TEACHING AND LEARNING TO CONSTRUCT DATA-BASED DECISION TREES USING DATA CARDS AS THE FIRST INTRODUCTION TO MACHINE LEARNING IN MIDDLE SCHOOL

YANNIK FLEISCHER
Paderborn University
yanflei@math.uni-paderborn.de

SUSANNE PODWORNY
Paderborn University
podworny@math.uni-paderborn.de

ROLF BIEHLER
Paderborn University
biehler@math.uni-paderborn.de

ABSTRACT

This study investigates how 11- to 12-year-old students construct data-based decision trees using data cards for classification purposes. We examine the students' heuristics and reasoning during this process. The research is based on an eight-week teaching unit during which students labeled data, built decision trees, and assessed them using test data. They learned to manually construct decision trees to classify food items as recommendable or not. They utilized data cards with a heuristic that is a simplified form of a machine learning algorithm. We report on evidence that this topic is teachable to middle school students, along with insights for refining our teaching approach and broader implications for teaching machine learning at the school level.

Keywords: *Statistics education research; Data science education; Decision trees; Machine learning; Artificial intelligence; Middle school teaching*

1. INTRODUCTION

Data and machine learning (ML) methods are increasingly prevalent in various areas, leading to artificial intelligence (AI) applications that impact our daily lives. This trend has been accompanied by calls for integrating data science education and ML into school curricula (Engel, 2017; Ridgway, 2016). Engel (2017) advocated for cultivating informed and critical consumers of data-driven phenomena among students. Recent initiatives such as those addressing AI literacy (Long & Magerko, 2020) reinforced this goal and concentrated on enhancing understanding of AI and ML concepts. The recently published German Data Literacy Charter (Schüller et al., 2021) also emphasized the importance and necessity of fostering a critical and competent approach to data, data-driven decision-making processes, and AI applications based on data. It stated:

> In concrete terms, this [the rising importance of data science] requires the inclusion of data literacy in the curricular and educational standards of schools, teacher training, and higher education. Learners should not only be addressed as passive consumers of data. We rather enable them to actively shape data-related insights and decision-making. (p. 3)

Because data science and ML have so far seemed to require expert knowledge that may not be accessible to students (Sulmont et al., 2019a), our approach is to elementarize and contextualize these topics for young students. Within the ProDaBi project (prodabi.de/en), we have experience in teaching these topics to upper secondary level students (Biehler & Fleischer, 2021; Fleischer et al., 2022). Based on this experience, we developed a teaching unit for Grade 6 (students aged 11–12 years) to introduce the idea of data-based decision trees (DTs) with an unplugged activity using data cards with nutrition information (Podworny et al., 2021). Nutrition data is available on most food products so that even

young students know them from their daily lives. Predicting whether a food item is recommendable or not based on nutrition data is the central context of the teaching unit that aims to teach students how DTs are derived from data and used as predictors. We report on an interview study conducted after the teaching unit, in which we investigate the students' use of data cards and reasoning with their manually self-built, one-level DTs.

2. BACKGROUND

2.1. DECISION TREES IN MACHINE LEARNING

Machine learning, as described by Shalev-Shwartz and Ben-David (2014) and Hastie et al. (2009), is a heterogeneous field that includes different methods and learning algorithms for solving different types of tasks automatically. The unifying element between all methods is that they are based on training data. We focus on the subtype of supervised learning, specifically on classification tasks that can be addressed with DTs.

Classification involves the task of providing objects or individuals of a population with (ideally) correct labels with regard to a certain question. In statistics, a population is a set of similar individuals, objects, or events that are of interest to a particular question or statistical investigation (Kauermann & Küchenhoff, 2011, p. 5). Typical examples of classification problems are assigning a patient (individual) with a diagnosis (label) or classifying emails as "spam" or "not spam." The possible labels come from a label set, with the number of labels determined by whether the problem is binary classification (two possible labels) or multiclass classification (a finite set of more than two labels).

The task of a learning algorithm is to create a classifier that predicts a label for any given object in the population. To make an informed prediction, an object is represented by a set of characteristics displayed as a vector. Because the characteristics inform the choice of the predicted label, they are called predictor variables. The labels are values of a so-called target variable. The creation of a classifier is based on training examples, that is, objects from the population for which predictor variables' values and correct labels are known. A set of training examples is called training data. As a measure of success, the aim is to quantify the error of a classifier, which is the probability that it does not predict the correct label for a randomly chosen object from the population. Practically, the error is estimated by using test data to calculate the misclassification rate. Test data is structurally identical to training data but is not used to create the classifier. Instead, test data is used to evaluate classifiers.

Data trees, as a method in supervised learning, are algorithmically constructed from data (Breiman et al., 1984; Quinlan, 1993) to serve as classifiers. Especially when the tree is not too large, using a hierarchical tree structure makes the decision highly transparent and understandable.

2.2. DATA-BASED MACHINE LEARNING METHODS IN SCHOOL

Machine learning, and especially the topic of DTs, is taken up by different recent curricular frameworks (Bargagliotti et al., 2020; International Data Science in Schools Project Curriculum Team., 2019; Ministerium für Schule und Bildung des Landes Nordrhein-Westfalen, 2021). Because these topics are quite new to school teaching, it is not obvious what knowledge should be taught or how it can be taught to students at school. Considering the theory of didactic transposition in mathematics education (Chevallard & Bosch, 2014), we can distinguish between scholarly knowledge, knowledge to be taught, taught knowledge, and learned knowledge. Transposing scholarly knowledge to taught knowledge includes two steps: first, the identification of knowledge to be taught, and second, the construction of learning environments and activities, which represent taught knowledge. We will use this as a framework for presenting the literature on this topic in the following.

Knowledge to be taught. The didactic transposition of scholarly knowledge of supervised ML concerns different aspects, including data-based ML model building (data preparation, algorithms, model evaluation, etc.), ML phenomena (overfitting, bias, ML workflow, etc.), and responsible use of ML (applications, context-based evaluation, fairness, etc.).

With this paper, we contribute to the transposition process by focusing on teaching data-based ML model building to young students. Two approaches for teaching this model-building process can be

found in recent literature. The first approach is to leave the model-building process and the model completely as a black box (Hitron et al., 2019; Vartiainen et al., 2020). Another approach is to focus on one model type, such as DTs or neural networks (Mobasher et al., 2019; Santana et al., 2020; Schönbrodt et al. 2022; Zieffler et al., 2021), using a so-called white box approach (Mike & Hazzan, 2022) with the goal for students to understand the model type and the algorithm that creates the model. In the research mentioned above, the white box approach was chosen for older students (high school and older), and the black box approach was chosen for young students (13 years and younger). For instance, Hitron et al. (2019) investigated teaching ML to 10- to 13-year-old students. They let students prepare data and investigate models created by ML algorithms but did not exactly teach how the algorithms work to "maintain an accessible and understandable experience (by keeping processes black-boxed)" (p. 4). They had the legitimate concern that the ML algorithms are too difficult for young students to understand. However, they encouraged future research to investigate other scenarios in order to further uncover "carefully selected underlying processes" (p. 10) in the lessons, for example, the model building process in ML.

The knowledge to be taught at school "cannot be a reproduction of scholarly knowledge" (Chevallard & Bosch, 2014, p. 170) but should "preserv[e] its main elements" (p. 170) in a way that appears authentic and genuine. So, there is a need for an elementarized version of a model-building process that appears authentic and teachable. For data-based DTs, the requirements to understand the model building are relatively low compared to other model types (Engel et al., 2018), so we are curious whether building data-based DTs is teachable and useful for young students, which is what we investigate in this paper.

Taught knowledge. The design of taught knowledge includes developing "appropriate environments with activities" that students can interact with and that are "aimed at making the knowledge 'teachable', meaningful, and useful" (Chevallard & Bosch, 2014, p. 170). Our design decisions were influenced by findings from general learning theories, statistics education research, and computer science education research that are presented below. In concrete learning environments, approaches with hands-on activities grounded in constructivism are popular and often successful in teaching about ML (Casal-Otero et al., 2023; Martins & Gresse von Wangenheim, 2022). The concrete hands-on activities found in the literature vary between using interactive software and using unplugged approaches (Martins & Gresse von Wangenheim, 2022). In most published teaching modules, hands-on activities use digital tools that "are focused on ML, with the aim of demystifying this learning in K–12 education" (Casal-Otero et al., 2023, p. 9). Martins and Gresse von Wangenheim (2022) found that for older students, the most common tools are either well-structured visual environments that cover the model as a black box such as Google Teachable Machine (https://teachablemachine.withgoogle.com/) or text-based programming languages such as Python. The latter is not suitable for an average class of 11- to 12-year-old students. For our approach of not using a black box, there seems to be a need for a tool that makes it possible to uncover the model-building process of a DT for young students.

In statistics education, unplugged, card-based approaches are known when it comes to teaching about data (Harradine & Konold, 2006), especially when young students are involved. The use of card-based approaches can be combined with the method of teaching and learning statistics with embodied activities (Lakoff & Núñez, 2000). Hancock and Rummerfield (2020) found that the combination of software use preceded by unplugged hands-on activities (dice, cards, etc.) was more beneficial for student learning than having students just use software. For instance, using data cards enables students to "play the machine"—a concept from computer science education research (Futschek & Moschitz, 2010)—before using software to apply the same ML algorithm. In this study, we investigate what students learned from participating in our teaching unit in which we utilized a card-based approach.

3. TEACHING UNIT ON DECISION TREES: FRAMEWORK AND STRUCTURE

In the previous section, we identified a gap in research on teaching young students about ML by using a white-box approach of uncovering the data-based model-building process. Before describing the teaching unit, we introduce our data example and how data cards can serve as a tool for creating data-based DTs. Building on our work where we described a simplified DT algorithm for teaching at

the upper secondary level (Biehler & Fleischer, 2021; Fleischer et al., 2022), we developed a further simplified heuristic for creating data-based, one-level DTs tailored to 11- to 12-year-old students.

3.1. DATA EXAMPLE

The data we used was comprised of 55 cases of food items. Data about the "big 7" nutrition variables found on many food packages were collected by the authors for all food items. Table 1 shows the first five cases and the corresponding variables and values. The classification problem we regard is classifying food items as rather recommendable or rather not recommendable. In the teaching unit, we always used the extension "rather" to avoid making statements that were too strong. In this paper, we will only use "recommendable" because it reads simpler. For creating training data, the food items are labeled as recommendable or not before any nutritional data are considered.

