm that assumes that data is generated by a set of Gaussian distributions (or components), each characterized by a centroid (mean) and a covariance matrix. The Gaussian distribution is also known as the normal distribution or bell distribution because of its symmetrical bell shape around its mean, with most values concentrated near the mean and a gradual decrease in probability density as one moves away from

it. The Gaussian distribution is completely determined by two parameters: mean (μ), which represents the center of the distribution and indicates the expected value of the random variable, and the standard deviation (σ), which represents the dispersion of the data around the mean and determines how widely the data are distributed around the mean, so a smaller standard deviation indicates a more concentrated distribution around the mean, while a larger standard deviation indicates a more dispersed distribution. The covariance matrix is a statistical measure that provides information about the covariance relationship between variables in a multivariate data set. It provides a measure of the joint variability of two or more variables and the direction of this variability.

In the model, initially, the parameters of the Gaussian components, including the centroids and covariance matrices, are randomly initialized using k-Means. For each data point in the dataset, the probabilities of belonging to each Gaussian component are calculated using the multivariate Gaussian probability density formula for each component, which are then used to update the parameters of the Gaussian components, including centroids and covariance matrices, using the Maximum Likelihood Estimation method, which is a technique used to estimate the parameters of a statistical model, maximizing the likelihood of the observed data under the assumption that the model is correct or, in other words, it tries to find those parameter values that make the observation of the data we have most likely. This process is iterated until the parameters converge or until a preset maximum number of iterations is reached [22].

This algorithm is very flexible and can model clusters of various shapes and sizes, but it also depends on the choice of the number of components **n_components** to use and the type of covariance matrix **covariance_type** between:

- full: each component has its own general covariance matrix,
- tied: all components share the same general covariance matrix,
- diag: each component has its own diagonal covariance matrix,
- spherical: each component has its own single variance.

To choose the best ones, you can search through the various combinations of the two parameters and choose the one with the lowest BIC score, as this indicates a

good compromise between adaptability to the data and complexity of the model. The Bayesian Information Criterion (BIC) is defined as:

$$BIC = -2log(L) + dlog(n),$$

where L is the maximum likelihood of the model to the data, d is the number of parameters of the model, and n is the number of elements of the dataset [102].

2.2.4 Metrics selection

To try to give meaning to the clustering results applied to the metrics calculated on the three representations, the intra-cluster mean and variance were evaluated. Specifically, for each metric and each cluster, it was checked that the open intervals, i.e. where extremes are excluded, created by adding and subtracting the variance of each cluster from its mean, did not overlap for more than the extreme values of the intervals, i.e. whether the intervals were at most adjacent. It was also checked if by chance the variance of the intra-clusters variances was greater than 1, which would mean that among any clusters identified some consistently separate the data while at least one overlaps all the others.

Chapter 3

Experimental results

3.1 Technologies

The experiments were conducted using Google Colab and the Python programming language (v. 3.10.12). The libraries used are Pandas (v. 1.5.3), Numpy (v. 1.25.2), Networkx (v. 3.2.1), Matplotlib (v. 3.7.1) and Scikit-learn (v. 1.2.2).

3.2 Results

Below are the results of all the experiments performed with the three clustering algorithms (k-Means, DBSCAN and Gaussian Mixture Model) on each of the six representations, i.e. the graphs, the deep spanning trees, the shallow spanning trees and the metrics applied to the three representations just mentioned.

Regarding the application of PCA on the linearized adjacency matrices of the graphs and the spanning trees, 0.995 was chosen as the value of the parameter **n_components** to maintain 99.5% of the variability. As for the evaluation range of the elbow method to decide the **k** value of k-Means, numbers of **k** clusters between 2 and 10 are tested for all representations. For all three algorithms, the parameter **random_state** has been set to 42 for the reproducibility of the experiments.

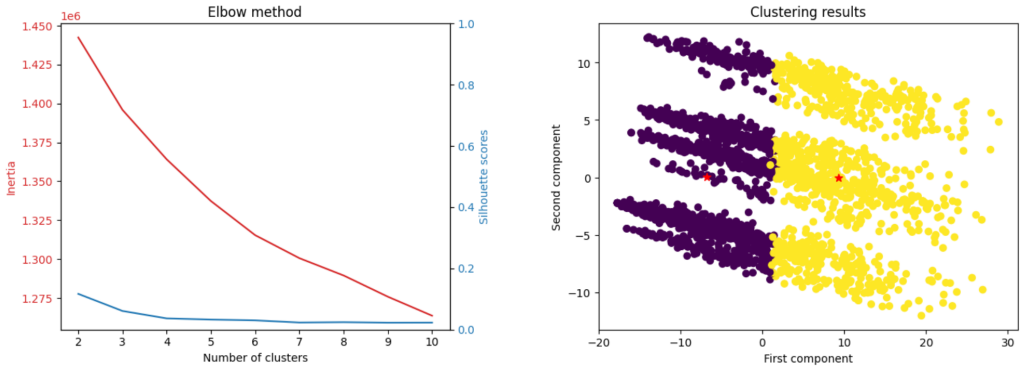

Figure 3.1: On the left the changes, based on the number of clusters, of inertia in red and silhouette scores in light blue in the elbow graph produced by the elbow method. On the right the result of the clustering performed with k-Means plotted on the two most significant components of the dataset with the centroids of each cluster identified by red stars.

3.2.1 Graphs

For k-Means the elbow method suggests using a value of **k** equal to 2, so the algorithm partitions the data in two clusters. Figure 3.1 shows the graph produced by the elbow method and the plot of the clustering results on the two main dimensions of the data.

The k-dist graph computed before DBSCAN suggests a value of ε equal to 28, so the model partitions the data in one cluster and identifies 522 noise points. Figure 3.2 shows the k-dist graph and the plot of the clustering results on the two main data dimensions.

The search for the best parameter combinations for the Gaussian Mixture model application suggests 3 as **n_components** and ”full” as **covariance_type**, so the model partitions the data in three clusters. Figure 3.3 shows the plot of the clustering results on the two main data dimensions.

The table 3.1 shows the silhouette scores of the three clustering algorithms applied. As we can see DBSCAN gets the highest score among the three models, however, all three do not produce a particularly satisfactory score.

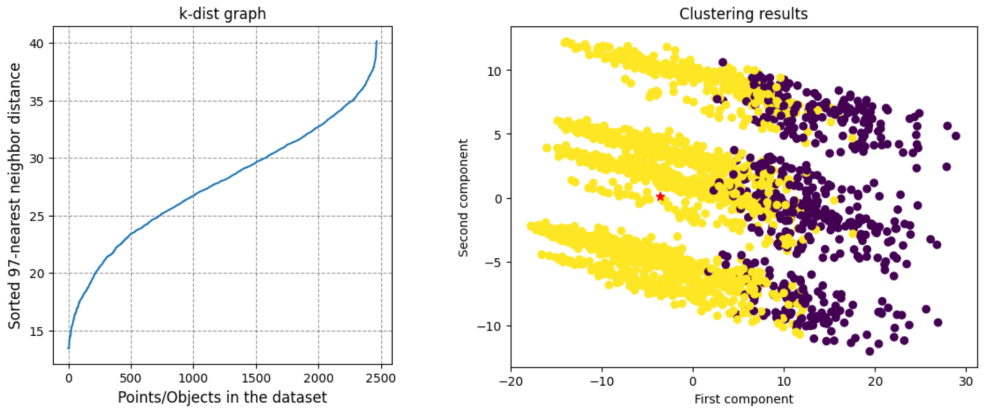

Figure 3.2: On the left the k-dist graph, that plots the changes of the k-nearest-neighbor distances computed for each point and ordered from largest to smallest. On the right the result of the clustering performed with DBSCAN plotted on the two most significant components of the dataset with the centroid of the cluster identified by a red star and the noise points identified by purple points.

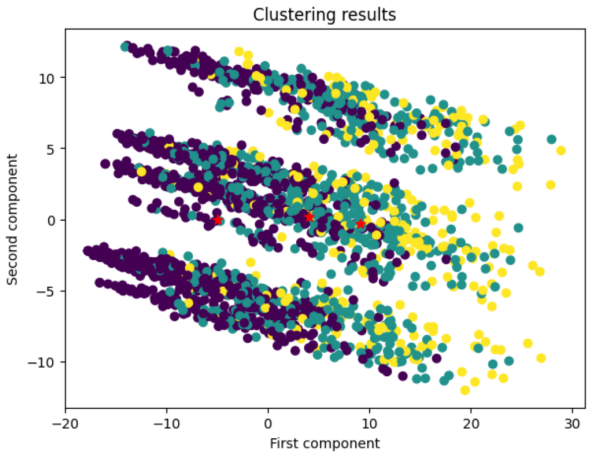

Figure 3.3: The result of the clustering performed with Gaussian Mixture model plotted on the two most significant components of the dataset with the centroids of each cluster identified by red stars.

	k-Means	DBSCAN	Gaussian Mixture Model
Silhouette score	0.11679	0.15517	0.04883

Table 3.1: Silhouette scores of clustering applied on graphs.

