MACHINE LEARNING FOR NON-MAJOR DATA SCIENCE STUDENTS: A WHITE BOX APPROACH

KOBY MIKE
Technion – Israel Institute of Technology
mike@campus.technion.ac.il

ORIT HAZZAN
Technion – Israel Institute of Technology
oritha@technion.ac.il

ABSTRACT

Data science is a new field of research that has attracted growing interest in recent years as it focuses on turning raw data into understanding, insight, knowledge, and value. New data science education programs, which are being launched at an increasing rate, are designed for multiple education levels and populations. Machine learning (ML) is an essential element of data science that requires an extensive background in mathematics. Whereas it is possible to teach the principles of ML only as a black box, novice learners might find it difficult to improve an algorithm's performance without a white box understanding of the underlying ML algorithms. In this paper, we suggest a pedagogical method, based on hands-on pen-and-paper tasks, to support white box understanding of ML algorithms for learners who lack the level of mathematics knowledge required for this purpose. Data were collected using a comprehension questionnaire and analyzed according to the process-object theory borrowed from mathematics education research. We present evidence of the effectiveness of this method based on data collected in an introduction-level data science course for graduate psychology students. This population had extensive psychology domain knowledge, as well as an established background in statistics, but had gaps in mathematical and computer science knowledge compared with data science majors. The research contribution is both practical and theoretical. Practically, we present a learning module that supports non-major data science students' white box understanding of ML. Theoretically, we propose a data analysis method to evaluate students' conceptions of ML algorithms.

Keywords: *Statistics education research; Data science education; Machine learning; Process-object duality theory*

1. INTRODUCTION

Data science is a new field of research that focuses on turning raw data into understanding, insight, knowledge, and value (Skiena, 2017; Wickham & Grolemund, 2016). It is an interdisciplinary field that integrates knowledge and methods from computer science, mathematics, and statistics, as well as from the domain knowledge of the data. More than a decade ago, Conway (2010) suggested representing the field of data science as a Venn diagram. While no consensus has been reached regarding the representation of data science as a Venn diagram, and regarding Conway's original diagram, we find it useful to represent the interdisciplinary essence of data science, including mathematics, statistics, computer science, and domain knowledge, using such a diagram (see authors' version in Figure 1).

The radical growth in recent years in the availability of both data and the computational resources required to process them has led to a corresponding increase in the demand for data scientists. As a result, new data science education programs are being launched at a growing rate, many of which are offered to undergraduate students (Berman et al., 2018). The interest in data science extends beyond undergraduate data science students, and so data science programs are being established in the context of computer science, statistics, and a variety of other domains, such as social science. Data science programs are also being developed for multiple education levels, from primary school children, through high school pupils and undergraduate and graduate students, to industry professionals and academic researchers.

Statistics Education Research Journal, 21(2), Article 10. https://doi.org/10.52041/serj.v21i2.45

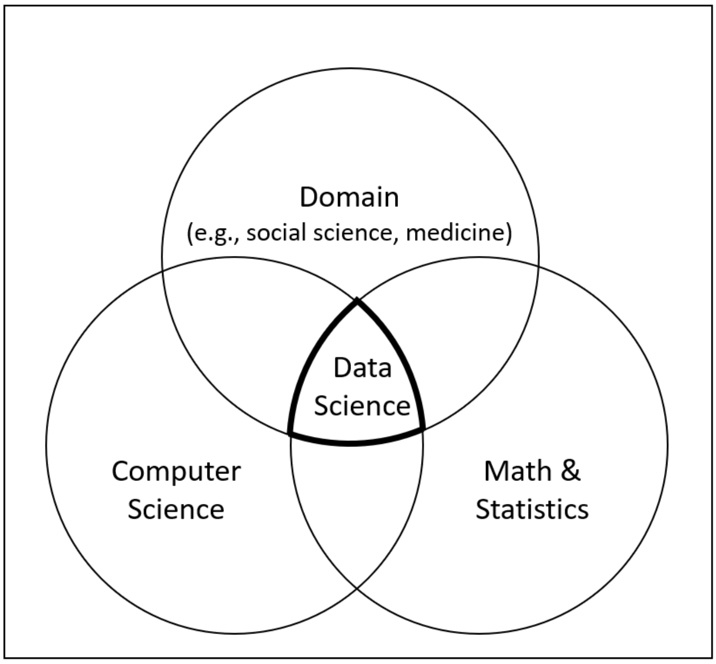


Figure 1. Data science is an interdisciplinary field that integrates knowledge and methods from computer science, mathematics, and statistics, as well as from the domain knowledge of the data