Table 1. Data table with nutrition information per 100 g for food items (five example cases)

Food item	Energy (kcal)	Fat (g)	Of which Saturated Fat (g)	Carbohydrates (g)	Of which Sugars (g)	Protein (g)	Salt (g)
Apple	52	0.2	0.0	13.8	11.0	0.3	0.0
Avocado	160	13	2.8	2	0.7	1.5	0.1
Banana	95	0.3	0.1	21	12	1.1	0.1
Pretzel	295	5.4	0.3	51	2.7	8.9	2.1
Butter	743	82	55	0.7	0.7	0.6	0.0

We consider the set to be a good data example because it meets several requirements for teaching ML in Grade 6. First, we wanted to choose a data example with adequate complexity for the target group. The example should not be too large so that learners can still grasp it in its entirety and manage it with the available tool. Second, there should be enough opportunities to try out and explore different starting points for a DT. The exact number of 55 was chosen because the number fulfills these requirements, and it is a common card game size (material production reasons). Third, our example should be generalizable. This is the case for the example chosen because the classification problem is technically comparable to problems of image recognition or recommender systems in online platforms. Similarities exist in the sense that these problems also address the prediction of a categorical target variable based on a set of numerical predictor variables. Fourth, we wanted our example to be motivating for students. Therefore, choosing an example from the learners' environment is beneficial (Garfield & Ben-Zvi, 2008). In previous investigations, we found that 70% of students who attended the teaching unit in Grade 6 stated they liked or rather liked the example, and 23% were neutral towards the example (Podworny et al., 2022).

3.2. DATA CARDS AS A TOOL FOR CREATING DATA-BASED DECISION TREES

We use data cards as a supporting tool for creating DTs in a white-box approach (Mike & Hazzan, 2022). We propose an unplugged and manual approach so that learners can better grasp what happens automatically when they later use the computer (Hancock & Rummerfield, 2020). To create engaging learning resources, we transferred the data from Table 1 to data cards, as displayed in Figure 1.

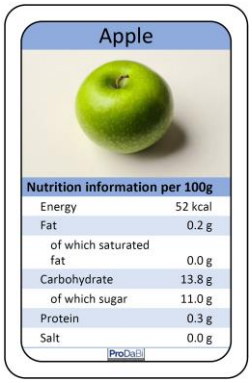

Figure 1. Example of a data card

To introduce the concept of using this tool, we will demonstrate how a small set of data cards can be utilized to construct data-based DTs manually. Following this demonstration, we will outline the corresponding teaching unit. The procedure we will introduce is a simplified version of professional DT algorithms that retains their core elements. The classification problem we are addressing involves classifying food items using either the label "recommendable" or "not recommendable" based on the provided nutrition values. That means we use a binary target variable and seven numerical predictor variables related to nutrition.

A DT is a hierarchical rule system that can be used as a classifier. An example of a DT for the described context is displayed in Figure 2, which can be utilized to classify the *Apple* from Figure 1. The DT is traversed from top to bottom, selecting branches based on fat and energy values. The first rule node checks for the fat value. Because the *Apple* contains less than 8 g of fat per 100 g, the left branch is followed, leading directly to a terminal node (leaf node). A terminal node contains a label, which is assigned to the object being classified. Thus, the *Apple* is classified as recommendable. For a food item with more than 8 g of fat, the right branch must be followed, and the energy value is evaluated in the subsequent level to reach a terminal node. This DT is merely an example without any claim that it classifies foods in a meaningful way. In principle, such a DT can contain any number of levels and predictor variables.

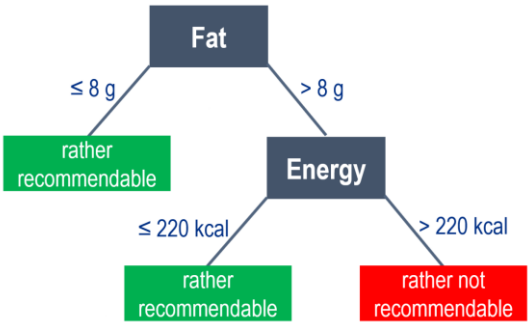

Figure 2. Example of a DT

The aim is to construct a DT for recommending foods using training data. To demonstrate how to utilize data cards to construct the DT, we have chosen eleven cards that represent different food items, as displayed in Figure 3. Initially, each case must be labeled. Here, we labeled the data cards for *Apple*, *Avocado*, *Fried egg*, *Noodles*, *Peas*, and *Slice of grey bread* as recommendable, which is represented by a green paper clip on each data card (see Figure 3). All other food items are labeled as not recommendable, indicated by a red paper clip. These eleven labeled data cards serve as training data.

To construct a data-based one-level DT as a classifier, a predictor variable is used to split the data by grouping cases according to the values of the variable. When dealing with a numerical predictor variable, a threshold is selected, and cases are assigned to either the subset below and including the threshold or the subset above the threshold. Subsequently, the one-level DT is built by using the

majority label in each subset as the decision for that subset. This process ensures that the majority (or at least 50%) of cases in each subgroup are correctly classified, and a minority may be misclassified. The total number of misclassifications across both groups serves as a measure of the quality of the one-level DT.

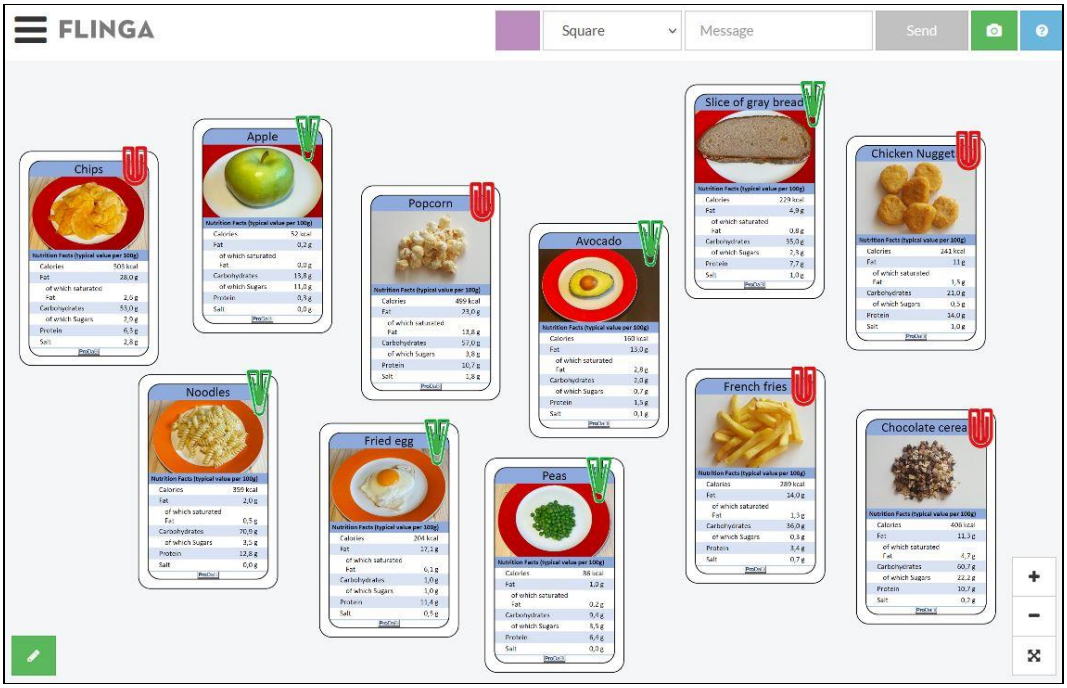

Figure 3. Eleven labeled data cards (Screenshot from flinga.fi)

Figure 4 displays an example of a data split for the predictor variable of fat with a threshold of 8 g. In the right branch are all food items with more than 8 g of fat, and in the left branch are food items with up to and including 8 g of fat. From that, a one-level DT is built by using majority labels in both subsets. Foods with fat of 8 g or fewer are thus labeled as recommendable, and foods with more than 8 g of fat are labeled as not recommendable. With this DT, two food items in the training data are misclassified, namely *Avocado* and *Fried egg* on the right-hand side because the majority label in that subset is "not recommendable" (with recommendable to not recommendable frequencies of 2 to 5).

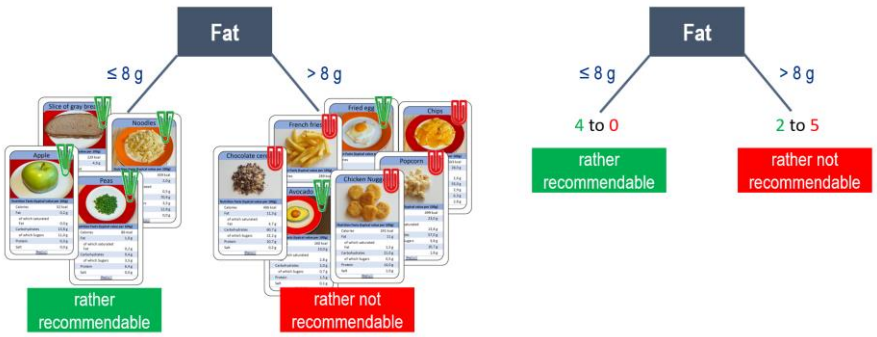

Figure 4. Data split with data cards (left), resulting in a one-level DT (right)

One core element of constructing data-based DTs is identifying the best one-level DT, which entails finding the most effective threshold for a given predictor variable to minimize misclassifications. How to do this with the data cards is exemplified in the following for the predictor variable of "fat per 100 gram." The first step is to sort the data cards in ascending order according to the variable's values, as shown in Table 2.

Table 2. Food items in ascending order with regard to fat per 100 g

Food item	Fat per 100 g	Label
Apple	0.2	Recommendable
Peas	1.0	Recommendable
Noodles	2.0	Recommendable
Slice of grey bread	4.9	Recommendable
Chicken nuggets	11.0	Not recommendable
Chocolate cereals	11.3	Not recommendable
Avocado	13.0	Recommendable
French fries	14.0	Not recommendable
Fried egg	17.1	Recommendable
Popcorn	23.0	Not recommendable
Chips	28.0	Not recommendable

An optimal threshold can be found by considering one threshold in all possible interspaces between two data cards and calculating the related number of misclassifications. Because we have eleven items with different values, we have to examine ten thresholds, each of which lies in an interval between two neighboring fat values (see Table 3). The ten "interspaces" between two neighboring values are identified with the letters A to J.