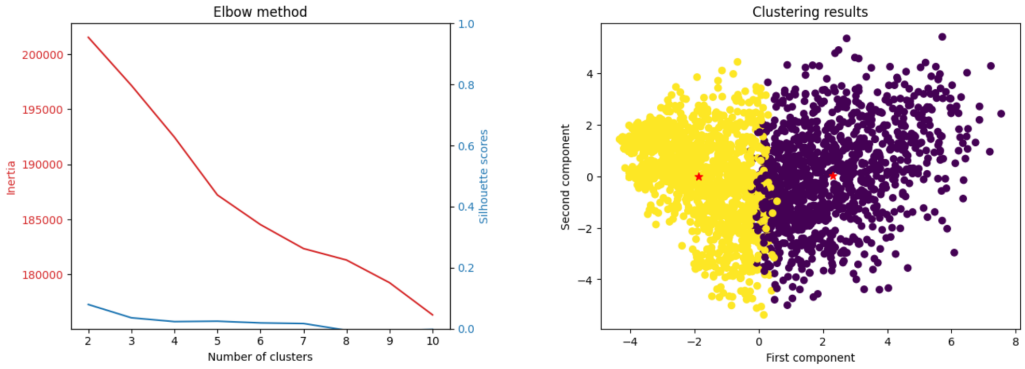

Figure 3.4: On the left the changes, based on the number of clusters, of inertia in red and silhouette scores in light blue in the elbow graph produced by the elbow method. On the right the result of the clustering performed with k-Means plotted on the two most significant components of the dataset with the centroids of each cluster identified by red stars.

3.2.2 Deep spanning trees

For k-Means the elbow method suggests using a value of **k** equal to 2, so the algorithm partitions the data in two clusters. Figure 3.4 shows the graph produced by the elbow method and the plot of the clustering results on the two main dimensions of the data.

The k-dist graph computed before DBSCAN suggests a value of ε equal to 10, so the model partitions the data in one cluster and identifies 531 noise points. Figure 3.5 shows the k-dist graph and the plot of the clustering results on the two main data dimensions.

The search for the best parameter combinations for the Gaussian Mixture model application suggests 4 as **n_components** and ”diag” as **covariance_type**, so the model partitions the data in four clusters. Figure 3.6 shows the plot of the

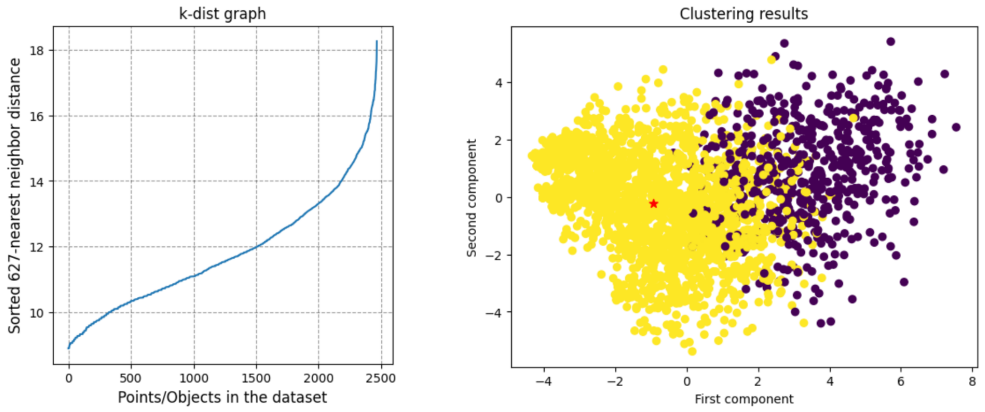

Figure 3.5: On the left the k-dist graph, that plots the changes of the k-nearest-neighbor distances computed for each point and ordered from largest to smallest. On the right the result of the clustering performed with DBSCAN plotted on the two most significant components of the dataset with the centroid of the cluster identified by a red star and the noise points identified by purple points.

	k-Means	DBSCAN	Gaussian Mixture Model
Silhouette score	0.07950	0.14599	-0.07165

Table 3.2: Silhouette scores of clustering applied on deep spanning trees.

clustering results on the two main data dimensions.

The table 3.2 shows the silhouette scores of the three clustering algorithms applied. As we can see DBSCAN gets the highest score among the three models again, however again, all three do not produce a particularly satisfactory score.

3.2.3 Shallow spanning trees

For k-Means the elbow method suggests using a value of **k** equal to 4, so the algorithm partitions the data in four clusters. Figure 3.7 shows the graph produced by the elbow method and the plot of the clustering results on the two main dimensions of the data.

The k-dist graph computed before DBSCAN suggests a value of ε equal to

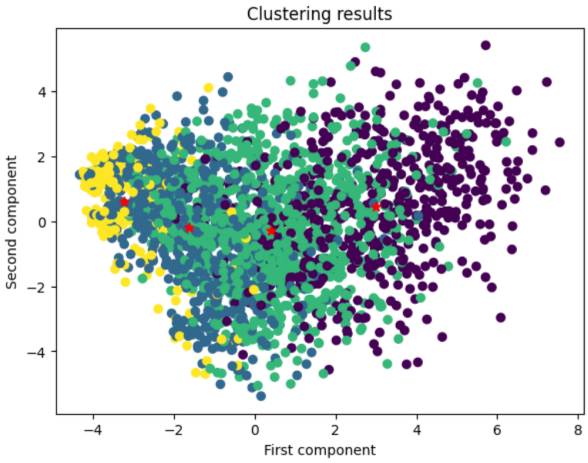

Figure 3.6: The result of the clustering performed with Gaussian Mixture model plotted on the two most significant components of the dataset with the centroids of each cluster identified by red stars.

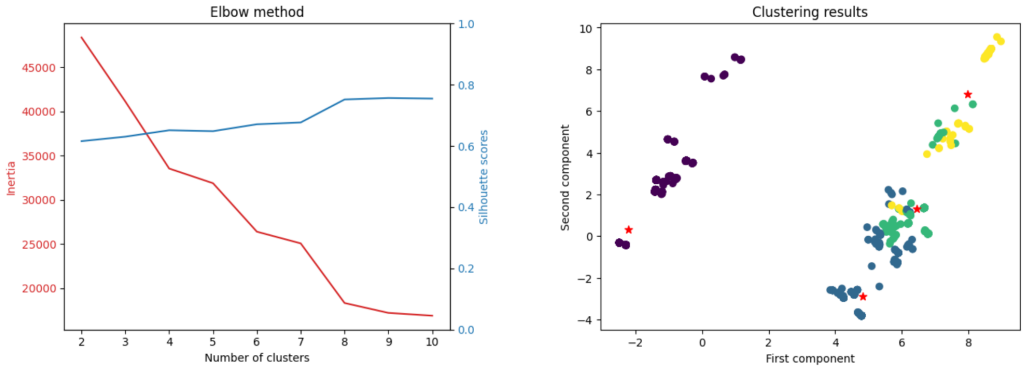

Figure 3.7: On the left the changes, based on the number of clusters, of inertia in red and silhouette scores in light blue in the elbow graph produced by the elbow method. On the right the result of the clustering performed with k-Means plotted on the two most significant components of the dataset with the centroids of each cluster identified by red stars.

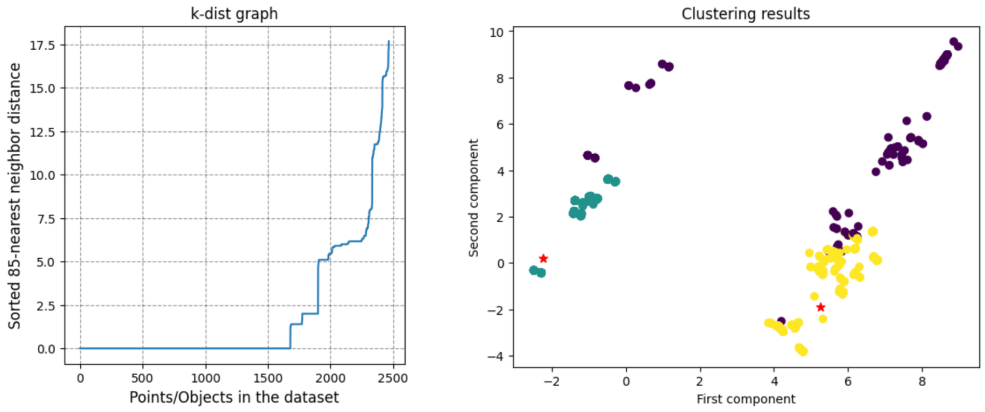

Figure 3.8: On the left the k-dist graph, that plots the changes of the k-nearest-neighbor distances computed for each point and ordered from largest to smallest. On the right the result of the clustering performed with DBSCAN plotted on the two most significant components of the dataset with the centroids of each cluster identified by red stars and the noise points identified by purple points and the noise points identified by purple points.

	k-Means	DBSCAN	Gaussian Mixture Model
Silhouette score	0.65136	0.64054	0.63353

Table 3.3: Silhouette scores of clustering applied on shallow spanning trees.

7, so the model partitions the data in two clusters and identifies 133 noise points. Figure 3.8 shows the k-dist graph and the plot of the clustering results on the two main data dimensions.

The search for the best parameter combinations for the Gaussian Mixture model application suggests 6 as **n_components** and "spherical" as **covariance_type**, so the model partitions the data in six clusters. Figure 3.9 shows the plot of the clustering results on the two main data dimensions.

The table 3.3 shows the silhouette scores of the three clustering algorithms applied. As we can see k-Means gets the highest score among the three models, and all three produce a particularly satisfactory score.

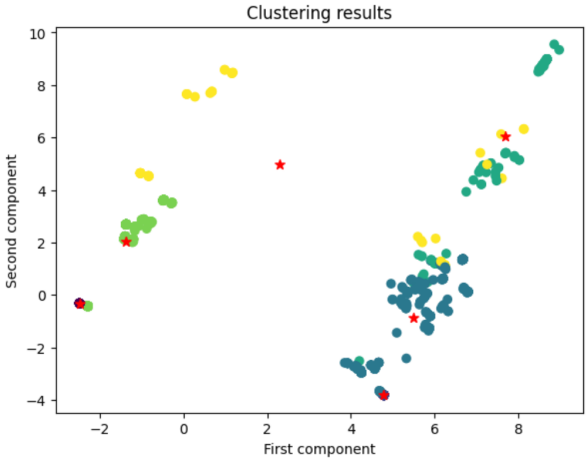

Figure 3.9: The result of the clustering performed with Gaussian Mixture model plotted on the two most significant components of the dataset with the centroids of each cluster identified by red stars.