This paper focuses on teaching machine learning (ML) to graduate psychology students. ML modeling is a significant phase of the data science lifecycle, which represents the process of transforming raw data into understanding, insight, knowledge, and value (Berman et al., 2018). Although several models of the data lifecycle exist, some phases are common to all variants and these include data acquisition, data cleaning and tidying, exploratory data analysis and visualization, modeling, and communications. ML is one of the effective methods for modeling huge and complex data and is considered to be a field of research located in the intersection of statistics and computer. Unlike traditional parametric modeling, which presumes an underlying statistical behavior of the researched phenomena and aims to fit the best parameters of this model, ML models learn from experience (Goodfellow et al., 2016) and can learn complex data patterns directly from raw data. At the same time, however, not dealing with random variation and uncertainty is often considered a shortcoming of ML.

Data science programs require extensive knowledge and skills in mathematics, statistics, and computer science (Anderson et al., 2014; Danyluk et al., 2019; De Veaux et al., 2017; Demchenko et al., 2016). In particular, ML requires an understanding of the algorithms themselves, as well as a broader view of the algorithms in the context of the domain. Such understanding requires knowledge about the role of ML as a component of the data science process, handling of biases in the data, the role of training and test data, the evaluation of ML methods and models in the context of the application, ethics, and social responsibility.

ML algorithms may therefore be a complex topic to learn in general, and particularly difficult for those *not* majoring in statistics, computer science, or data science. Sulmont et al. (2019a, 2019b) described non-major learners of ML as learners who lack sufficient knowledge in mathematics, statistics, and programming. For example, according to the results of their interviews with instructors of ML courses offered to such learners, Sulmont et al. (2019a) mapped the difficulties such learners encounter while learning ML according to the Structure of the Observed Learning Outcome (SOLO) taxonomy. The SOLO taxonomy, which classifies learning outcomes in terms of their complexity, consists of five levels: pre-structural, unistructural, multi-structural, relational, and extended abstract (Biggs & Collis, 2014). Sulmont et al. (2019a) found that, a) at the unistructural stage of the SOLO taxonomy, students' preconception of human thinking vs. computer processing was a barrier; b) at the relational stage, understanding decision making in ML was a barrier; and c) in the extended abstract stage, students' difficulty perceiving the limits of ML application correctly was a barrier as well. In addition, both mathematics and programming were found to be barriers, and so mathematics was, in general, omitted from ML courses and programming was not included in all cases. Sulmont and her colleagues concluded that "Realizing that higher SOLO learning goals are more difficult to teach is useful for informing course design, public outreach, and the design of educational tools for teaching ML" (p. 1). Accordingly, in this paper, we present an educational tool that has the potential to support non-major data science students learning ML by mitigating the mathematical barrier.

The term "black box" understanding refers to understanding the relations between the input and the output of an algorithm without understanding how the algorithm itself works. Taking the black box approach, it is possible to learn the principles of ML even without sufficient mathematical and computational knowledge. For example, one can understand *how to use* a logistic regression as a classifier without understanding *how the algorithm works*, i.e., without understanding the process required to find the model parameters. Biehler and Schulte (2018), however, asked "what if machine learning is used in a data science course: Would it be appropriate to treat it as a black box …? Probably not" (p. 9). One reason for this assertion was that ML algorithms require multiple human decisions and tuning to achieve high performance, for which white box understanding is needed. For example, one challenging task when designing an ML algorithm is hyperparameter tuning (Sulmont et al., 2019b). ML hyperparameters are parameters of algorithms that are not learned from the data, but rather are set by the human developer of the application. Such parameters are, for example, the number of layers in a neural network or K in the KNN algorithm. The hyperparameters are used to control the learning process of the algorithm. Hyperparameter tuning is essential for optimizing the performance of the learning algorithm, and at the same time, requires an understanding of the mathematical details of the ML algorithm. Thus, whereas it is possible to teach the principles of ML without teaching the mathematical and computational knowledge required to fully understand them, individuals lacking this knowledge may find it difficult to optimize the performance of ML algorithms.

The term "white box" understanding, however, refers to understanding the details of the algorithm; that is, *how* it works. Using this terminology, an individual must have a white box understanding of the algorithm to understand its parameters and hyperparameter tuning and, as a result, is able to improve the algorithm's performance. We note that the application of ML has two distinct phases each of which can be understood as either black box or white box: model training and model usage. However, since each phase may require different mathematical knowledge, each of them can be understood, independently, as either a black box or a white box. Our goal is to support a white box understanding of both phases.