Table 3. Possible thresholds with number of misclassifications

ID	Data split between	Interval for threshold	Number of misclassifications (using majority labels)
A	Apple & Peas	[0.2, 1.0)	5
B	Peas & Noodles	[1.0, 2.0)	4
C	Noodles & Slice of bread	[2.0, 4.9)	3
D	Slice of bread & Chicken nuggets	[4.9, 11.0)	2
E	Chicken nuggets & Chocolate cereals	[11.0, 11.3)	3
F	Chocolate cereals & Avocado	[11.3, 13.0)	4
G	Avocado & French fries	[13.0, 14.0)	3
H	French fries & Fried egg	[14.0, 17.1)	4
I	Fried egg & Popcorn	[17.1, 23.0)	3
J	Popcorn & Chips	[23.0, 28.0)	4

According to this procedure, the best data split for the predictor variable of fat is given for a threshold in interval D [4.9, 11.0), yielding two misclassified items (see Figure 4). Intuitively, we might select the value of 8 that lies in the middle of the interval but choosing either the edge value of 4.9 or any other value within the interval is also valid. Professional DT algorithms use different solutions by calculating and choosing the mean value or choosing one edge value. To be accurate in this paper, we included the interval notation in Table 3, which is not known in Grade 6 so that students just choose a threshold between two given values. The above-shown procedure can be repeated for all predictor variables to find the overall best one-level DT according to the given data.

As a next step for building a multi-level DT, the subgroups of data that still contain misclassified cases are used to repeat the same procedure for finding the predictor variable (and threshold) for the next level of the DT. In the right branch displayed in Figure 4, the most effective data split is done using the variable of energy with the threshold of 220 kcal so that no item is misclassified afterward. The result is the DT in Figure 2. After achieving "pure" subsets, in which all cases have the same label such as in the left branch displayed in Figure 4, the construction process is terminated.

Compared to professional DT algorithms, some simplifications have been made in our procedure to favor accessibility for (young) students. For example, we use the number of misclassifications instead of the misclassification rate, entropy, gini index, or some other quality measure for data splits. These

more complicated measures have advantages in practice that can be neglected in school teaching. Moreover, in professional DT algorithms, additional termination criteria are set to avoid overfitting the training data. This could be the maximum depth of a tree or the minimum number of cases in a branch. However, we disregarded the topic of overfitting for Grade 6 because it is probably too challenging for 11- to 12-year-old students.

3.3. FRAMEWORK FOR A SIMPLIFIED HEURISTIC FOR CREATING A DECISION TREE

We described the elementarized process of finding the best one-level DT for a given predictor variable in detail. However, another simplification is needed to make the whole procedure more feasible in class. If all possible data splits for all predictor variables were considered, as indicated in Section 3.2, a resorting of the data cards would be necessary six times, and a total of 70 data splits (10 for every predictor variable) would have to be examined in this example with eleven cards. Moreover, the students use up to forty data cards as training data in class, so the number would be even bigger, and the resorting would be very time-consuming. Therefore, to enhance practicality, the need to explicitly consider every possible data split for all variables can be reduced.

In our simplified heuristic, we propose a blend of formal and informal aspects that resemble a professional DT algorithm. Instead of exhaustively exploring all possible data splits, students initially choose a threshold for a data split, which may be arbitrary, and then seek to refine their results by iteratively searching for better splits. They consider different splits sequentially, selecting a threshold, using it for a data split, deriving a one-level DT using majority labels, and assessing the new DT by comparing the number of misclassifications with the previous one. In this iterative process, the best rule found is stored and compared to possible new rules. This cycle is repeated until students decide to terminate the process and select their final DT for representation. We identify four formal components of understanding within this process.

Component 1: Performing a data split with a variable and a certain threshold. The first component involves understanding how to perform data splits based on predictor variables to create two subgroups. This is achieved by selecting a threshold for a given predictor variable and then grouping cases with values up to and including the threshold and grouping cases with values greater than the threshold.

Component 2: Deriving a one-level decision tree from a data split. The second component involves understanding how to derive a DT from a data split. After dividing the data into two subgroups using a predictor variable and threshold, this is accomplished by using majority labels in each subgroup to classify the assigned items.

Component 3: Evaluating a decision tree. DTs constructed as described above can be evaluated in different ways. A straightforward and intuitive approach for Grade 6 students is to count the number of misclassified cases in each subgroup and sum them to determine the total number of misclassifications. This total sum allows for comparing different DTs.

Component 4: Representing a decision tree. After selecting a certain DT, the next step involves representing and communicating it. This can be achieved through written form, verbal form, or any other form of representation.

Some aspects of the process, such as selecting the initial threshold, selecting subsequent thresholds/data splits, and deciding when to stop, are not formally instructed. Instead, they serve as implicit differentiation opportunities, allowing students to decide the duration and depth of their engagement with the process. After at least two DTs are considered, they can be compared to select the best one. If students are satisfied with a (low) number of misclassifications, they can terminate the process, select the corresponding DT, and represent it.

Figure 5 represents the heuristic that we had in mind when designing the teaching unit, with the four formal components (white) and two informal components (shaded). We had two criteria for the development of the framework: (a) the heuristic is performable (by young students) with reasonable effort using data cards manually, and (b) the core elements of professional algorithms are preserved so

that they can serve as a starting point for understanding how DTs are created automatically using algorithms. We use this framework in two ways: (a) it structures the teaching unit because we teach the different components first and let the students explore how to use them, and (b) it is the subject of study because we want to investigate whether the students apply a heuristic and, if this is the case, how they interpret the parts that are not prescribed (shaded components in Figure 5).

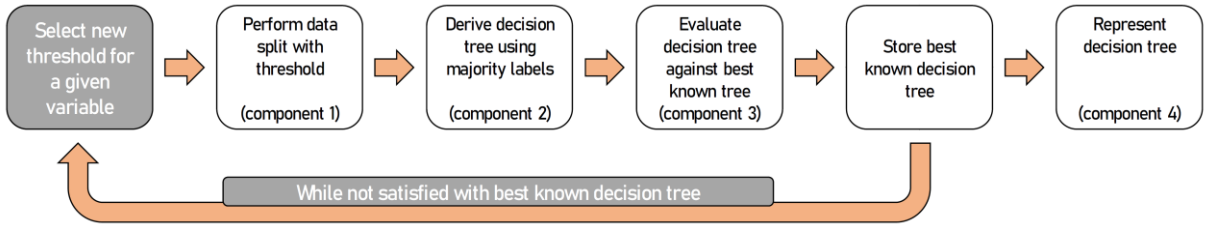

Figure 5. Flowchart for the process of creating a one-level DT with prescribed components (white) and non-prescribed components (shaded)

3.4. THE TEACHING UNIT

The above framework was used to design a teaching unit on the ML method of data-based DTs as a mainly unplugged activity. The teaching unit consists of eight lessons, 45 minutes each. In the following, details about the first five lessons (Part 1–4) are described because the content relevant to the study was taught there. The contents of the last three lessons (Part 5–7) are only briefly summarized.

Part 1: Introducing the context and data preparation (2 lessons). Initially, students are introduced to the nutrition context. Following this, they work in pairs to label food items using a worksheet containing pictures of the items without nutrition data. After that, a class discussion moderated by the teacher follows, where all pairs agree on labels. Each pair receives a deck of data cards to label as recommendable (green paperclip) or not recommendable (red paperclip) to prepare the training data with the agreed-upon labels. This process yields a training deck of up to 40 (usually about 30) cards, with potentially controversial food items such as the *Fried egg* set aside. Including data preparation in student activities is advocated by Hitron et al. (2019).

Part 2: Deriving a one-level decision tree from data with an embodied experience (1 lesson). To foster understanding of a data split and how to derive a one-level DT, an embodied experience (Lakoff & Núñez, 2000) is utilized. Each student is assigned a food item, and the student used to represent the data card for the item. The focus of this phase is to introduce the students to the four formal components of the framework (see Figure 5). The objective is to find a data split where the two resulting groups are pure. The teacher selects a variable and threshold, instructing students to move to one side of the classroom based on whether the value of the food item they represent is above or below the threshold (Component 1). Subsequently, students use hand signals to indicate the number of recommendable and not-recommendable food items that are included in each subgroup, with the majority determining the subgroup's label (Component 2). This process is repeated three times and recorded (Component 4) to evaluate which resulting DT best minimizes misclassifications (Component 3).

Part 3: Constructing a one-level decision tree using data cards (1 lesson). After the students have learned the formal components of the framework, they know how to derive and compare DTs using data cards. They then work in pairs with all labeled data cards to find good DTs and to develop their own heuristic. Each pair is assigned a predictor variable to ensure all variables are utilized in class at least once. At this point, sorting the data cards in ascending order for a specific variable is introduced as a strategy that allows for comparing different data splits with low effort. The exact procedure for considering data splits and when to terminate the process is left open. Students can experiment with multiple thresholds to get a feeling for which data splits are promising.

In the end, pairs share their final choices in class, allowing the entire class to collectively assess and select the best rule. This collaborative process ensures thorough consideration of all variables, leading to the establishment of the best one-level DT.

Part 4: Creating a multi-level decision tree with data cards (1 lesson). In Part 4, the students create multi-level DTs. Because there is no combination of variable and threshold that leads to a "perfect" one-level DT without any misclassifications, the idea of adding a second level to the DT is introduced. The procedure of creating a DT is repeated for the subgroups of data. The students can decide whether they continue with the one-level DT they created earlier or whether they use a different (potentially better) first rule from another pair. They then freely select which variable to proceed with. Each pair of students is tasked with creating at least a two-level DT, documented in a worksheet as depicted in Figure 6 (Component 4). Fast-working pairs are encouraged to create a third level or to consider different variables. Ultimately, all student pairs compare their two or three-level DTs, assessing them based on the number of misclassifications (Component 3).

Part 5: Applying a decision tree to new items (1 lesson). Every student uses a new data card to run through all created DTs that are exhibited in class. That process uncovers that different trees can make different decisions for a certain food item, which reveals uncertainty in DTs and motivates the idea of testing.

Part 6: The idea of test data (1 lesson). All DTs are tested with a given set of fifteen test data cards (half the number of training cards) using the number of misclassifications as the evaluation criterion.

Part 7: Experiencing an automatically created decision tree (1 lesson). In the last lesson, the transition is made to what a computer-based ML algorithm would do. Based on the simplified approach the students learned with the data cards, they are introduced to the systematic procedure of an ML algorithm. Students then interact with an ML algorithm in a specifically designed digital environment.