3.2.4 Metrics on graphs

First, we report the mean and the variance of the metrics calculated on the graphs in table 3.4.

For k-Means the elbow method suggests using a value of **k** equal to 3, so the algorithm partitions the data in three clusters. Figure 3.10 shows the graph produced by the elbow method and the plot of the clustering results on the two main dimensions of the data. Moreover, from the process of selecting the characteristic metrics of the various clusters based on mean and variance, as explained in the 2.2.4 section, we get the metrics:

- density,
- s-metric,
- average clustering,
- number of strongly connected components,
- transitivity,

	Mean	Variance
Average in-degree	25.289	29.965
Average out-degree	25.289	29.965
Compactness	1.339	0.075
Average closeness centrality	0.435	0.012
Average betweenness centrality	0.012	0.0
Average edge betweenness centrality	0.002	0.0
Density	0.694	0.025
S-metric	939852.250	2.175635e+11
Number of isolates	0.0	0.0
Number of weakly connected components	1.0	0.0
Number of strongly connected components	16.032	61.767
Connectedness	0.509	0.046
Average node connectivity	6.696	9.042
Edge connectivity	0.048	0.818
Global reaching centrality	0.393	0.017
Average clustering	0.689	0.001
Overall reciprocity	0.689	0.026
Transitivity	0.515	0.013
Flow hierarchy	0.290	0.023
Degree assortativity coefficient	0.394	0.021

Table 3.4: Metrics computed on graphs.

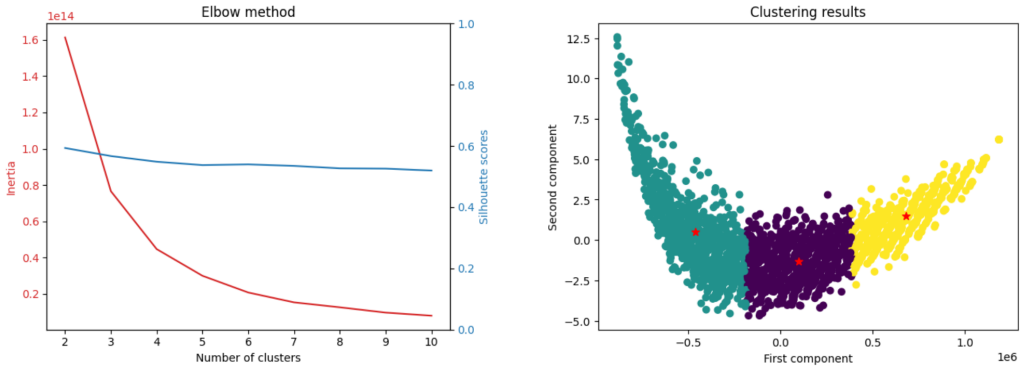

Figure 3.10: On the left the changes, based on the number of clusters, of inertia in red and silhouette scores in light blue in the elbow graph produced by the elbow method. On the right the result of the clustering performed with k-Means plotted on the two most significant components of the dataset with the centroids of each cluster identified by red stars.

- compactness,
- connectedness,
- flow hierarchy,
- global reaching centrality,
- average betweenness centrality,
- average closeness centrality,
- average node connectivity,
- edge connectivity,
- overall reciprocity,
- average in-degree.

The k-dist graph computed before DBSCAN suggests a value of ε equal to 30000, so the model partitions the data in two clusters and identifies 24 noise

points. Figure 3.11 shows the k-dist graph and the plot of the clustering results on the two main data dimensions. Moreover, from the process of selecting the characteristic metrics of the various clusters based on mean and variance, as explained in the 2.2.4 section, we get the metrics:

- density,
- s-metric,
- average clustering,
- number of strongly connected components,
- transitivity,
- compactness,
- connectedness,
- flow hierarchy,
- global reaching centrality,
- average closeness centrality,
- average node connectivity,
- edge connectivity,
- overall reciprocity,
- average in-degree.

The search for the best parameter combinations for the Gaussian Mixture model application suggests 4 as **n_components** and ”full” as **covariance_type**, so the model partitions the data in four clusters. Figure 3.12 shows the plot of the clustering results on the two main data dimensions. Moreover, from the process of selecting the characteristic metrics of the various clusters based on mean and variance, as explained in the 2.2.4 section, we get the metrics:

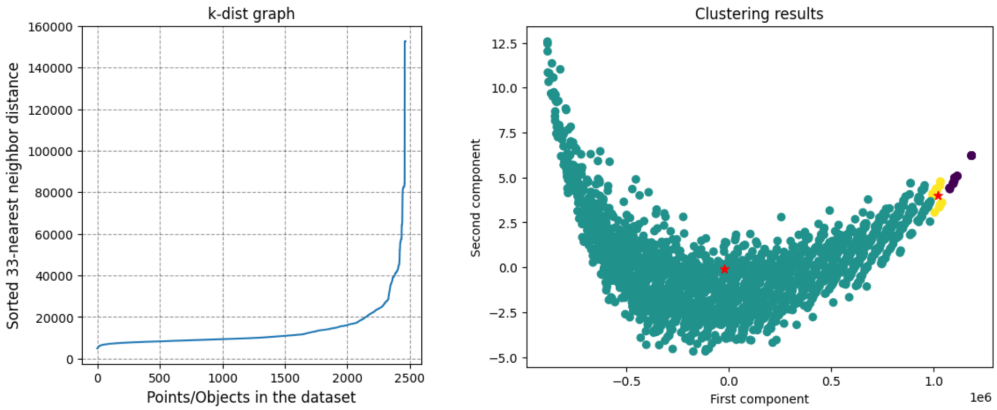

Figure 3.11: On the left the k-dist graph, that plots the changes of the k-nearest-neighbor distances computed for each point and ordered from largest to smallest. On the right the result of the clustering performed with DBSCAN plotted on the two most significant components of the dataset with the centroids of each cluster identified by red stars and the noise points identified by purple points.

- density,
- s-metric,
- average clustering,
- number of strongly connected components,
- transitivity,
- compactness,
- connectedness,
- flow hierarchy,
- global reaching centrality,
- average betweenness centrality,
- average closeness centrality,

Figure 3.12: The result of the clustering performed with Gaussian Mixture model plotted on the two most significant components of the dataset with the centroids of each cluster identified by red stars.

- average node connectivity,
- edge connectivity,
- overall reciprocity,
- average in-degree.

If we take a look at the metrics that characterize the clusters identified by each algorithm we notice that all three selected these metrics:

- density,
- s-metric,
- average clustering,
- number of strongly connected components,
- transitivity,
- compactness,

	k-Means	DBSCAN	Gaussian Mixture Model
Silhouette score	0.56707	0.40530	0.41015

Table 3.5: Silhouette scores of clustering applied on metrics on graphs.

- connectedness,
- flow hierarchy,
- global reaching centrality,
- average closeness centrality,
- average node connectivity,
- edge connectivity,
- overall reciprocity,
- average in-degree.

This means that 14 out of the 20 metrics computed on graphs are significant and ”average betweenness centrality” is the only metric that is considered significant for k-Means and Gaussian Mixture Model but not for DBSCAN.

The table 3.5 shows the silhouette scores of the three clustering algorithms applied. As we can see k-Means gets the highest score among the three models, and all three produce a satisfactory score.

3.2.5 Metrics on deep spanning trees

First, we report the mean and the variance of the metrics calculated on the deep spanning trees in table 3.6.

For k-Means the elbow method suggests using a value of **k** equal to 3, so the algorithm partitions the data in three clusters. Figure 3.13 shows the graph produced by the elbow method and the plot of the clustering results on the two main dimensions of the data. Moreover, from the process of selecting the characteristic metrics of the various clusters based on mean and variance, as explained in the 2.2.4 section, we get the metrics:

	Mean	Variance
Average out-degree	1.766	0.006
Compactness	9.486	13.585
Average closeness centrality	0.036	0.0
Average betweenness centrality	0.092	0.001
Average edge betweenness centrality	0.098	0.001
Density	0.046	0.0
S-metric	278.661	4048.156
Number of isolates	0.0	0.0
Number of weakly connected components	1.017	0.017
Size	89.860	10.327
Breadth	4.651	4.763
Load balance	1.345	0.204
Height	33.062	72.596

Table 3.6: Metrics computed on deep spanning trees.

- s-metric,
- average out-degree,
- average edge betweenness centrality,
- average betweenness centrality,
- load balance,
- height,
- breadth,
- size.

The k-dist graph computed before DBSCAN suggests a value of ε equal to 20, so the model partitions the data in one cluster and identifies 10 noise points. Figure 3.14 shows the k-dist graph and the plot of the clustering results on the two main data dimensions. Moreover, from the process of selecting the characteristic metrics of the various clusters based on mean and variance, as explained in the 2.2.4 section, we get the metrics:

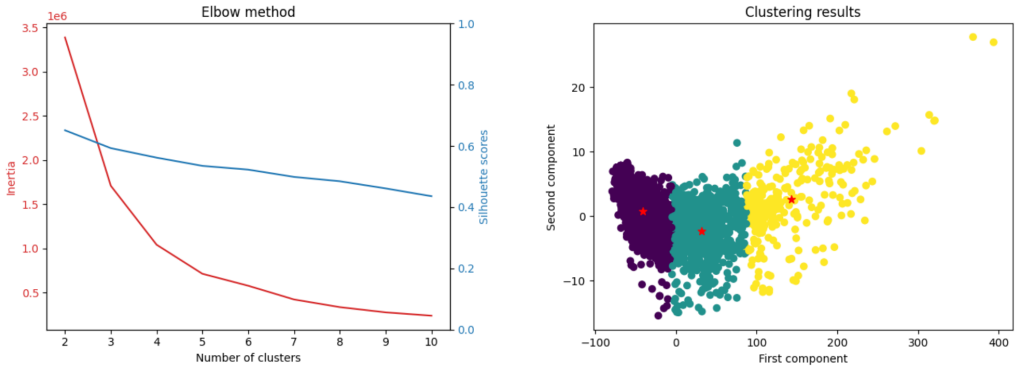

Figure 3.13: On the left the changes, based on the number of clusters, of inertia in red and silhouette scores in light blue in the elbow graph produced by the elbow method. On the right the result of the clustering performed with k-Means plotted on the two most significant components of the dataset with the centroids of each cluster identified by red stars.