Although the importance of a white box understanding of ML algorithms is apparent, the literature survey undertaken indicated that pedagogical methods for teaching white box understanding of ML algorithms to learners who lack a mathematical and computer science background, have yet to be proposed. In our research, we attempted to close this gap, and in this paper, we present one possible pedagogical method to achieve this goal. Specifically, a hands-on pen-and-paper activity was developed based on theories borrowed from statistics and mathematics education research and students' answers were analyzed using these theories. Thus, the research has both practical and theoretical contributions. The practical contribution is expressed in the presentation of a learning module that supports non-major data science students' white box understanding of ML. The theoretical contribution is expressed by the introduction of a data collection and analysis method to evaluate students' conception of ML algorithms.

2. DATA SCIENCE AND STATISTICAL EDUCATION

New data science education programs are being launched at a growing rate (Raj et al., 2019). Most of these programs are designed for undergraduate students, and vigorous discussions have been taking place regarding the appropriate curricula of such programs (e.g., Danyluk et al., 2019; De Veaux et al., 2017). Undergraduate data science programs usually extend over three to four years, and they include learning a large body of knowledge from the fields of computer science, mathematics, and statistics. Some programs are designed as interdisciplinary programs and include the domain knowledge of one or more disciplines (Anderson et al., 2014; Havill, 2019; Khuri et al., 2017; Tartaro & Chosed, 2015).

The various target audiences for which data science courses are designed, do not always have extensive backgrounds, and so different programs have been tailored for primary school children (Srikant & Aggarwal, 2017), middle school pupils (Bryant et al., 2019; Dryer et al., 2018), high school pupils (Fisher et al., 2019; Gould et al., 2018; Haqqi et al., 2018; Heinemann et al., 2018), and liberal arts students (Havill, 2019). A recent review of data science programs suggests that a more balanced approach is needed with respect to the computation and statistics components of data science (Adams, 2020). Specifically, while the need for data science education for non-major data science students is evident (Dichev & Dicheva, 2017), there has been little discussion of the implementation of such a program (Cassel et al., 2016). In the following, we focus on the literature on statistics education for undergraduate and graduate social science students that is relevant for the readership of this journal.

The history of statistics education for undergraduate and graduate social science students includes a long list of pedagogical reforms and improvements (Carter et al., 2017; Crooks et al., 2019; Fillebrown, 1994; Hazan et al., 2018; Immekus, 2019; Kolaczyk et al., 2021; Prodromou & Dunne, 2017). Since many social science students experience some degree of anxiety about learning statistics, frequently as a result of a bad experience of learning mathematics at school, several methods have been proposed to better engage students with statistics. These methods are based on using real-life data that are relevant to the students (Fillebrown, 1994; Neumann et al., 2013; Prodromou & Dunne, 2017; Wiberg, 2009), problem-based learning (Buckley et al., 2015), project-based learning (Fillebrown, 1994), flipped classrooms (Immekus, 2019), real-life practicum (Kolaczyk et al., 2021), and hands-on activities (Hancock & Rummerfield, 2020; Pfaff & Weinberg, 2009).

Among these approaches, we elaborate on hands-on activities, which were also applied in our research. Pfaff and Weinberg (2009) examined the effectivity of hands-on activities in an introductory college statistics course. Although these activities did not significantly affect the students' grades, the students reacted positively to the hands-on modules and many students listed them as the most interesting element of the course. Students' responses to the question: "How was the hands-on module beneficial for your learning?" showed a median score of 4 on a scale of 1 to 5. In another study, Hancock and Rummerfield (2020) examined the effect of hands-on simulations on the understanding of sampling distribution and found a significant positive effect of the hands-on activities on the students' final exam grades.

In the context of understanding statistics, researchers describe several types of learning outcomes that differentiate between the understanding of *how* some statistical tool is calculated and the understanding of *when and how* that statistical tool is *used* correctly (Moore, 1997). Crooks et al. (2019) defined *conceptual knowledge* of statistics as "an understanding of the why of statistics in addition to the how" (p. 46).

In the context of ML, however, understanding the algorithm and how to use it is not sufficient. One of the main concerns when using ML algorithms is explainability: Can we understand what the algorithm learns? (Elad, 2017) and, can we understand why the algorithm suggests a specific prediction for a specific input? (Páez, 2019) These questions are the focus of the new field of explainable artificial intelligence (XAI), which is crucial to acquire human trust in ML algorithms.