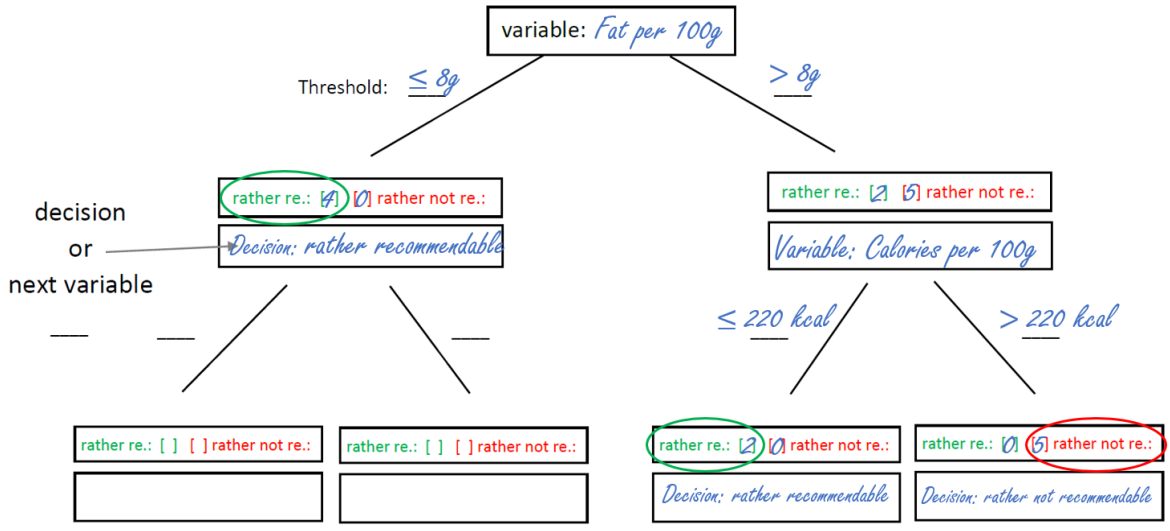

Figure 6. Documenting a decision tree on a worksheet

4. RESEARCH QUESTIONS

This study explores how students engage in constructing DTs after participating in our teaching unit. We examine what they learned from engaging with the unit content, what their own ideas are, and any obstacles they encounter in the process. In the terminology of Chevallard and Bosch (2014), we want to assess the learned knowledge that was acquired based on the taught knowledge. We are interested in students' use of the data cards and their reasoning during the process. As mentioned above,

the procedure we offer to students is not completely formally prescribed because it is left to the students which data splits they consider and when they terminate the process. We are especially interested in how they interpret these parts of the procedure and the corresponding reasoning. Our research question is:

- How do students approach deriving a one-level decision tree from data?

We pose the following sub-questions to answer the research question:

- How do students utilize the data cards to derive a one-level decision tree?
- How do students select which data splits to consider?
- What criteria do students use to reason when creating a decision tree based on data?

5. METHOD

5.1. IMPLEMENTATION OF THE TEACHING UNIT

The teaching unit was taught in two classes from different German middle schools (Gymnasien) in the summer of 2021 during a pandemic situation with changes between in-class and distance teaching. Two teachers at two schools taught the teaching unit to sixth graders in computer science classes. The teachers got a 60-minute introduction to the material from the ProDaBi project team and then taught the lessons using the prepared material on their own.

5.2. PARTICIPANTS

Fourteen students from two classes volunteered to participate in pairs for the interview study. Participants were comprised of six females and eight males aged 11–12 years. Pairing was done by the students themselves, resulting in three female pairs and four male pairs. Each pair was familiar with each other from their classes. Due to the volunteer situation, we do not have a representative selection of participants from the two classes. We assume those who volunteered felt comfortable with the teaching unit and with being interviewed. Nevertheless, for an exploratory study, seven pairs are a good basis for getting insights into students' heuristics and reasoning with DTs.

5.3. STUDY AND TASK DESIGN

The study was an exploratory interview study to get insights concerning the research questions. Due to the distance between the schools and the researchers and because of the pandemic situation, we chose an online setting for the interviews. Students were used to using a video conference system such as Zoom (https://zoom.us/) or Microsoft Teams in their classes, so we chose to use Zoom as the platform for the interviews along with an application called Flinga (https://flinga.fi/) that allows interaction with data cards directly via a browser. The prepared Flinga environment looked like the image in Figure 3. Flinga provides an environment supporting data card movements that are close to real-world data card usage. Additional drawings and markings can be implemented by using several tools such as a pencil (bottom left of Figure 3) or a labeled square (top of Figure 3). None of the participants had prior experience with Flinga.

The study was comprised of interviews in which pairs of students were tasked with creating a one-level DT using the predictor variable of "fat per 100 g" with the provided data cards in a Flinga environment. The data example with eleven cards from Section 3.2 is the same one we used for the task of the interview study. The students were asked to explain their actions as they proceeded. The thinking-aloud approach (Ericsson & Simon, 1980) was used to encourage students to share their reasoning. Interviews lasted between 12 and 20 minutes per pair. The concrete opening prompt was as follows:

- Here (opening the Flinga environment), you can see eleven data cards like those you know from class. Please show how you go about creating a one-level DT for classifying food items as recommendable/not recommendable using the variable fat. Explain what you are doing as you do it.

Three optional prompts were prepared if needed:

- Prompt 1: You are allowed to move the cards.
- Prompt 2: Remember the lessons, how did you proceed?
- Prompt 3: Try to proceed by choosing a threshold.

After the students finished, they were asked:

- Which threshold did you choose, and why did you choose this threshold?

In the teaching unit, the students worked on similar tasks but not on this exact task. The solution process described in Section 3.2 can be seen as an ideal solution.

5.4. DATA AND METHODS OF DATA ANALYSIS

The seven online interviews were recorded, capturing both the verbal communication and on-screen activities of the students. Transcripts were generated from the recordings, which included the dialogues and screenshots from the Flinga activities. Transcript analysis was based on the framework outlined in Section 3.3 and utilized qualitative content analysis (Mayring, 2015) to examine the data in relation to the research questions. This method involved a systematic and rule-based analysis to identify patterns in the students' approaches and draw conclusions about students' heuristics for creating a DT. A category manual was developed for the analysis, which is explained next.

In the analysis, our focus was on understanding the process of students in creating a one-level DT, particularly for components not formally taught (shaded components in Figure 5). We examined how students selected their thresholds (shaded box in by analyzing their actions with the data cards, the thresholds they considered, and their reasoning behind considering or rejecting thresholds. This reasoning provided insights into their evaluation process and criteria for choosing a final DT. We developed three categories to identify these critical points in the process. We captured occurrences of students' interactions with the data cards with Category 1, identified mentions of new thresholds (or data splits) with Category 2, and recorded reasoning about the threshold (or data split) with Category 3. These categories corresponded to the three sub-research questions. For transparency, we also documented the interviewer's interventions in a fourth category (see the first column in Table 4).

Table 4. Category manual

Category	Code	Description
1. Interacting with data cards	1.1	Sorting data cards ascending to a specific variable
	1.2	Grouping data cards according to a pre-selected threshold
	1.3	Grouping data cards according to another criterion
	1.4	Looking at the values without moving the data cards
2. Considering a threshold for a data split	2.1.i	Mentioning new threshold (interspace between two items)
	2.2.i	Selecting final threshold (interspace between two items)
	2.1.v	Mentioning new threshold (concrete value)
	2.2.v	Selecting final threshold (concrete value)
3. Reasoning about a threshold/ data split	3.1	Reasoning with the number of misclassifications
	3.2	Reasoning with the purity/impurity of subgroups of data
	3.3	Reasoning with content knowledge about food
4. Interventions of the interviewer	4.1	Prompt 1 about moving data cards
	4.2	Prompt 2 about remembering the lessons
	4.3	Prompt 3 about choosing a threshold
	4.4	Posing the follow-up question

Using codes in the transcripts, we marked critical points, offering a concise overview of the heuristic used by the students. The codes were developed both deductively, based on our expectations of knowledge to be taught, and inductively, based on observations from the interviews. The categories and resulting codes are presented in the category manual in Table 4. Some codes may require further

explanation. For instance, applying the Code 1.4 indicates that students interacted with the data cards without rearranging them; they simply examined the cards and compared different values while the cards remained in their original position. After the students rearranged the cards, this code was no longer applied. The codes related to reasoning in Category 3 will be illustrated in Section 6. Additionally, the codes in Category 4 corresponded to the prepared optional prompts and the follow-up question outlined in Section 5.3.

After applying the category system to all transcripts, we visualized a chronological sequence of all relevant events during each pair's process of creating a one-level DT (see Figure 7 & Figure 9). For every pair, we arranged all codes given in the respective transcript chronologically from left to right. All codes from Category 1 are displayed in green, Category 2 in red, Category 3 in yellow, and Category 4 in blue. In Category 2, dark red indicates that an interspace was considered as a threshold, and light red indicates that a value was considered as a threshold. The ten interspaces between the fat values of neighboring food items (ascending order) were denoted by letters from A to J, as indicated in Table 3.

6. RESULTS

We divided the student pairs into two subgroups based on their general strategy for utilizing the data cards. Five pairs employed a strategy centered around sorting the cards, whereas two pairs repeatedly grouped and regrouped the cards. To provide a detailed insight into the approaches, we delve into the processes of specific pairs.

6.1. RESULTS FOR STUDENTS WITH A SORTING STRATEGY

First, we describe the heuristics we found for the five pairs (displayed names are pseudonyms) shown in Figure 7. The following descriptions are structured by the four categories of the analysis.

Ben, Theo	1.1	2.1.i	1.2	3.1	2.1.i	3.1	2.2.i	1.2	2.2.v	4.4	3.1
	sort	D	group	miscl	G	miscl	D	group	8	quest	miscl
Karoline, Corinna	1.1	2.1.i	3.2	2.1.i	2.1.v	1.2	3.1	2.1.v	2.2.v	4.4	3.1
	sort	G	purity	D	5	group	misc	10	5	quest	misc
Vincent, Leo	1.1	2.1.i	3.1	2.2.v	4.4	3.2	3.1				
	sort	D	misc	8	quest	purity	misc				
René, Steffen	1.1	2.1.i	2.2.v	3.1	4.4	3.2					
	sort	D	5	misc	quest	purity					
Gina, Pia	1.1	2.1.i	2.1.v	1.1	2.2.v	4.4	3.1				
	sort	D	7	sort	7	quest	misc				

Figure 7. Processes of creating a one-level decision tree with the data cards—Five pairs

These five pairs all started with sorting the data cards according to the values of fat (Code 1.1). Only Gina and Pia initially made a sorting error (swapped *Chicken nuggets* and *Chocolate cereals*), which they later rectified through resorting, resulting in Code 1.1 occurring twice for Gina and Pia.