- density,
- s-metric,
- compactness,
- number of weakly connected components,
- average edge betweenness centrality,
- average betweenness centrality,
- average closeness centrality,
- load balance,
- height,
- breadth,
- size.

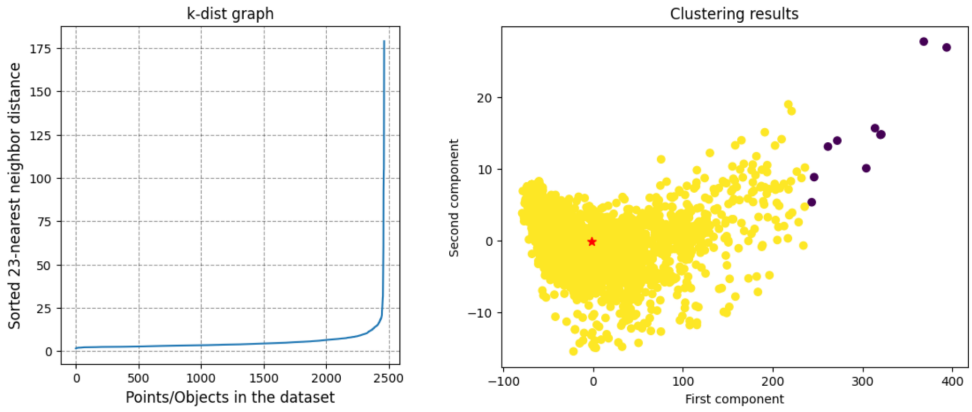

Figure 3.14: On the left the k-dist graph, that plots the changes of the k-nearest-neighbor distances computed for each point and ordered from largest to smallest. On the right the result of the clustering performed with DBSCAN plotted on the two most significant components of the dataset with the centroid of the cluster identified by a red star and the noise points identified by purple points.

The search for the best parameter combinations for the Gaussian Mixture model application suggests 3 as **n_components** and ”full” as **covariance_type**, so the model partitions the data in three clusters. Figure 3.15 shows the plot of the clustering results on the two main data dimensions. Moreover, from the process of selecting the characteristic metrics of the various clusters based on mean and variance, as explained in the 2.2.4 section, we get the metrics:

- density,
- s-metric,
- compactness,
- average out-degree,
- average edge betweenness centrality,
- average betweenness centrality,
- load balance,

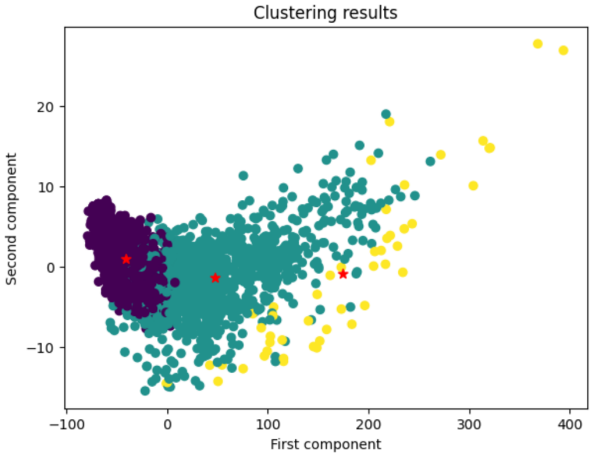

Figure 3.15: The result of the clustering performed with Gaussian Mixture model plotted on the two most significant components of the dataset with the centroids of each cluster identified by red stars.

- height,
- breadth,
- size.

If we take a look at the metrics that characterize the clusters identified by each algorithm we notice that all three selected these metrics:

- s-metric,
- average edge betweenness centrality,
- average betweenness centrality,
- load balance,
- height,
- breadth,
- size.

	k-Means	DBSCAN	Gaussian Mixture Model
Silhouette score	0.59290	0.74845	0.47334

Table 3.7: Silhouette scores of clustering applied on metrics on deep spanning trees.

	Mean	Variance
Average out-degree	0.966	0.013
Compactness	0.088	0.0
Average closeness centrality	0.017	0.0
Average betweenness centrality	0.001	0.0
Average edge betweenness centrality	0.001	0.0
Density	0.073	0.0
S-metric	1522.583	18392.244
Number of isolates	0.0	0.0
Number of weakly connected components	1.017	0.017
Size	76.681	34.939
Breadth	19.003	0.006
Load balance	1.793	0.049
Height	5.599	0.467

Table 3.8: Metrics computed on shallow spanning trees.

DBSCAN and Gaussian Mixture Model select pretty much the same metrics except for ”number of weakly connected components” and ”average closeness centrality” in DBSCAN, while k-Means select fewer metrics. In total 7 out of 13 metrics were selected.

The table 3.7 shows the silhouette scores of the three clustering algorithms applied. As we can see DBSCAN gets the highest score among the three models, and all three produce a satisfactory score.

3.2.6 Metrics on shallow spanning trees

First, we report the mean and the variance of the metrics calculated on the shallow spanning trees in table 3.8.

For k-Means the elbow method suggests using a value of **k** equal to 3, so the

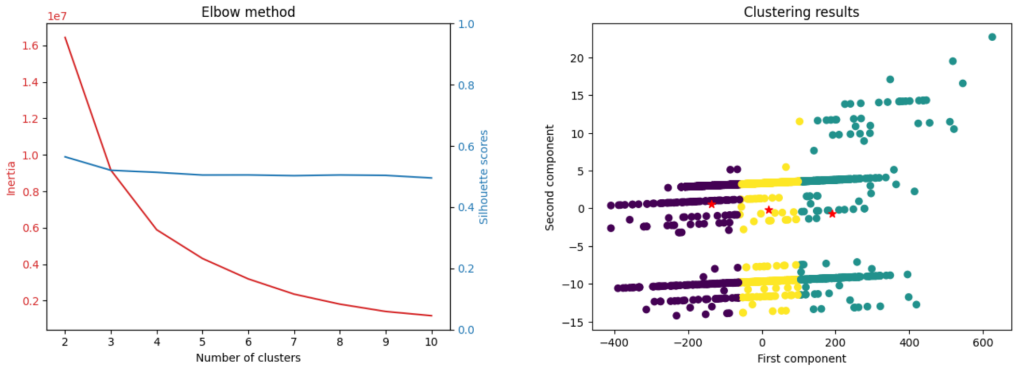

Figure 3.16: On the left the changes, based on the number of clusters, of inertia in red and silhouette scores in light blue in the elbow graph produced by the elbow method. On the right the result of the clustering performed with k-Means plotted on the two most significant components of the dataset with the centroids of each cluster identified by red stars.

algorithm partitions the data in three clusters. Figure 3.16 shows the graph produced by the elbow method and the plot of the clustering results on the two main dimensions of the data. Moreover, from the process of selecting the characteristic metrics of the various clusters based on mean and variance, as explained in the 2.2.4 section, we get the metrics:

- density,
- s-metric,
- compactness,
- average out-degree,
- average closeness centrality,
- load balance,
- size.

The k-dist graph computed before DBSCAN suggests a value of ε equal to 20, so the model partitions the data in one cluster and identifies 78 noise points. Figure 3.17 shows the k-dist graph and the plot of the clustering results on the two main data dimensions. Moreover, from the process of selecting the characteristic metrics of the various clusters based on mean and variance, as explained in the 2.2.4 section, we get the metrics:

- density,
- s-metric,
- number of weakly connected components,
- compactness,
- average out-degree,
- average closeness centrality,
- average betweenness centrality,
- size.

The search for the best parameter combinations for the Gaussian Mixture model application suggests 6 as **n_components** and "diagonal" as **covariance_type**, so the model partitions the data in six clusters. Figure 3.18 shows the plot of the clustering results on the two main data dimensions. Moreover, from the process of selecting the characteristic metrics of the various clusters based on mean and variance, as explained in the 2.2.4 section, we get the metrics:

- density,
- s-metric,
- compactness,
- size.

If we take a look at the metrics that characterize the clusters identified by each algorithm we notice that all three selected these metrics:

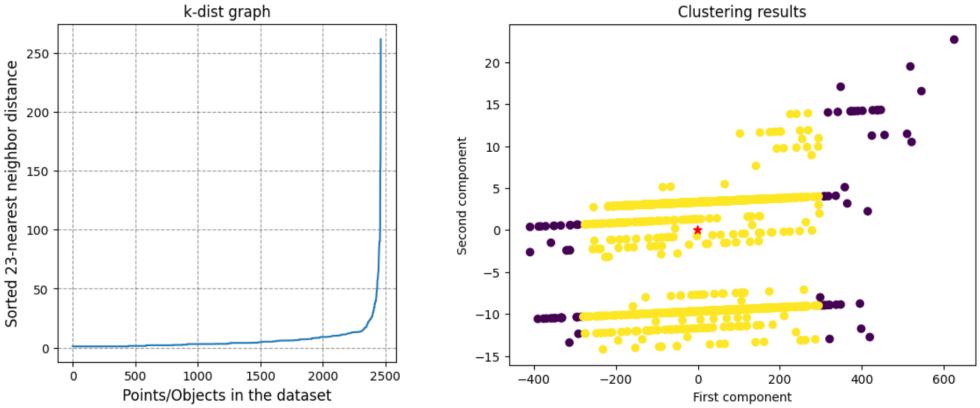

Figure 3.17: On the left the k-dist graph, that plots the changes of the k-nearest-neighbor distances computed for each point and ordered from largest to smallest. On the right the result of the clustering performed with DBSCAN plotted on the two most significant components of the dataset with the centroid of the cluster identified by a red star and the noise points identified by purple points.