3. THE OBJECT-PROCESS DUALITY THEORY AND ML

In this section, we first present the object-process duality theory borrowed from mathematical education research for our data analysis, and then present its applicability for ML education, in general, and for the understanding of ML algorithms as white box, in particular.

3.1. THE OBJECT-PROCESS DUALITY THEORY

The different ML algorithms taught in the program require students to understand the mathematical concepts of vectors, vector distances, dot products, derivatives and partial derivatives, and function optimization. None of these mathematical concepts is part of the psychology graduate curriculum (Rabin et al., 2018). The teaching of these algorithms, therefore, requires that the gaps in the students' mathematical knowledge be filled as part of the data science course or, alternatively, that some of the mathematical details be omitted and the gap filled with alternative intuitive explanations. As our pedagogical goal is to foster students' learning of ML algorithms using a white box approach, our objective was to fill the mathematical knowledge gap and so we looked for pedagogical theories and methods that were applicable from mathematics education research. Among these theories, we found the process-object duality to be suitable.

According to the process-object duality theory, abstract mathematical concepts can be represented in the human mind as either objects or processes (Sfard, 1991). As an object, an abstract mathematical concept is conceived of as a fixed construct, and as a process, an abstract mathematical concept is conceived of as an algorithm or a computation that generates an output from an input. For example, as an object, the concept of a function can be conceived of as a set of ordered pairs $\{(x_i, y_i)\}$, whereas as a process, a function can be represented in the human mind as the steps required to calculate the function output value y_i for a given input value x_i. In the learning processes of most mathematical concepts, the learner passes through three phases. First, the concept is conceived of as a process. Then, the process is mentally repacked (encapsulated), and an object representation is created in the learner's mind. In the final step, after the concept has been repacked and has become an object, it can be used by the learner as an element of a more complex process. Following the theory of process-object mathematical comprehension, it can be argued that understanding the algorithm as a process is an essential step toward understanding it as a procept and as an object. Conceiving of abstract mathematical concepts as objects, therefore, reflects a deeper understanding than conceiving of them as processes.

The concept of *procept*, introduced by Gray and Tall (1994), represents the mathematical duality that exists between the understanding of a process and a concept. It reflects the idea that an advanced thinker can hold both mental structures in his or her mind and can move back and forth between the two. Hence, a procept is a hybrid schema, an amalgam of the two representations, as a process and as an object.

In practice, we should ask: *How do learners build their mental representations of mathematical objects, and how do they advance their representations along the process-object continuum?* A current mathematical education theory suggests that the learning of mathematical concepts is accomplished by participating in mathematical routines, which can be viewed as repeated executions of mathematical tasks (Heyd-Metzuyanim & Graven, 2019; Sfard & Lavie, 2005). There are two types of mathematical routines: rituals and exploration (Lavie et al., 2019). A ritual is the execution of a mathematical routine on a lower level of thinking, i.e., a simple repetition of the mathematical procedure, based on a representation of the mathematical concept as a process. Exploration is the execution of a mathematical routine on a higher level of thinking and consists of constructing the mathematical concept as an object. Learners are required to practice rituals to be able to proceed toward an exploration process. Lavie et al. (2019) claimed that "in initial encounters with a new discourse, the learners can only participate in this discourse in ritualized ways. In further learning, their routines are expected to undergo gradual de-ritualization until they eventually turn into full-fledged explorations" (p. 1).

3.2. WHITE BOX UNDERSTANDING OF MACHINE LEARNING

The goal of our study was to design a pedagogical approach for white box understanding of ML algorithms. Not only are ML algorithms based on advanced mathematical concepts, but they themselves can be described as the "mathematical function mapping [of] some set of input values to output values" (Goodfellow et al., 2016, p. 5). It is therefore relevant to examine learners' conception of ML algorithms in term of understanding mathematical concepts as processes and objects.

Understanding an ML algorithm as a black box refers to the ability to call a library procedure that executes the algorithm without understanding the internal process of how the algorithm's output is generated. Such understanding, however, is not sufficient for hyperparameter tuning, for example, since the students' unfamiliarity with how the algorithm works does not allow them to understand how to tune it. For example, one of the teachers interviewed by Sulmont et al. (2019a) suggested that "their students cannot understand how tuning works, because they lack the mathematical prerequisite to understand parameters. Therefore, they claim, they think [ML] is magic when you tune parameters and get different results." (p. 11).