When considering thresholds, all five pairs initially examined interspaces (Code 2.1.i) between two data cards and not concrete values. Four pairs began with interspace D, located between *Slice of grey bread* and *Chicken nuggets* (see Figure 8), as a potential threshold, whereas one pair started with interspace G situated between *Avocado* and *French fries*. All five pairs ended up selecting a concrete value within the range of 4.9 to 11 in interspace D (Code 2.2.v) as their threshold. A threshold in interspace D is optimal because it results in the lowest possible number of misclassifications (Code 2). The pair of René and Steffen and the pair of Ben and Theo explicitly stated that they chose the best data split, but it is uncertain whether the other three pairs were aware of this optimization.

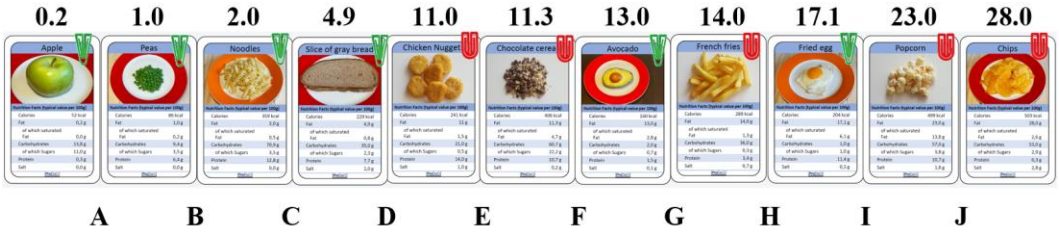

Figure 8. Data cards sorted in ascending order for the variable of fat per 100 g and interspace identifiers A–J

Regarding the reasoning behind the choice of a threshold, the predominant approach was to base the choice on misclassifications (Code 3.1), which was observed nine times across all groups. Each group based their choice on misclassifications at least once. For instance, Ben and Theo solely relied on misclassifications (Code 3.1) for their reasoning, as demonstrated in the following excerpt:

Ben: I think it would be smart to put a cut HERE between the chicken nuggets and the slice of grey bread. Then, on the left side, [the majority of items] is recommendable, and on the other side, [the majority] is not recommendable. That way, one would classify the least items wrongly, no?

Theo: Yes, so you would then have two recommendable items on the wrong side. If one would separate now, however, between avocado and fries. (…)

Ben: Yes (...) Then, there would be three wrong ones.

Other pairs demonstrated reasoning based on the purity or impurity of the subgroups (three instances of Code 3.2). This type of reasoning is based on the idea that the goal of data splitting is to create subgroups of data that are as pure as possible, ideally completely pure. Students either aimed to achieve complete purity or to avoid impurity, ensuring that each subgroup contained only one type of label (green or red). For instance, René justified their threshold choice by emphasizing the purity of the resulting subgroups (Code 3.2) without mentioning misclassifications. He highlighted that all cards with red labels were on the right side, whereas only those with green labels were on the left side:

René: That was pretty much the threshold, where then the not recommendable items were perfectly on the right side, so on the unhealthy side, and on the left are only green items.

Whereas Ben and Theo explicitly built a DT first and then calculated the number of misclassifications as a criterion, René and Steffen argued by inspecting the appearance of the subgroups of data without building a DT. This illustrates different approaches to argumentation, both of which are valid. Two pairs exclusively communicated reasoning with misclassifications, whereas three also communicated reasoning with the purity of subgroups, once each. None of these pairs required any additional prompting from the interviewer to select their final threshold. After the interviewer posed the follow-up question (Code 4.4), all five groups gave reasonable explanations by referring to misclassifications (Code 3.1) or to the purity/impurity of the subgroups (Code 3.2).

In the following, we will take a closer look at the students' processes and relate them to our framework.

Processes for the two pairs: Ben and Theo, Karoline and Corinna. From the processes used by the five pairs in Figure 7, we can again form two subgroups: Two pairs at the top and three pairs at the bottom. The two pairs at the top explicitly went through the process of finding a DT as we would expect in an optimal student solution. We illustrate this process using Ben and Theo as an example. Based on their communication, we inferred that they followed the heuristic while designing the teaching unit.

Initially, Ben and Theo sorted the data cards (Code 1.1) in ascending order based on fat values. They then chose interspace D as their initial threshold (Code 2.1.i) for data splitting (Component 1), constructed a DT via majority labels (Component 2), and assessed the DT using the number of

misclassifications (Code 3.1; Component 3). These three steps correspond to one iteration of the cycle described in Figure 5. They proceeded by considering another data split (Code 2.1.i), using interspace G to initiate a new iteration of the cycle. Again, they evaluated the resulting DT based on misclassifications (Code 3.1). Subsequently, they decided to end the process by selecting interspace D as their choice (Code 2.2.i), specifying a threshold value (Code 2.2.v) within the range of 4.9 to 11, and communicating their final DT (Component 4). Throughout, they utilized the data cards actively to build subgroups (Code 1.2) and visualize their data splits. Overall, Ben and Theo completed two iterations of the cycle, explicitly considering two data splits and opting for the better one based on misclassifications. In the end, they communicated their termination criterion by stating that they could not find a better data split in terms of misclassifications.

Processes for the three pairs: Vincent and Leo, René and Steffen, Gina and Pia. The process observed for the bottom three pairs in Figure 7 appeared shorter based on their explicit communication. These pairs initially sorted the cards (Code 1.1) and proceeded to consider interspace D (Code 2.1.i) (Component 1). After evaluating their rule (Code 3.1), they quickly decided to select their final threshold (Code 2.2.v). Gina and Pia did not provide reasoning for their threshold selection until prompted by the interviewer (Code 4.4). For these pairs, we observed only one iteration of the cycle described in the framework. None of these three pairs explicitly mentioned considering another data split before the interviewer's follow-up question (Code 4.4). It is possible that they mentally compared different splits without verbalizing it, or they simply selected one without feeling the need to compare it with others. Their responses to the follow-up question about why they chose their DT provided some insight into this. Regarding René and Steffen, we interpret that they made more comparisons than they communicated explicitly.

Steffen: So, it was practically the best result we could achieve with fat only.

Steffen stated that he and René found the best result, which could mean that they considered different solutions. Gina and Pia may have made similar comparisons. They stated that with the variable of fat, they could not isolate all green items because *Avocado* and *Fried egg* had too much fat. For Vincent and Leo, we did not see similar indicators of whether they considered other data splits.

6.2. RESULTS FOR STUDENTS WITH A GROUPING STRATEGY

The two pairs whose processes are displayed in Figure 9 used a different approach. We first describe the general differences in their processes compared to the processes of the five pairs above, again structured by the four categories of the analysis. Then, we will go into more detail on their concrete actions and reasoning.

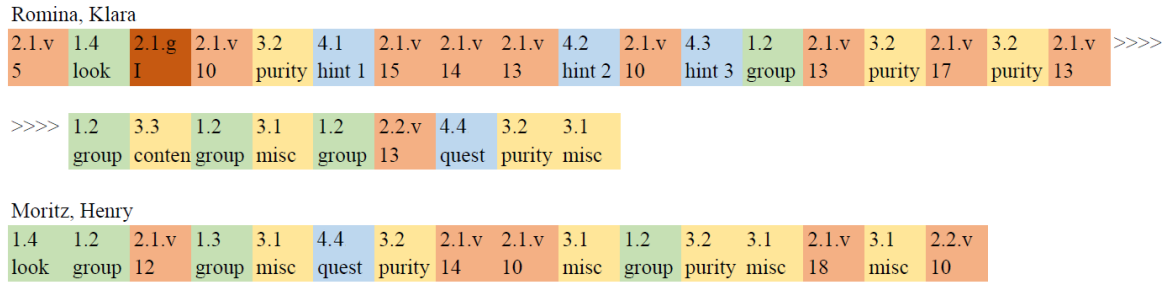

Figure 9. Processes of creating a one-level decision tree with the data cards—Two pairs

Unlike the five pairs discussed previously, they did not sort the cards in ascending order at any point in the process. Instead, they focused on creating two subgroups based on a pre-selected threshold (Code 1.2) and explored various threshold options. When considering a threshold (Component 1), both groups communicated that they wanted to make sense of the values on the cards first to come up with a good

threshold, but they did not move the data cards during this process (Code 1.4) or define what a "good" threshold was.

With one exception, these two pairs only referred to concrete values as thresholds (Code 2.1.v) and did not refer to interspaces. Without sorting the cards, it is difficult to recognize the interspaces between two neighboring fat values because they are not easily apparent.

When reasoning about the choice of a threshold, both groups reasoned by referring to the number of misclassifications (six times Code 3.1) and the purity/impurity of the subgroups (six times Code 3.2). For Romina and Klara, the dominant way of reasoning was impurity, even though this was not intended during the teaching unit, and for Moritz and Henry, it was the number of misclassifications.

In contrast to the five pairs whose processes are displayed in Figure 7, these two pairs did not reach their final solution without the interviewer's intervention. Romina and Klara required all three prepared prompts (Codes 4.1, 4.2, 4.3) to progress, whereas Moritz and Henry did not receive these prompts but initiated another cycle due to the follow-up question (Code 4.4). Next, we delve into the detailed processes of the two pairs to analyze the obstacles and challenges they encountered.

Moritz and Henry. Moritz and Henry started by looking at the values without moving the data cards (Code 1.4) and then proceeded to group them into two subgroups (Code 1.3), as illustrated in Figure 10, without communicating a threshold.