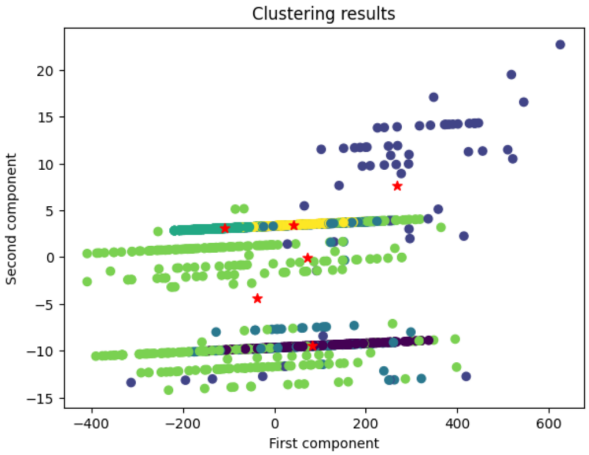

Figure 3.18: The result of the clustering performed with Gaussian Mixture model plotted on the two most significant components of the dataset with the centroids of each cluster identified by red stars.

	k-Means	DBSCAN	Gaussian Mixture Model
Silhouette score	0.52055	0.59599	0.06579

Table 3.9: Silhouette scores of clustering applied on metrics on shallow spanning trees.

- density,
- s-metric,
- compactness,
- size.

Here only four metrics were selected mainly because of Gaussian Mixture Model, while k-Means and DBSCAN have only a few differences between the two of them.

The table 3.9 shows the silhouette scores of the three clustering algorithms applied. As we can see DBSCAN gets the highest score among the three models, with k-Means producing a similar score, but the Gaussian Mixture Model doesn't produce a satisfactory score.

Chapter 4

Discussion

4.1 Results discussion

We begin to discuss the results of the analysis phase from the graph representation. From the silhouette score of k-Means we note how none of the three clustering algorithms produce satisfactory results in partitioning the students, with DBSCAN providing the best but still poor performance among the three. This was already imaginable from the elbow graph where the inertia did not form a true elbow and the silhouette score always remained very low. Even the k-dist graph calculated before DBSCAN already showed that there was no drastic improvement as the radius ε of the algorithm changed. In fact, DBSCAN can only detect one cluster and classifies the remaining one-fifth of the students as noise points. Gaussian Mixture Model also does not give great results despite the fact that these three methods all use different techniques to partition students. In fact, in contrast to what one might expect when looking at the plot of the data on the two principal components, which are the data features that capture the most variance in the data, in which there appear to be three distinct clusters, none of these three algorithms succeeds in identifying them, suggesting that the differences in the data are probably not as marked as one would expect.

The same considerations about the results of the clustering can also be made for the deep spanning trees, where the silhouette scores show even worse performance than the graphs, with Gaussian Mixture Model even having a negative

score. However, the plot on the two principal components does not show a clear separation between the data, rather, there seems to be only small differences between the points and nothing that makes the data separable.

The case of the shallow spanning trees, on the other hand, is more interesting. The silhouette scores of the three algorithms are promising with values of all three above 0.5. In addition, the three algorithms suggest a conspicuous number of clusters compared to those identified by clustering on the other representations, in fact k-Means identifies 4 clusters, DBSCAN 7 and Gaussian Mixture Model 6. Particularly interesting is the elbow graph visualized before applying k-Means, as we notice how the inertia and silhouette scores follow inverse trends, i.e. as the inertia decreases dramatically the silhouette score increases, although by less than two decimal points.

The metrics calculated on the three representations, however, can tell us interesting things about the tests. Starting with the graphs we notice that there are four metrics, average betweenness centrality, average edge betweenness centrality, number of isolates and number of weakly connected components, that have variance equal to 0, which means that the results of these metrics for each student are identical. The number of isolates has also a mean equal to 0 indicating that all graphs are connected and there are no external nodes. The number of weakly connected components has mean 1 which means no graph is disconnected. The two metrics on betweenness centrality have mean close to 0 indicating that there are no particularly critical edges or nodes in the graphs. Also, average in-degree and average out-degree have the same mean and variance indicating that the number of dimensions sharing nodes is fairly balanced. All other metrics except s-metric, number of strongly connected components, average in-degree, average out-degree and average node connectivity have very low variance indicating that according to these metrics the graphs are all similar. Looking, however, at the various averages we can see that the graphs tend to be very compact, connected and fairly dense. In addition, there are several cycles within them and transitivity is discretely high, as well as average clustering and overall reciprocity. As for connectivity, it is high for nodes and very low for edges.

The metrics on deep spanning trees have some things in common with the previous ones just described. For example, the number of isolates here also has

variance equal to 0, as does average closeness centrality and density. Average betweenness centrality, average edge betweenness centrality, and number of weakly connected components, along with average out-degree and load balance, have a variance that is close to 0, while all other metrics have more or less high variance. Moving to the averages, the average out-degree is in line with expectations, as are the various centrality measures, which are all close to 0 because the graphs are very deep. For the same reason we have a low breadth and a very high height. These trees appear compact but not dense since the trees are constructed with the minimum number of edges possible. The number of weakly connected components suggests to us that most students have only one tree and that node pairs tend more likely to share one or two dimensions.

Finally, the metrics on spanning trees have the peculiarity that almost all of them have variance equal to 0 or tending to 0 except s-metric and size, which indicates that the graphs are very similar according to these metrics. If we study the averages we again see that the average of the number of isolates is 0, that the centralities have an average close to 0 and that the trees are not very compact nor dense. The number of weakly connected components has the same mean and variance as the other tree type, but here obviously breadth is higher than height. The size is slightly lower than its counterpart, while the load balance is slightly higher. Here the average out-degree is lower than in the deep spanning trees as it is easier for fictitious nodes to be added.

The clustering results on the metrics, on the other hand, can be much more helpful to us. Starting again with the metrics on the graphs, the silhouette scores of the three clustering algorithms are quite good even though the three algorithms identify a different number of clusters, which could mean that the three algorithms capture different features of the data or with a different level of depth. However, all three algorithms select three-fourths of the metrics, with a minimal difference of one metric that DBSCAN does not select. Of particular interest is the elbow graph, which shows that as the inertia changes, the k-Means silhouette score hardly changes, even though the inertia forms a discrete elbow. The k-dist graph also forms a significant elbow, however DBSCAN partitions almost all of the data into a single cluster and the remaining points between a very small cluster and noise points.

The same considerations about the silhouette scores and the elbow graph and the k-dist graph also apply to the deep spanning trees. In this case, about half of the calculated metrics are selected with some differences between the various algorithms, among other things, all of which are different from those selected by the graphs except one, the s-metric.

Again the same considerations about the elbow graph and the k-dist graph apply to the shallow spanning trees, but this time the silhouette score is good only for k-Means and DBSCAN, while for Gaussian Mixture Model it is very low. The metrics in common selected by the three algorithms are only four because of the poor results of Gaussian Mixture Model, while the other two models identify similar but not identical sets of metrics of which some are also significant for graphs.

4.2 Method validity and limitations

Clustering applied directly on the first three representations unfortunately provides no particular insights and in the case of graphs and deep spanning trees is also quite disappointing. An attempt was made to check whether there was any correlation between the clusters identified by the three algorithms with the possible future dropout of students, a datum present in the initial dataset from which the students' test responses were isolated, but in fact even this was not found to be significant. In fact, although the relative frequencies indicated that some clusters partitioned the data more according to dropout, the absolute frequencies showed that the separation was not sharp enough to be significant. In spite of all this, shallow spanning trees gave promising clustering results, however, it is not easy to understand and extract what factors determine data separation, and a great deal of work would be needed to understand whether the clusters can be linked with background information of the students or with the size of the various questions according to which ones were answered correctly.

In addition, the fact that the connections between pairs of nodes indicate only how many dimensions they share and do not specify which ones can be limiting for the next step of evaluating the clustering results, especially the one applied directly on the three representations.

Clustering applied to metrics, on the other hand, gives good results, and in this case we can also identify which metrics characterize the clusters the most, but only an expert in this domain can actually draw meaningful considerations from these results and then translate them into our context, so only at that point it will be possible to evaluate whether indeed these results are useful for assessing and understanding the students' learning state.

Certainly the choice to focus only on the students' responses to the test and the metrics calculated on the representations ensures that the data do not contain bias arising from the students' background, however, the classification of the questions based on the three dimensions was done by hand by an expert. Also, if we look at the distribution of the data based on dropout this is not balanced in the dataset.

To recap, with the current method only clustering on the spanning trees and on the metrics computed on the various representations give satisfactory results that are worth analyzing. However, direct clustering on the spanning trees is very complex to interpret since it is very difficult to extract any factors that characterize the data since the clustering is performed directly on the adjacency matrix, whereas in clustering on the metrics we can isolate characterizing metrics for individual clusters that perhaps globally did not give particularly relevant information.

4.3 Future works

In light of the limitations just outlined there are several future directions that this work can take to further explore the issues of analyzing a student's learning path and learning state.

First and foremost, other clustering algorithms could be attempted especially with regard to graphs and deep spanning trees where those tested gave poor results.

In addition, a multi-layer graph structure could be proposed so that in addition to knowing how many dimensions each pair of nodes shares, we can also know which ones are shared by isolating the layers of the graph.

Another interesting insight that would help shed light on the results already obtained instead, would be a deep study of the direct clustering results on the graph and spanning tree representations to look for possible correlations with other student data, such as family or educational background, or with the three dimensions

of the questions, area, process and macro-process.