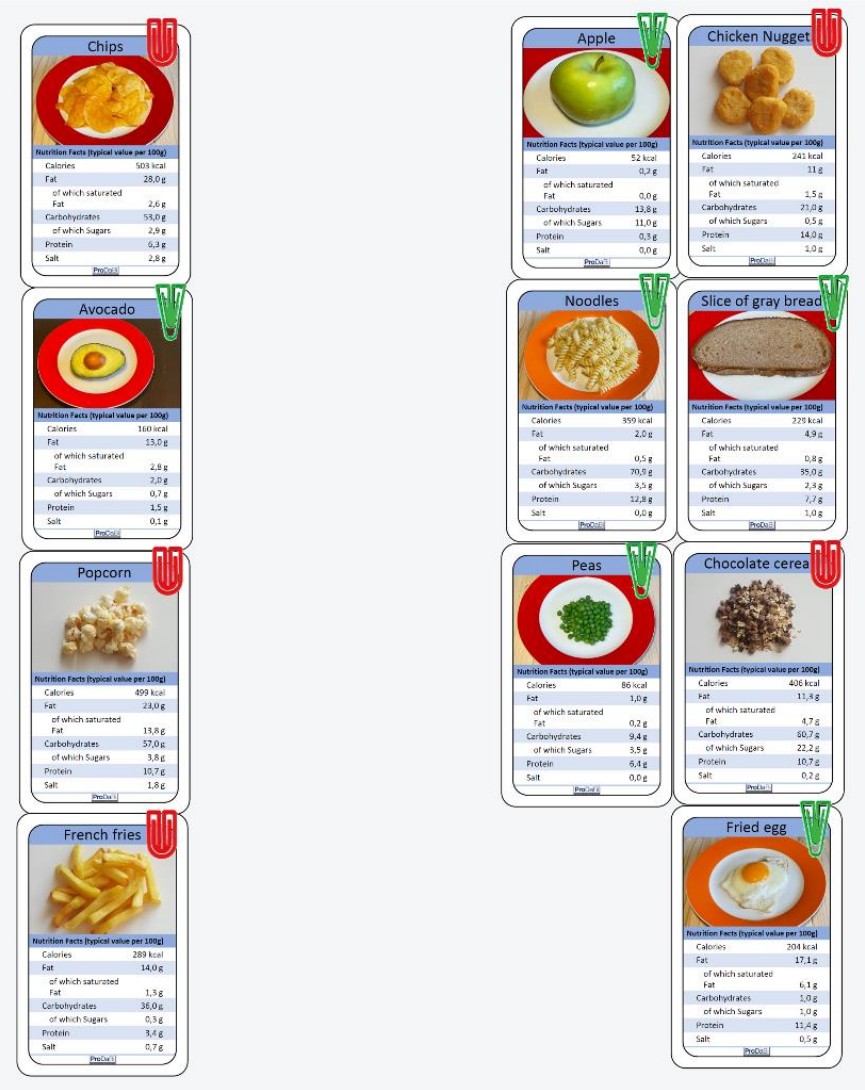

Figure 10. First attempt at grouping data cards by Moritz and Henry

After looking at the arrangement in Figure 10 for a while, Moritz and Henry decided to use 12 as a threshold because all items on the left had fat values higher than 12 g, whereas on the right, all values were lower than 12 g, except for the *Fried egg*. They explicitly noted "12" as a threshold in Flinga and rearranged the cards accordingly (Code 1.2), as depicted in Figure 11. We interpret that their initial loose grouping (Figure 10) informed the selection of a threshold, which they then applied strictly to refine their grouping.

Moritz and Henry created a unique visualization by arranging the data cards into two rows in both subsets, ensuring that all misclassified cards were in the lower row (Code 1.3). This approach, not suggested by the teacher, offers visual support for DT evaluation by highlighting potential misclassifications. It lacks, however, support for further threshold refinement due to the cards not being sorted by fat values. Nonetheless, Moritz and Henry positively evaluated their threshold choice of 12 despite four misclassified cards:

> Moritz [With the threshold of 12], we have four correctly assigned and two wrong on one side. This is good so far because we have two wrongs on one side and two wrongs on the other. More correctly assigned in each group. A total of seven out of eleven. That's, yes, a good score. So, more than half in any case.

Figure 11. Regrouping of Moritz and Henry after choosing the threshold 12

Until then, Moritz and Henry conducted one iteration of the cycle in the framework, and after terminating the process, they even made a useful representation. They were satisfied because more than half of the items were classified correctly.

However, after the interviewer posed the follow-up question (Code 4.4), they started to question their solution. They considered other thresholds near the initial one (10, 14, and 18) (Code 2.1.v) and evaluated them based on misclassifications (Code 3.1). They adjusted the grouping of cards accordingly (Code 1.2) to visualize new data splits. Ultimately, they opted for 10 as the best threshold (Code 2.2.v), which aligned with the lowest misclassification number. Despite not sorting the cards, Moritz and Henry demonstrated proficiency in all four components of understanding (build data split, derive one-level DT, evaluate DT, represent DT). Their strategy involved more effort due to multiple regroupings and lacked an easy overview of various data splits, potentially leading to initial satisfaction with a suboptimal threshold of 12. Overall, their approach mirrored that of the previous five pairs by initially selecting a threshold and exploring nearby options.

Romina and Klara. Initially, Romina and Klara just looked at the values without moving the data cards (Code 1.4) and suggested different thresholds (5, 10, 13, 14, and 15) seemingly arbitrarily and without clear justifications (Code 2.1.v). The first two optional prompts (Codes 4.1, 4.2) did not enhance their process. We interpret that they had difficulties getting started because they used the strategy of grouping the data cards with a pre-selected threshold. They needed a threshold to begin rearranging the

cards, but without rearranging them, they could not come up with a threshold. This was somewhat circular. After the prompt to select a threshold (Code 4.3), they chose 10 without clear justification and began grouping the data cards (Code 1.2). Before finishing the grouping, they switched to a threshold of 13. In this process, they incorrectly placed *Chicken nuggets* (11 g fat) in the wrong group on the right (see Figure 12). Their reasoning during threshold consideration was as follows:

Klara: Yes. I would say 13 because then I think we have almost all of them [the green cards on the left side].
Romina: Yeah, the fried egg is still missing, then, but I think over 17 is a little bit too much.
Klara: Yes, then we have the chocolate cereal [on the left side] again. That's bad.
Romina: Yeah. Ah, but if we take 13, we have the chocolate cereal in it anyway.

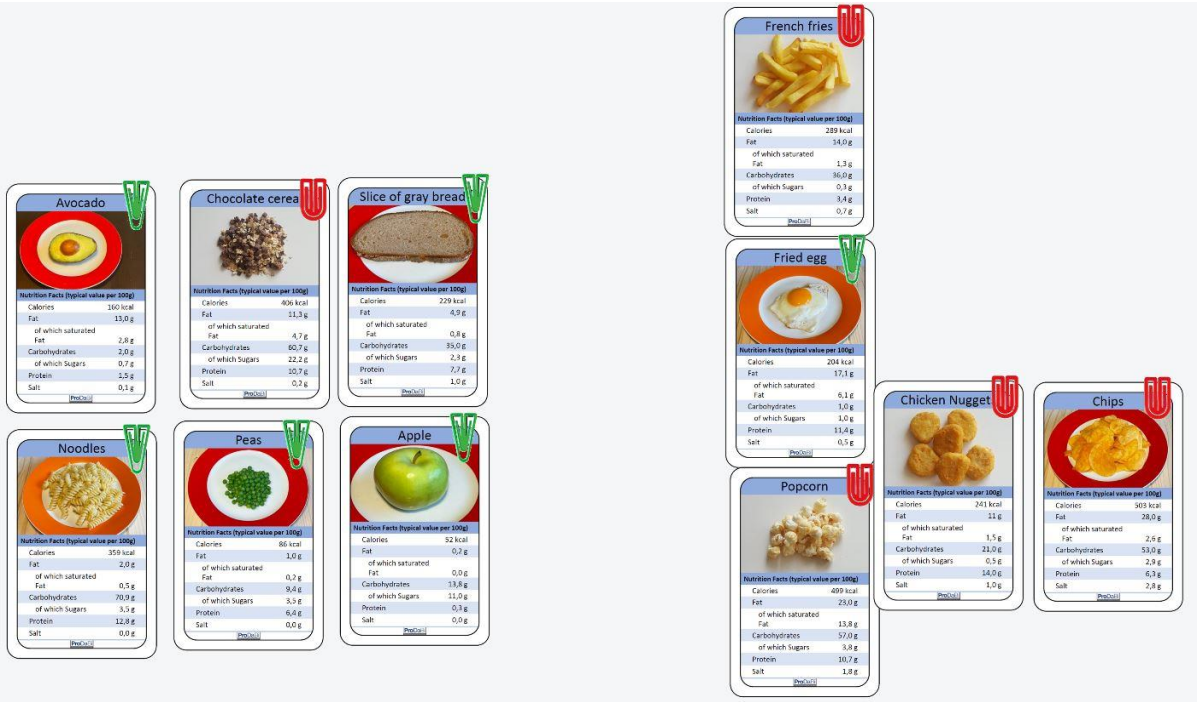

Figure 12. Grouped data cards for a threshold of 13 from Romina and Klara (putting chicken nuggets on the wrong side)

When Romina and Klara first considered 13 as a threshold to move the *Avocado* (green, 13 g) to the left, they initially were not aware that this would imply moving *Chocolate cereals* (red, 11.3 g) to the left as well. Subsequently, they considered 17 to shift the *Fried egg* (green, 17 g) to the left, aiming for all green items on the left side. They rejected 17 because Romina argued that this would mean bringing *Chocolate cereals* (red) to the left. Romina's argument that they had *Chocolate cereals* on the left anyway with a threshold of 13 was not accepted as an argument to choose 17. The actual disadvantage of 17 would have been that *French fries* (red, 14 g) had to be moved to the left, but this was not discussed. It was difficult for Romina and Klara to justify their choice because they did not use a clear criterion such as the number of misclassifications. During the process of considering different thresholds, Romina and Klara mostly reasoned by referring to the purity of subgroups and avoiding impurity (3.2), which is illustrated by them trying to get *Fried egg* (green) to the left side and trying to avoid getting *Chocolate cereals* (red) on the left side. Their implicit aim seemed to be to have all green cards on the left side and all red cards on the right side.

Tentatively, Romina and Klara decided on 13 as the threshold without a clear justification. After forming the subgroups displayed in Figure 12, they reassessed the threshold, this time considering the number of misclassifications (Code 3.1). They compared thresholds of 13 and 10 and found two misclassifications for each, leading them to retain 13 as their final choice. This interpretation was

correct for the grouping they made but finding only two misclassifications for a threshold of 13 was based on mistakenly putting *Chicken nuggets* on the wrong side.

Overall, after receiving different prompts, Romina and Klara demonstrated basic elements of the framework cycle. They considered several data splits, tried to evaluate them, and compared them to each other to select a final threshold. However, the second and third components of the framework (build one-level DT, evaluate DT) were not performed as systematically as by the other groups. They did not argue consistently with a clear criterion for evaluation during the whole process of comparing thresholds. Moreover, they did not sort the cards at any point but regrouped again and again when changing the threshold and made small mistakes in grouping the cards. However, if we disregard all these shortcomings, the basic approach of starting with one threshold to do a data split and then looking for alternative thresholds "nearby" was similar to the other groups.