The results of this work are also a good starting point for experts in this domain, particularly those who design tests to assess students' learning paths, to try to find more details about the difficulties that students may find during learning and while taking the test, as well as on the test itself.

Conclusions

Assessment of student learning states and paths is a crucial task for educators and researchers in education. In the course of this work, we examined different representations of the data obtained from fifth-grade students' answers to the INVALSI math test using clustering algorithms. The main objective was to understand if and how it was possible to divide students into homogeneous clusters based on their performance on the test in order to identify any distinctive patterns or characteristics in the students' learning state.

In our study, we examined different representations of the data, graphs, deep spanning trees and shallow spanning trees, as well as computed metrics on these representations for an initial stage of analysis. Next, we applied three clustering algorithms, k-Means, DBSCAN and Gaussian Mixture Model, on each of these representations and the metrics calculated on them to try to identify any significant clusters in the students' learning state.

The results obtained highlight a variety of challenges and opportunities in applying clustering to this type of data. In particular, direct clustering analysis on graphs and deep spanning trees showed disappointing performance. This suggests that the structure of the data may not be easily interpretable or separable using such approaches. However, the analysis conducted on shallow spanning trees produced more promising results, with more clusters identified and higher silhouette scores. This may indicate that reducing the complexity of data representation may improve the ability of clustering methods to identify meaningful patterns in the data.

In addition, analysis of the metrics computed on each representation provided additional insights, revealing patterns and characteristics of the data that may not have been highlighted by clustering methods directly applied to the representa-

tions. For example, we observed that the metrics computed on the graphs tended to show strong connectivity and density, while the metrics computed on the deep and shallow spanning trees showed different characteristics such as height, width and load balance. However, even here, understanding the results requires in-depth knowledge of the specific domain and educational contexts.

In any case, it is important to note that the results obtained in this study have some limitations. For example, we found that the correlation between the clusters identified and student dropout was not significant, suggesting that the clusters identified may not be closely correlated with student performance, but the unbalanced distribution of the data based on student dropout might have influenced the results of the analysis. Furthermore, interpreting the results obtained directly from clustering on the data representations requires an additional and in-depth analysis work to be able to find patterns.

To address these limitations and deepen our understanding of the student learning state, further study and future research is needed. For example, we could explore the use of other clustering algorithms or try other representations. In addition, we could examine how the identified clusters correlate with other variables, such as students' backgrounds or the size of test questions.

In conclusion, this study represents a first step toward a better understanding of students' learning states through the analysis of test data. Despite the challenges and limitations encountered, the results obtained provide interesting insights for further research and can be used as a basis for the development of targeted and personalized educational interventions. However, it is important to emphasize that understanding the student learning state is a complex and ever-evolving field, and what we present in this study represents only a small part of what might be possible. With further research and collaborative efforts, we hope to develop a deeper understanding of this important topic and make a significant contribution to improving student instruction and assessment.

Bibliography

[1] Kay Sambell, Sally Brown, and Phil Race. Assessment to support student learning: eight challenges for 21st century practice. *All Ireland Journal of Teaching and Learning in Higher Education (AISHE-J) Creative Commons Attribution-NonCommercial-ShareAlike*, 11(2), 2019.

[2] Sue Margaret Norton. Challenges to effective assessment of learning. *Academic Exchange Quarterly*, 10(3):234–236, 2006.

[3] Cecilia L Lopez. Assessment of student learning: challenges and strategies. *The Journal of Academic Librarianship*, 28(6):356–367, November 2002. ISSN 0099-1333. doi: 10.1016/s0099-1333(02)00345-2. URL http://dx.doi.org/10.1016/S0099-1333(02)00345-2.

[4] Chan Yuen Fook and Gurnam Kaur Sidhu. Investigating learning challenges faced by students in higher education. *Procedia - Social and Behavioral Sciences*, 186:604–612, May 2015. ISSN 1877-0428. doi: 10.1016/j.sbspro.2015.04.001. URL http://dx.doi.org/10.1016/J.SBSPRO.2015.04.001.

[5] William D. Hendricson and John H. Kleffner. Assessing and helping challenging students: Part one, why do some students have difficulty learning? *Journal of Dental Education*, 66(1):43–61, January 2002. ISSN 1930-7837. doi: 10.1002/j.0022-0337.2002.66.1.tb03507.x. URL http://dx.doi.org/10.1002/J.0022-0337.2002.66.1.TB03507.X.

[6] Stephen L. Chew and William J. Cerbin. The cognitive challenges of effec-

tive teaching. *The Journal of Economic Education*, 52(1):17–40, November 2020. ISSN 2152-4068. doi: 10.1080/00220485.2020.1845266.

[7] Jonathan Michael Spector. Assessing progress of learning in complex domains. In *The 11th International Conference on Education Research*, 2010.

[8] Filip J. R. C. Dochy. *Investigating the Use of Knowledge Profiles in a Flexible Learning Environment: Analyzing Students' Prior Knowledge States*, page 235–242. Springer Berlin Heidelberg, 1994. ISBN 9783642791499. doi: 10.1007/978-3-642-79149-9_30.

[9] Sergi Rovira, Eloi Puertas, and Laura Igual. Data-driven system to predict academic grades and dropout. *PLOS ONE*, 12(2):e0171207, February 2017. ISSN 1932-6203. doi: 10.1371/journal.pone.0171207.

[10] Ashay Tamhane, Shajith Ikbal, Bikram Sengupta, Mayuri Duggirala, and James Appleton. Predicting student risks through longitudinal analysis. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '14. ACM, August 2014. doi: 10.1145/2623330.2623355.

[11] Adena M. Klem and James P. Connell. Relationships matter: Linking teacher support to student engagement and achievement. *Journal of School Health*, 74(7):262–273, September 2004. ISSN 1746-1561. doi: 10.1111/j.1746-1561.2004.tb08283.x.

[12] Nancy Feyl Chavkin and David L Williams. Critical issues in teacher training for parent involvement. *Educational Horizons*, 66(2):87–89, 1988.

[13] Aurora-Tatiana Dina. Challenges faced by educational leadership on influencing student learning. *Procedia - Social and Behavioral Sciences*, 93: 290–295, October 2013. ISSN 1877-0428. doi: 10.1016/j.sbspro.2013.09.192.

[14] Donald Shipman, Susan L. Aloi, and Elizabeth A. Jones. Addressing key challenges of higher education assessment. *The Journal of General Educa-*

tion, 52(4):335–346, 2003. ISSN 1527-2060. doi: 10.1353/jge.2004.0016. URL http://dx.doi.org/10.1353/JGE.2004.0016.

[15] Lorelei Carpenter and P. Matters. Learning communities today - who benefits? 01 2003.

[16] INVALSI. Istituto nazionale per la valutazione del sistema educativo di istruzione e di formazione. URL https://www.invalsi.it/.

[17] Maarten Van Steen. Graph theory and complex networks. *An introduction*, 144:1–287, 2010.

[18] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, June 1972. ISSN 1095-7111. doi: 10.1137/0201010.

[19] Dongkuan Xu and Yingjie Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193, June 2015. ISSN 2198-5812. doi: 10.1007/s40745-015-0040-1. URL http://dx.doi.org/10.1007/s40745-015-0040-1.

[20] J. MacQueen. Some methods for classification and analysis of multivariate observations. 1967.

[21] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Knowledge Discovery and Data Mining*, 1996. URL https://api.semanticscholar.org/CorpusID:355163.

[22] Richard A. Redner and Homer F. Walker. Mixture densities, maximum likelihood and the em algorithm. *SIAM Review*, 26(2):195–239, 1984. ISSN 00361445. URL http://www.jstor.org/stable/2030064.

[23] C. Romero and S. Ventura. Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, 33(1):135–146, 2007. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2006.04.005.

[24] Tom M. Mitchell. Machine learning, 1997.

[25] Mrinal Pandey and Vivek Sharma. A decision tree algorithm pertaining to the student performance analysis and prediction. *International Journal of Computer Applications*, 61:1–5, 01 2013. doi: 10.5120/9985-4822.

[26] Surjeet Kumar Yadav and Saurabh Pal. Data mining: A prediction for performance improvement of engineering students using classification. *World of Computer Science and Information Technology Journal*, 2(2):51–56, 2012. doi: 10.48550/ARXIV.1203.3832.

[27] Surjeet Kumar Yadav, Brijesh Bharadwaj, and Saurabh Pal. Data mining applications: A comparative study for predicting student's performance. *INTERNATIONAL JOURNAL OF INNOVATIVE TECHNOLOGY CREATIVE ENGINEERING*, 1(12), December 2012. doi: 10.48550/ARXIV.1202.4815.

[28] Gerben Dekker, Mykola Pechenizkiy, and Jan Vleeshouwers. Predicting students drop out: A case study. *Computers, Environment and Urban Systems*, pages 41–50, 01 2009.

[29] Amjad Abu Saa. Educational data mining students' performance prediction. *International Journal of Advanced Computer Science and Applications*, 7, 05 2016. doi: 10.14569/IJACSA.2016.070531.

[30] Ruhi R. Kabra and R. S. Bichkar. Performance prediction of engineering students using decision trees. *International Journal of Computer Applications*, 36(11), December 2011.

[31] Brijesh Baradwaj and Saurabh Pal. Mining educational data to analyze students' performance. *International Journal of Advanced Computer Science and Applications*, 2:63–69, 10 2011. doi: 10.14569/IJACSA.2011.020609.

[32] Ramanathan Lakshmanan, Saksham Dhanda, and D. Kumar. Predicting students' performance using modified id3 algorithm. *International Journal of Engineering and Technology*, 5:2491–2497, 06 2013.