6.3. SUMMARY OF RESULTS FROM THE PERSPECTIVE OF THE RESEARCH QUESTIONS

Looking at the results described above, a first remark is that all student pairs were able to utilize the data cards to find a solution for the task by creating a reasonable DT. Six of seven pairs even chose an optimal solution. Five pairs did not need any intervention, one pair got one intervention, and one pair got many interventions. Whereas some pairs used the cards to perform a well-structured heuristic (similar to Figure 5), the weakest pairs were at least able to use the cards to perform the basic steps of deriving a one-level DT (using the formal components).

Use of data cards in the process of deriving a decision tree from data. All pairs of students used the data cards in some way. A major difference in using the data cards was observed between the pairs who sorted the data cards in ascending order at the beginning and the pairs that did not sort but grouped and regrouped the data cards multiple times during the process. For both approaches, we saw the underlying strategy to choose an initial threshold for a data split, run through one cycle of the process (see Figure 5), and then run at least one more cycle with a new threshold. However, the "sorters" seemed to be more successful because all of them found the optimal solution, and none of them needed a prompt to find their solution. A flaw of not sorting the data cards but grouping them multiple times was that there had been a greater chance of making mistakes due to incorrect grouping, as seen in the example of Romina and Klara. Sorting offered several advantages, elaborated upon below.

Another interesting observation was that one pair (Moritz and Henry) used the data cards to make a useful representation (Component 4) of their DT (see Figure 11), which was not intended by the teaching unit but could be considered for a redesign.

Choice of data splits to consider. We observed a difference between the "sorters" and the "groupers" in considering data splits. The sorters all began by considering interspaces between two neighboring data cards; concrete values for a threshold were chosen after they had decided in which interspace the threshold should lie. The groupers almost exclusively mentioned concrete threshold values when they intended to consider a data split. Our explanation for that difference is as follows. Interspaces only become apparent by sorting the data cards, allowing for the consideration of intervals for data splits before deciding on specific thresholds. Groupers need a concrete value as a threshold before they can visualize and investigate a data split. Additionally, when sorted, all interspaces are visible simultaneously, whereas grouping leads to investigating only one data split at a time. These disadvantages might be an explanation for why the groupers seemed to have more difficulties coming up with a (good) threshold or realizing that a chosen threshold is not optimal. When trying to compare thresholds while looking at messily arranged data cards (not sorted, not grouped), there is some potential for confusion.

Another notable aspect is which concrete data splits the different pairs considered throughout the whole process. Even if we regard all pairs, there were only a few data splits that were explicitly considered. The interspaces of D and G were frequently considered, and for the sorters, these were the only two interspaces considered. The groupers also considered F, H, and I by considering values in the corresponding intervals. Six pairs chose D as their final decision, and one pair chose G.

In the following, we aim to hypothesize how the students might have chosen their data splits and decided when to terminate the process. We particularly focus on the strategy of the sorters, exemplified by Ben and Theo (see Figure 7). Explicitly, they only considered two out of ten possible data splits, namely, interspaces D and G. Their selection indicates what we call a "keen-eyed observation strategy" because it is reasonable to consider exactly these two spaces. We hypothesize that their selection was based on scanning the sorted cards by eye, thereby setting a threshold by visual impression. We assume that choosing the first threshold was done with the objective of obtaining one subset of cards with the highest possible proportion (high purity) of green cards and a subset with the highest possible proportion (high purity) of red cards, which can be assessed visually without counting cards. This visual assessment can, of course, be wrong, so a double check is done by considering other data splits and comparing them with a hard criterion such as the number of misclassifications.

We now explain why D and G are reasonable interspaces to consider. To choose an initial threshold, we start at the card with the lowest fat value (*Apple*, labeled green) and go up until the first occurrence of a different label (*Chicken nuggets*, labeled red). We select the interspace right before the label change between *Slice of grey bread* and *Chicken nuggets*. This leads to the interspace D with only green labels in the left subgroup (see Figure 8) and all red labels in the right subgroup (along with two green labels). It could be intuitively clear to the students that data splits to the left of this choice lead to more misclassifications because all cards on the left are labeled green. Then, other interspaces to the right of D are considered for comparison. Moving from D to the right, the next reasonable interspace is G because with E and F, only cards with red labels (*Chicken nuggets*, *Chocolate cereals*) are moved to the subgroup dominated by green labels, leading to more misclassifications. This kind of (purity-related) reasoning could be a rationale for starting with interspace D and then tentatively considering interspace G for an alternative data split. The interspaces of D and G are compared based on the number of misclassifications, leading to the choice of the initial interspace D. Then the procedure is terminated because the reasonable interspaces "near" D have been tested, and a locally optimal split is found. We speak of the locally optimal solution (which, of course, can be the overall optimal solution as well).

Ben and Theo did not explicitly communicate this strategy except for the comparison of D and G by misclassifications. However, it could be the explanation for their reasonable choice to consider D and G. We assume that considering D as the first threshold was not done as systematically by starting from the lowest interspace but rather by observing the sorted cards and by visual impression as described above as a "keen-eyed observation strategy." For now, we cannot tell for sure how the students proceeded. However, it is clear that with this data example, the students' strategies were effective.

Criteria for comparing different data splits. When reasoning about data splits, all seven student pairs referred to the number of misclassifications, with six pairs predominantly using this type of reasoning and thus aligning with the teaching unit's intended approach. Additionally, we found that five of seven pairs referred to the purity/impurity of the subgroups at some point, and for one pair, this was the dominant type of reasoning. The number of student pairs who reasoned about the purity of subgroups is remarkable because the authors did not anticipate this type of reasoning. Both types (misclassification or purity) are reasonable, but they have different advantages and disadvantages during the process, as elaborated in the following.

There are different phases in the process that we would expect in an optimal student solution. There is a first phase in which a starting threshold is chosen. In subsequent cycles, different thresholds are compared to make a final choice. In the first phase, reasoning with the purity/impurity of the subgroups by visual impression is beneficial because reasoning with the number of misclassifications would mean this number would have to be calculated for all data splits. Calculating the number for all data splits is the approach of an ML algorithm that we wanted to simplify here in favor of manual practicability. In the following cycles, in which a preselection of a few data splits is to be formally compared, the number of misclassifications is beneficial as a criterion because it is a formal criterion that can be calculated. Reasoning about the purity/impurity of subgroups is less suitable there, as seen in the reasoning of Romina and Klara, because it cannot be calculated formally but can just be evaluated according to visual impression. The pairs with the sorting strategy conducted the use of both criteria well. Some of them did not communicate any reasoning referring to purity/impurity, but we cautiously interpret that the successful choice of a good first threshold was based on a well-developed "keen-eyed observation

strategy." The pairs that did not sort also tried to come up with a good threshold by keen observation, but they had difficulties. For six of seven pairs, the "keen-eyed observation strategy," in combination with cycles of local improvement, yielded the same result that an algorithm would yield.

7. DISCUSSION AND OUTLOOK

The goal of the designed teaching unit was in line with the German Data Literacy Charta "to enable learners to actively shape data-related knowledge and decision-making" (Schüller et al., 2021, p. 3) using the example of data-based DTs. The study shows promising results concerning this goal.

As a first result, our study shows some evidence that our approach to teaching young students the core components of a DT algorithm enables them to construct and manually perform a heuristic for finding good data-based DTs using data cards. We see evidence that this elementarized data-based model-building process of DTs is teachable to middle school students. This adds to findings from Sulmont et al., who found for non-major students at the university level that "it is possible to teach ML to those with little to no math/CS background" (2019a, p. 953) and that barriers for students in the teaching process are not related to understanding the algorithms in ML (Sulmont et al., 2019b).

In the terminology of Chevallard and Bosch (2014), we found the learned knowledge to be largely consistent with the taught knowledge, and additionally, we found interesting aspects of learned knowledge that have the potential to motivate further investigation and inform the redesign of our teaching unit. Although the heuristics that were used by different pairs of students are similar to each other, the pairs that sorted the data cards were more successful than the pairs who used a grouping strategy. The teaching unit intended to favor the sorting strategy. However, in an early phase of the teaching unit, building data splits was introduced by grouping and regrouping the data cards in the form of embodied activity (Lakoff & Núñez, 2000) to foster an understanding of performing a data split. Later, sorting was offered as a strategy to quickly look at different data splits without having to regroup the data cards physically. Some pairs did not seem to take up this strategy from the latter phase and proceeded as in the early phase of the teaching unit. A redesign of the teaching unit could strengthen the role of sorting the data cards by making it not only optional but obligatory and addressing it in the phase of teaching and learning statistics with embodied activities and to a larger extent in reflections to clarify its usefulness as compared to the grouping strategy.

Another takeaway for the redesign of the teaching unit is to explicitly discuss the scanning of the sorted data cards ("keen-eyed observation strategy") with students after they have had some experiences with building DTs. Not all students may naturally develop this strategy on their own. It can further help to contrast between what a human does in such a situation and what a machine does in an ML algorithm. The "keen-eyed observation strategy," as described above, cannot be conducted by a machine because of its informal nature. An ML algorithm considers all data splits and compares them using a formal criterion.

Besides informing the redesign of our teaching unit, how research about teaching ML can profit from the insights of this study is described next. We see performing a heuristic for good data-based DTs using data cards as a first step towards understanding an ML algorithm. We consider "playing the machine" (Futschek & Moschitz, 2010) a valuable strategy for understanding the core components of ML algorithms in an approach based in constructivism. The students can understand what a machine does based on their own experiences. Only a few abstractions are necessary, which help to contrast what a machine does to what the students do as humans in such a heuristic. In the last part of our teaching unit, the transfer from the heuristic to an ML algorithm is done. Of course, this has to be embedded into a greater context of a teaching module that addresses additional aspects. For older students, it is possible and desirable to address the responsible use of ML and more complex ML phenomena (e.g., overfitting) with the support of digital tools as we do in other teaching modules (Biehler & Fleischer, 2021; Fleischer et al., 2022). We consider this study as a first step in using the concepts of data-based DTs to teach (young students) about ML.