[33] Javier Bravo and Alvaro Ortigosa. Detecting symptoms of low performance using production rules. *EDM'09 - Educational Data Mining 2009: 2nd*

International Conference on Educational Data Mining, pages 31–40, 01 2009.

[34] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. *ACM SIGMOD Record*, 22 (2):207–216, June 1993. ISSN 0163-5808. doi: 10.1145/170036.170072. URL http://dx.doi.org/10.1145/170036.170072.

[35] G V S Ch S L V Prasad, M. Rambabu, and K. Naresh Kumar. Association rule generation for student performance analysis using apriori algorithm. *Journal of Science and Technology*, 7(03):107–112, May 2022.

[36] Xiaodong Wu and Yuzhu Zeng. Using apriori algorithm on students' performance data for association rules mining. In *Proceedings of the 2nd International Seminar on Education Research and Social Science*, 01 2019. doi: 10.2991/iserss-19.2019.300.

[37] Tengfei Wang, Baorong Xiao, and Weixiao Ma. Student behavior data analysis based on association rule mining. *International Journal of Computational Intelligence Systems*, 15(1), May 2022. ISSN 1875-6883. doi: 10.1007/s44196-022-00087-4. URL http://dx.doi.org/10.1007/s44196-022-00087-4.

[38] Mushtaq Hussain, Wenhao Zhu, Wu Zhang, and Syed Muhammad Raza Abidi. Student engagement predictions in an e-learning system and their impact on student course assessment scores. *Computational Intelligence and Neuroscience*, 2018:1–21, October 2018. ISSN 1687-5273. doi: 10.1155/2018/6347186. URL http://dx.doi.org/10.1155/2018/6347186.

[39] Azwa Abdul Aziz, Ismail Nor Hafieza, and Ahmad Norashikin. First semester computer science students ' academic performances analysis by using data mining classification algorithms. *Journal of Artificial Intelligence and Computer Science*, 14, 2014.

[40] Dorina Kabakchieva. Predicting student performance by using data mining methods for classification. *Cybernetics and Information Technologies*, 13

(1):61–72, March 2013. ISSN 1311-9702. doi: 10.2478/cait-2013-0006. URL http://dx.doi.org/10.2478/cait-2013-0006.

[41] Genetu Yohannes Vuda Sreenivasarao. Improving academic performance of students of defence university based on data warehousing and data mining. *Global Journal of Computer Science and Technology*, 12(2), jan 2012.

[42] Sreedevi Kadiyala and Chandrasekhar Potluri. Analyzing the student's academic performance by using clustering methods in data mining. *International Journal of Scientific Engineering Research*, 5(6), June 2014.

[43] Alaa El-Halees. Mining students data to analyze learning behavior: A case study. 01 2009.

[44] Simon Haykin. *Neural networks and learning machines*. Pearson Education, third edition, 2009.

[45] Sajadin Sembiring, M. Zarlis, Dedy Hartama, S. Ramliana, and E. Wani. Prediction of student academic performance by an application of data mining techniques. *International Conference on Management and Artificial Intelligence*, 6:110–114, 01 2011.

[46] Ahmet Tekin. Early prediction of students' grade point averages at graduation: A data mining approach. *Eurasian Journal of Educational Research*, 14:207–226, 02 2014. doi: 10.14689/ejer.2014.54.12.

[47] Xiao Hu, Christy W. L. Cheong, Wenwen Ding, and Michelle Woo. A systematic review of studies on predicting student learning outcomes using learning analytics. In *Proceedings of the Seventh International Learning Analytics amp; Knowledge Conference*, LAK '17. ACM, March 2017. doi: 10.1145/3027385.3029438. URL http://dx.doi.org/10.1145/3027385.3029438.

[48] Abdallah Namoun and Abdullah Alshanqiti. Predicting student performance using data mining and learning analytics techniques: A systematic literature review. *Applied Sciences*, 11(1):237, December 2020. ISSN

2076-3417. doi: 10.3390/app11010237. URL `http://dx.doi.org/10.3390/app11010237`.

[49] Alejandro Peña-Ayala. Educational data mining: A survey and a data mining-based analysis of recent works. *Expert Systems with Applications*, 41(4):1432–1462, March 2014. ISSN 0957-4174. doi: 10.1016/j.eswa.2013.08.042.

[50] Karthikeyan Govindasamy and Velmurugan Thambusamy. Analysis of student academic performance using clustering techniques. *International Journal of Pure and Applied Mathematics*, 119:309–322, 01 2018.

[51] Bindiya Varghese, Avittathur Unnikrishnan, and K. Jacob. Clustering student data to characterize performance patterns. *IJACSA*, Special Issue, 09 2011. doi: 10.14569/SpecialIssue.2011.010322.

[52] V. K. Anand, S. K. Abdul Rahiman, E. Ben George, and A. S. Huda. Recursive clustering technique for students' performance evaluation in programming courses. In *2018 Majan International Conference (MIC)*, pages 1–5, 2018. doi: 10.1109/MINTC.2018.8363153.

[53] Tallal Omar, Abdullah Alzahrani, and Mohamed Zohdy. Clustering approach for analyzing the student's efficiency and performance based on data. *Journal of Data Analysis and Information Processing*, 08(03): 171–182, 2020. ISSN 2327-7203. doi: 10.4236/jdaip.2020.83010.

[54] O. J. Oyelade, O. O. Oladipupo, and I. C. Obagbuwa. Application of k means clustering algorithm for prediction of students academic performance. *International Journal of Computer Science and Information Security*, 7(1), 2010. doi: 10.48550/ARXIV.1002.2425.

[55] Rakesh Arora and Dr Badal. Evaluating student's performance using k-means clustering. *www.ijcst.com*, 4, 06 2013.

[56] J. Jamesmanoharan, S. Hari Ganesh, M. Lovelin Ponn Felciah, and A. K. Shafreenbanu. Discovering students' academic performance based on gpa using k-means clustering algorithm. In *2014 World Congress on Computing*

and Communication Technologies. IEEE, February 2014. doi: 10.1109/wccct.2014.75. URL http://dx.doi.org/10.1109/WCCCT.2014.75.

[57] Shashikant Pradip Borgavakar and Amit Shrivastava. Evaluating student's performance using k-means clustering. *International Journal of Engineering Research & Technology (IJERT)*, 6(5), may 2017.

[58] Minimol Anil Job. Data mining techniques applying on educational dataset to evaluate learner performance using cluster analysis. *European Journal of Engineering Research and Science*, 3(11):25–31, November 2018. ISSN 2506-8016. doi: 10.24018/ejers.2018.3.11.966. URL http://dx.doi.org/10.24018/ejers.2018.3.11.966.

[59] Alkadhwi Ali Hussein and Adelaja Oluwaseun. Data mining application using clustering techniques (k-means algorithm) in the analysis of student's result. *Journal of Multidisciplinary Engineering Science Studies*, 5(5):2458–925, 05 2019.

[60] Dianwei Chi. Research on the application of k-means clustering algorithm in student achievement. In *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pages 435–438, 2021. doi: 10.1109/ICCECE51280.2021.9342164.

[61] Haiyun Bian. A preliminary study on clustering student learning data. In Sofia Visa, Atsushi Inoue, and Anca L. Ralescu, editors, *Proceedings of The 22nd Midwest Artificial Intelligence and Cognitive Science Conference 2011, Cincinnati, Ohio, USA, April 16-17, 2011*, volume 710 of *CEUR Workshop Proceedings*, pages 128–132. CEUR-WS.org, 2011.

[62] Pavlo D. Antonenko, Serkan Toy, and Dale S. Niederhauser. Using cluster analysis for data mining in educational technology research. *Educational Technology Research and Development*, 60(3):383–398, February 2012. ISSN 1556-6501. doi: 10.1007/s11423-012-9235-8. URL http://dx.doi.org/10.1007/s11423-012-9235-8.

[63] Agung Triayudi and Isfahani Fitri. Alg clustering to analyze the behavioural patterns of online learning students. *Journal of Theoretical and Applied Information Technology*, 96:5327–5337, 08 2018.

[64] Angela Bovo, Stéphane Sanchez, Olivier Héguy, and Yves Duthen. Clustering moodle data as a tool for profiling students. In *2013 Second International Conference on E-Learning and E-Technologies in Education (ICEEE)*, pages 121–126, 2013. doi: 10.1109/ICeLeTE.2013.6644359.

[65] K. D. DeFreitas and Margaret Bernard. Comparative performance analysis of clustering techniques in educational data mining. *IADIS International Journal on Computer Science Information Systems*, 10(2), 2015.

[66] Gamila Obadi, Pavla Drázdilová, Jan Martinovic, Kateřina Slaninová, and Vaclav Snasel. Using spectral clustering for finding students' patterns of behavior in social networks. *CEUR Workshop Proceedings*, 567:118–130, 01 2010.

[67] Agathe Merceron and Kalina Yacef. *Clustering Students to Help Evaluate Learning*, page 31–42. Springer US, 2005. ISBN 9780387240473. doi: 10.1007/0-387-24047-0_3. URL `http://dx.doi.org/10.1007/0-387-24047-0_3`.

[68] Zachary A. Pardos, Shubhendu Trivedi, Neil T. Heffernan, and Gábor N. Sárközy. *Clustered Knowledge Tracing*, page 405–410. Springer Berlin Heidelberg, 2012. ISBN 9783642309502. doi: 10.1007/978-3-642-30950-2_52. URL `http://dx.doi.org/10.1007/978-3-642-30950-2_52`.

[69] Argelia B. Urbina Nájera, Jorge de la Calleja, and Ma. Auxilio Medina. Associating students and teachers for tutoring in higher education using clustering and data mining. *Computer Applications in Engineering Education*, 25(5):823–832, June 2017. ISSN 1099-0542. doi: 10.1002/cae.21839. URL `http://dx.doi.org/10.1002/cae.21839`.