Because we saw some evidence for the topic being teachable to middle school students, its usefulness can be further investigated. Kim et al. (2023) found different naive conceptions middle school students have of AI and ML (e.g., AI is seen as a cure-all solution). They state that "naive conceptions could be modified based on the learning interventions that were designed in age-appropriate hands-on practices" (Kim et al., 2023, p. 9848). The next step can be to investigate whether

our interventions are useful in this regard. Moreover, in future studies, it has to be investigated whether being able to understand and perform such a heuristic actually has an influence on the understanding of an ML algorithm, related ML phenomena, or general attitudes towards and conceptions of ML and AI Literacy (Long & Magerko, 2020).

Another follow-up study should deepen the insights about the "keen-eyed observation strategy" of scanning the data cards by eye. An eye-tracking study would be suitable to investigate the process in more detail and to examine if students behave as we have hypothesized for selecting the initial threshold.

8. LIMITATION

Participants in the exploratory study were volunteers, which may be a positive selection. The framework and results should be tested in further studies without a potential selection bias. We call for this and plan further publications ourselves.

REFERENCES

Bargagliotti, A., Franklin, C., Arnold, P., Gould, R., Johnson, S., Perez, L., & Spangler, D. A. (2020). *Pre-K–12 guidelines for assessment and instruction in statistics education II (GAISE II): A framework for statistics and data science education* (2nd ed.). American Statistical Association.

Biehler, R., & Fleischer, Y. (2021). Introducing students to machine learning with decision trees using CODAP and Jupyter notebooks. *Teaching Statistics*, *43*(S1), 133–142. https://doi.org/10.1111/test.12279

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. Taylor & Francis. https://doi.org/10.1201/9781315139470

Casal-Otero, L., Catala, A., Fernández-Morante, C., Taboada, M., Cebreiro, B., & Barro, S. (2023). AI literacy in K–12: A systematic literature review. *International Journal of STEM Education*, *10*(1), Article 29. https://doi.org/10.1186/s40594-023-00418-7

Chevallard, Y., & Bosch, M. (2014). Didactic transposition in mathematics education. In S. Lerman (Ed.), *Encyclopedia of mathematics education* (pp. 170–174). Springer Netherlands.

Engel, J. (2017). Statistical literacy for active citizenship: A call for data science education. *Statistics Education Research Journal*, *16*(1), 44–49. https://doi.org/10.52041/serj.v16i1.213

Engel, J., Erickson, T., & Martignon, L. (2018). Teaching and learning about tree-based methods for exploratory data analysis. In M. A. Sorto, A. White, & L. Guyot (Eds.), *Looking back, looking forward. Proceedings of the tenth international conference on teaching statistics (ICOTS10, July 2018), Kyoto, Japan*. International Statistical Institute.

Ericsson, K. A., & Simon, H. A. (1980). Verbal reports as data. *Psychological Review*, *87*(3), 215–251. https://doi.org/10.1037/0033-295X.87.3.215

Fleischer, Y., Biehler, R., & Schulte, C. (2022). Teaching and learning data-driven machine learning with educationally designed Jupyter notebooks. *Statistics Education Research Journal*, *21*(2), Article 7. https://doi.org/10.52041/serj.v21i2.61

Futschek, G., & Moschitz, J. (2010). Developing algorithmic thinking by inventing and playing algorithms. In J. E. Clayson & I. Kalaš (Eds.), *Constructionist approaches to creative learning, thinking and education: Lessons for the 21st century* (Proceedings of Constructionism 2010, pp. 1–10). Comenius University. https://publik.tuwien.ac.at/files/PubDat_187461.pdf

Garfield, J. B., & Ben-Zvi, D. (2008). *Developing students' statistical reasoning: Connecting research and teaching practice*. Springer.

Hancock, S. A., & Rummerfield, W. (2020). Simulation methods for teaching sampling distributions: Should hands-on activities precede the computer? *Journal of Statistics Education*, *28*(1), 9–17. https://doi.org/10.1080/10691898.2020.1720551

Harradine, A., & Konold, C. (2006). How representational medium affects the data displays students make. In A. Rossman & B. Chance (Eds.), *Working co-operatively in statistics education. Proceedings of the seventh international conference on teaching statistics* (ICOTS7), Salvador, Bahia, Brazil. International Statistical Institute.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Springer Verlag. https://doi.org/10.1007/978-0-387-84858-7

Hitron, T., Orlev, Y., Wald, I., Shamir, A., Erel, H., & Zuckerman, O. (2019). Can children understand machine learning concepts? The effect of uncovering black boxes. In A. Cox & V. Kostakos (Eds.), *Proceedings of the 2019 CHI conference on human factors in computing systems* (Paper 415). Association for Computing Machinery. https://doi.org/10/ghnn97

International Data Science in Schools Project Curriculum Team. (2019). *Curriculum frameworks for introductory data science*. http://idssp.org/files/IDSSP_Frameworks_1.0.pdf

Kauermann, G., & Küchenhoff, H. (2011). *Stichproben: Methoden und praktische Umsetzung mit R* [Samples: Methods and practical implementation with R]. Springer . https://doi.org/10.1007/978-3-642-12318-4

Kim, K., Kwon, K., Ottenbreit-Leftwich, A., Bae, H., & Glazewski, K. (2023). Exploring middle school students' common naive conceptions of artificial intelligence concepts, and the evolution of these ideas. *Education and Information Technologies, 28*(8), 9827–9854. https://doi.org/10.1007/s10639-023-11600-3

Lakoff, G., & Núñez, R. E. (2000). *Where mathematics comes from: How the embodied mind brings mathematics into being.* Basic Books.

Long, D., & Magerko, B. S. (2020). What is AI literacy? Competencies and design considerations. In J. McGrenere, Andy Cockburn, I. Avellino, & A. Goguey (Eds.), *Proceedings of the 2020 CHI conference on human factors in computing systems* (Paper 598). Association for Computing Machinery. https://doi.org/10/ghbz2q

Martins, R. M., & Gresse von Wangenheim, C. (2022). Findings on teaching machine learning in high school: A ten-year systematic literature review. *Informatics in Education, 22*(3), Article 4. https://doi.org/10.15388/infedu.2023.18

Mayring, P. (2015). Qualitative content analysis: Theoretical background and procedures. In A. Bikner-Ahsbahs, C. Knipping, & N. Presmeg (Eds.), *Approaches to qualitative research in mathematics education* (pp. 365–380). Springer. https://doi.org/10.1007/978-94-017-9181-6_13

Mike, K., & Hazzan, O. (2022). Machine learning for non-majors: A white box approach. *Statistics Education Research Journal, 21*(2), Article 10. https://doi.org/10.52041/serj.v21i2.45

Ministerium für Schule und Bildung des Landes Nordrhein-Westfalen. (2021). *Kernlehrplan für die Sekundarstufe I*: *Klasse 5 und 6 in Nordrhein-Westfalen Informatik* [Core curriculum for secondary school I: Grades 5 and 6 in North Rhine-Westphalia computer science]. https://www.schulentwicklung.nrw.de/lehrplaene/lehrplan/256/si_kl5u6_if_klp_2021_07_01.pdf

Mobasher, B., Dettori, L., Raicu, D., Settimi, R., Sonboli, N., & Stettler, M. (2019). Data science summer academy for Chicago Public School students. *ACM SIGKDD Explorations Newsletter, 21*(1), 49–52. https://doi.org/10.1145/3331651.3331661

Podworny, S., Fleischer, Y., & Hüsing, S. (2022). Grade 6 students' perception and use of data-based decision trees. In S. A. Peters, L. Zapata-Cardona, F. Bonafini, & A. Fan (Eds.), *Bridging the gap: Empowering and educating today's learners in statistics. Proceedings of the 11th international conference on teaching statistics (ICOTS11, 2022), Rosario, Argentina*. International Association for Statistical Education. https://doi.org/10.52041/iase.icots11.T2H3

Podworny, S., Fleischer, Y., Hüsing, S., Biehler, R., Frischemeier, D., Höper, L., & Schulte, C. (2021). Using data cards for teaching data based decision trees in middle school. In O. Seppälä & A. Peterson (Eds.), *Koli calling '21: 21st Koli calling international conference on computing education research* (pp. 1–3). Association for Computing Machinery. https://doi.org/10.1145/3488042.3489966

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers.

Ridgway, J. (2016). Implications of the data revolution for statistics education: The data revolution and statistics education. *International Statistical Review, 84*(3), 528–549. https://doi.org/10/f3q6f6

Santana, O. A., Sousa, B. A. d., Monte, S. R. S. d., Lima, M. L. d. F., & Silva, C. F. e. (2020). Deep learning practice for high school student engagement in STEM careers. In A. Cardoso, G. R. Alves, & T. Restivo (Eds.), *Proceedings of the 2020 IEEE global engineering education conference (EDUCON)* (pp. 164–169). Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/EDUCON45650.2020.9125281

Schönbrodt, S., Camminady, T., & Frank, M. (2022). Mathematische Grundlagen der Künstlichen Intelligenz im Schulunterricht: Chancen für eine Bereicherung des Unterrichts in linearer Algebra [Mathematical foundations of artificial intelligence in school lessons: Opportunities for enrichment of teaching linear algebra]. *Mathematische Semesterberichte, 69*. https://doi.org/10.1007/s00591-021-00310-x

Schüller, K., Koch, H., & Rampelt, F. (2021). *Data literacy charter*. Stifterverband. https://www.stifterverband.org/sites/default/files/data-literacy-charter.pdf

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms.* Cambridge University Press. https://doi.org/10.1017/CBO9781107298019

Sulmont, E., Patitsas, E., & Cooperstock, J. R. (2019a). Can you teach me to machine learn? In E. K. Hawthorne, M. A. Pérez-Quiñones, S. Heckman, & J. Zhang (Eds.), *SIGCSE: Proceedings of the 50th ACM technical symposium on computer science education* (pp. 948–954). Association for Computing Machinery. https://doi.org/10.1145/3287324.3287392

Sulmont, E., Patitsas, E., & Cooperstock, J. R. (2019b). What is hard about teaching machine learning to non-majors? Insights from classifying instructors' learning goals. *ACM Transactions on Computing Education, 19*(4), Article 33. https://doi.org/10.1145/3336124

Vartiainen, H., Tedre, M., & Valtonen, T. (2020). Learning machine learning with very young children: Who is teaching whom? *International Journal of Child-Computer Interaction, 25*, Article 100182. https://doi.org/10/gjvbc9

Zieffler, A., Justice, N., delMas, R., & Huberty, M. D. (2021). The use of algorithmic models to develop secondary teachers' understanding of the statistical modeling process. *Journal of Statistics and Data Science Education, 29*(1), 131–147. https://doi.org/10.1080/26939169.2021.1900759