[70] Danuta Zakrzewska. *Cluster Analysis for Users' Modeling in Intelligent E-Learning Systems*, page 209–214. Springer Berlin Heidelberg, 2008.

ISBN 9783540690528. doi: 10.1007/978-3-540-69052-8_22. URL http://dx.doi.org/10.1007/978-3-540-69052-8_22.

[71] Prashant Sahai Saxena and MC Govil. Prediction of student's academic performance using clustering. In *National conference on cloud computing & big data*, pages 1–6, 2009.

[72] Ahmad Fikri Mohamed Nafuri, Nor Samsiah Sani, Nur Fatin Aqilah Zainudin, Abdul Hadi Abd Rahman, and Mohd Aliff. Clustering analysis for classifying student academic performance in higher education. *Applied Sciences*, 12(19):9467, September 2022. ISSN 2076-3417. doi: 10.3390/app12199467. URL http://dx.doi.org/10.3390/app12199467.

[73] Tanja Käser, Alberto Giovanni Busetto, Barbara Solenthaler, Juliane Kohn, Michael von Aster, and Markus Gross. *Cluster-Based Prediction of Mathematical Learning Patterns*, page 389–399. Springer Berlin Heidelberg, 2013. ISBN 9783642391125. doi: 10.1007/978-3-642-39112-5_40. URL http://dx.doi.org/10.1007/978-3-642-39112-5_40.

[74] Andrew Laghos and Panayiotis Zaphiris. Sociology of student-centred e-learning communities: A network analysis. 01 2006.

[75] Julio Guerra, Yun Huang, Roya Hosseini, and Peter Brusilovsky. Graph analysis of student model networks. *CEUR Workshop Proceedings*, 1446, 06 2015.

[76] Mengfan Liu, Pengyang Shao, and Kun Zhang. Graph-based exercise- and knowledge-aware learning network for student performance prediction, 2021.

[77] Hiromi Nakagawa, Yusuke Iwasawa, and Yutaka Matsuo. Graph-based knowledge tracing: Modeling student proficiency using graph neural network. In *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 156–163, 2019.

[78] Pilar Ortiz-Vilchis and Aldo Ramirez-Arellano. Learning pathways and students performance: A dynamic complex system. *Entropy*, 25(2):291,

February 2023. ISSN 1099-4300. doi: 10.3390/e25020291. URL http://dx.doi.org/10.3390/e25020291.

[79] Eva Millán, Tomasz Loboda, and Jose Luis Pérez-de-la Cruz. Bayesian networks for student model engineering. *Computers amp; Education*, 55(4): 1663–1683, December 2010. ISSN 0360-1315. doi: 10.1016/j.compedu.2010.07.010. URL http://dx.doi.org/10.1016/j.compedu.2010.07.010.

[80] Tanja Käser, Severin Klingler, Alexander Gerhard Schwing, and Markus Gross. *Beyond Knowledge Tracing: Modeling Skill Topologies with Bayesian Networks*, page 188–198. Springer International Publishing, 2014. ISBN 9783319072210. doi: 10.1007/978-3-319-07221-0_23. URL http://dx.doi.org/10.1007/978-3-319-07221-0_23.

[81] Michel C. Desmarais and Michel Gagnon. *Bayesian Student Models Based on Item to Item Knowledge Structures*, page 111–124. Springer Berlin Heidelberg, 2006. ISBN 9783540462347. doi: 10.1007/11876663_11. URL http://dx.doi.org/10.1007/11876663_11.

[82] Patricio García, Analía Amandi, Silvia Schiaffino, and Marcelo Campo. Evaluating bayesian networks' precision for detecting students' learning styles. *Computers amp; Education*, 49(3):794–808, November 2007. ISSN 0360-1315. doi: 10.1016/j.compedu.2005.11.017. URL http://dx.doi.org/10.1016/j.compedu.2005.11.017.

[83] Aldo Ramirez-Arellano. Students learning pathways in higher blended education: An analysis of complex networks perspective. *Computers amp; Education*, 141:103634, November 2019. ISSN 0360-1315. doi: 10.1016/j.compedu.2019.103634. URL http://dx.doi.org/10.1016/J.COMPEDU.2019.103634.

[84] Daqian Shi, Ting Wang, Hao Xing, and Hao Xu. A learning path recommendation model based on a multidimensional knowledge graph framework for e-learning. *Knowledge-Based Systems*, 195:105618, May 2020.

ISSN 0950-7051. doi: 10.1016/j.knosys.2020.105618. URL http://dx.doi.org/10.1016/j.knosys.2020.105618.

[85] S. Jiang, S.M. Fitzhugh, and M. Warschauer. Social positioning and performance in moocs. *CEUR Workshop Proceedings*, 1183:55–58, 01 2014.

[86] Irad Ben-Gal, Shahar Weinstock, Gonen Singer, and Nicholas Bambos. Clustering users by their mobility behavioral patterns. *ACM Transactions on Knowledge Discovery from Data*, 13(4):1–28, August 2019. ISSN 1556-472X. doi: 10.1145/3322126. URL http://dx.doi.org/10.1145/3322126.

[87] Jose Daniel P. Ribeiro Filho, Ariel S. Teles, Francisco J.S. Silva, and Luciano R. Coutinho. Towards clustering human behavioral patterns based on digital phenotyping. In *2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE, June 2021. doi: 10.1109/cbms52027.2021.00076. URL http://dx.doi.org/10.1109/CBMS52027.2021.00076.

[88] Wajid Rafique, Maqbool Khan, Nadeem Sarwar, Muhammad Sohail, and Asma Irshad. *A Graph Theory Based Method to Extract Social Structure in the Society*, page 437–448. Springer Singapore, 2019. ISBN 9789811360527. doi: 10.1007/978-981-13-6052-7_38. URL http://dx.doi.org/10.1007/978-981-13-6052-7_38.

[89] Farzad V. Farahani, Waldemar Karwowski, and Nichole R. Lighthall. Application of graph theory for identifying connectivity patterns in human brain networks: A systematic review. *Frontiers in Neuroscience*, 13, June 2019. ISSN 1662-453X. doi: 10.3389/fnins.2019.00585. URL http://dx.doi.org/10.3389/fnins.2019.00585.

[90] Nataliya Boyko, Hanna Komarnytska, Yurii Kryvenchuk, Yurii Malynovskyi, and Iryna Koval. Clustering algorithms for economic and psychological analysis of human behavior. In *International Workshop on Conflict Management in Global Information Networks*, 11 2019.

[91] Benjamin W. K. Hung, Anura P. Jayasumana, and Vidarshana W. Bandara. Finding emergent patterns of behaviors in dynamic heterogeneous social networks. *IEEE Transactions on Computational Social Systems*, 6 (5):1007–1019, October 2019. ISSN 2373-7476. doi: 10.1109/tcss.2019.2938787. URL http://dx.doi.org/10.1109/TCSS.2019.2938787.

[92] Agnideven Palanisamy Sundar, Feng Lilt, Xukai Zou, and Tianchong Gao. Deepdynamic clustering of spam reviewers using behavior-anomaly-based graph embedding. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*. IEEE, December 2020. doi: 10.1109/globecom42002.2020.9322330. URL http://dx.doi.org/10.1109/GLOBECOM42002.2020.9322330.

[93] Tom Menaker, Joke Monteny, Lin Op de Beeck, and Anna Zamansky. Clustering for automated exploratory pattern discovery in animal behavioral data. *Frontiers in Veterinary Science*, 9, June 2022. ISSN 2297-1769. doi: 10.3389/fvets.2022.884437. URL http://dx.doi.org/10.3389/fvets.2022.884437.

[94] Shyam Varan Nath. Crime pattern detection using data mining. In *2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops*, pages 41–44, 2006. doi: 10.1109/WI-IATW.2006.55.

[95] M A Syakur, B K Khotimah, E M S Rochman, and B D Satoto. Integration k-means clustering method and elbow method for identification of the best customer profile cluster. *IOP Conference Series: Materials Science and Engineering*, 336:012017, April 2018. ISSN 1757-899X. doi: 10.1088/1757-899x/336/1/012017. URL http://dx.doi.org/10.1088/1757-899X/336/1/012017.

[96] Andrea Zanellati, Stefano Pio Zingaro, and Maurizio Gabbrielli. *Student Low Achievement Prediction*, page 737–742. Springer International Publishing, 2022. ISBN 9783031116445. doi: 10.1007/978-3-031-11644-5_76. URL http://dx.doi.org/10.1007/978-3-031-11644-5_76.

[97] Karl Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 1*, 2:559–572, 1901.

[98] Trupti Kodinariya and Prashant Makwana. Review on determining of cluster in k-means clustering. *International Journal of Advance Research in Computer Science and Management Studies*, 1:90–95, 01 2013.

[99] Dhendra Marutho, Sunarna Hendra Handaka, Ekaprana Wijaya, and Muljono. The determination of cluster number at k-mean using elbow method and purity evaluation on headline news. In *2018 International Seminar on Application for Technology of Information and Communication*, pages 533–538, 2018. doi: 10.1109/ISEMANTIC.2018.8549751.

[100] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: Why and how you should (still) use dbscan. *ACM Transactions on Database Systems*, 42(3):1–21, July 2017. ISSN 1557-4644. doi: 10.1145/3068335. URL `http://dx.doi.org/10.1145/3068335`.

[101] Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery*, 2(2):169–194, 1998. ISSN 1384-5810. doi: 10.1023/a:1009745219419. URL `http://dx.doi.org/10.1023/A:1009745219419`.

[102] Andrew A. Neath and Joseph E. Cavanaugh. The bayesian information criterion: background, derivation, and applications. *WIREs Computational Statistics*, 4(2):199–203, December 2011. ISSN 1939-0068. doi: 10.1002/wics.199. URL `http://dx.doi.org/10.1002/wics.199`.