Understanding an ML algorithm as a white box refers to knowing how it works. Since ML algorithms are mathematical objects, white box understanding can be further described as understanding ML algorithms as processes, as objects, or as procepts. According to Sfard (1991), understanding an algorithm as an object requires an initial understanding of it as a process. Thus, the more a learner practices the execution of the algorithm as a process, the more chances he or she has of gradually developing an understanding of it as an object. Accordingly, the following question guided our research: What pedagogical methods can enhance students' understanding of an ML algorithm as a process, as the first step toward its encapsulation into an object?

Specifically, we asked ourselves whether it would be effective to implement the algorithms as computer codes and track their execution flow. Tracing problems are a common type of question in computer science education (Hazzan et al., 2020). In most cases, however, such problems focus on software execution tracking rather than on mathematical calculations. The learning of mathematical concepts can be supported by programming exercises (Leron & Dubinsky, 1995), but they require an understanding of the mathematical concepts, at least as processes, as well as proficiency in programming. This does not apply, however, to students who are not majoring in data science and who are studying ML; as such learners lack the required mathematical background and programming proficiency, and therefore, implementing an ML algorithm programmatically might not support their learning of the underlying mathematical concepts.

We thus developed a different approach, building on mathematical education rather on computer science education. First, based on the process-object theory, we explained the ML algorithms using only mathematical objects the learners are familiar with. For example, in our explanations of the KNN algorithm (see Section 4), we used only the Euclidean distance, which requires only basic mathematical operations (addition, subtraction, squaring, and taking square roots). Second. Building on the theory of mathematical ritual and exploration, we provided the students with learning opportunities by executing hands-on manual calculations of the algorithms. Similar to first grade pupils learning how to calculate the difference between two numbers by practicing ritual calculation in mathematical exercises until the ritual evaluation turns into exploration from which learning is derived, our students learned, for example, to calculate the difference between two vectors, using a hand-on task, as illustrated in the next section. We designed pen-and-paper tasks that mimic the hybrid, consisting of the mathematical routine and the computer science tracing task, in which the students are required to perform the calculations using pen and paper rather than computer software. These exercises focused solely on the mathematical concepts and were phrased using mathematical equations and symbols rather than code or pseudo-code.

Hands-on calculation, as a pedagogical tool, has been experimentally investigated to determine whether it improves the understanding of various statistical concepts. Although in some experiments, no significant improvement was found (Pfaff & Weinberg, 2009), others showed significant improvement (Hancock & Rummerfield, 2020). In both cases, students' feedback on the activity indicated that it positively influenced their learning and motivation.

Our approach to hands-on activities designed to impart the mathematical background required for learning ML is illustrated in the next section using one of the ML algorithms learned in the course: the KNN algorithm.

4. HANDS-ON ACTIVITY FOR LEARNING THE MATHEMATICS UNDERLYING MACHINE LEARNING

To illustrate the idea of using mathematical rituals in the teaching of ML algorithms, we present one example of implementing our hands-on approach for teaching the K-nearest neighbors (KNN) algorithm. KNN is the first algorithm in the curriculum since it is a relatively simple and intuitive classifier that uses the classification algorithm presented in Figure 2 and explained below. As part of our comprehensive research on data science education, similar hands-on tasks were developed for the perceptron, gradient descent, linear regression, logistic regression, and neural network algorithms, but are not presented here due to space limitations.

To classify a new object, U, with features $u_1, u_2, \ldots, u_n$:
(a) Calculate the distance of U from each sample $X^{(i)}$ in the training dataset.
(b) Find the K nearest neighbors of U.
(c) Find the most common label among these K nearest neighbors.
(d) U is classified accordingly to this most common label.

Figure 2. The classification algorithm of the KNN algorithm

The distance between a new object and its neighbors is calculated by vector distances. Several distance measures may be selected; we teach the Euclidian distance measure, as it is more intuitive to explain using its geometrical meaning. We therefore begin with a classification problem of objects with two features, i.e., a two-dimensional case. In this case, the Euclidian distance can be calculated based on the Pythagoras theorem. The distance, $d^{(i)}$, between the unknown object, U, and the i^{th} sample ($X^{(i)}$) in the training set is given by

$$d^{(i)} = \sqrt{(u_1 - x_1^{(i)})^2 + (u_2 - x_2^{(i)})^2}$$

To generalize the algorithm to the *n*-dimensional case, we present the vector distance between the unknown object, U, and the i^{th} sample ($X^{(i)}$) as a generalization of the Pythagoras theorem, given by

$$d^{(i)} = \sqrt{(u_1 - x_1^{(i)})^2 + (u_2 - x_2^{(i)})^2 + \cdots + (u_n - x_n^{(i)})^2}$$

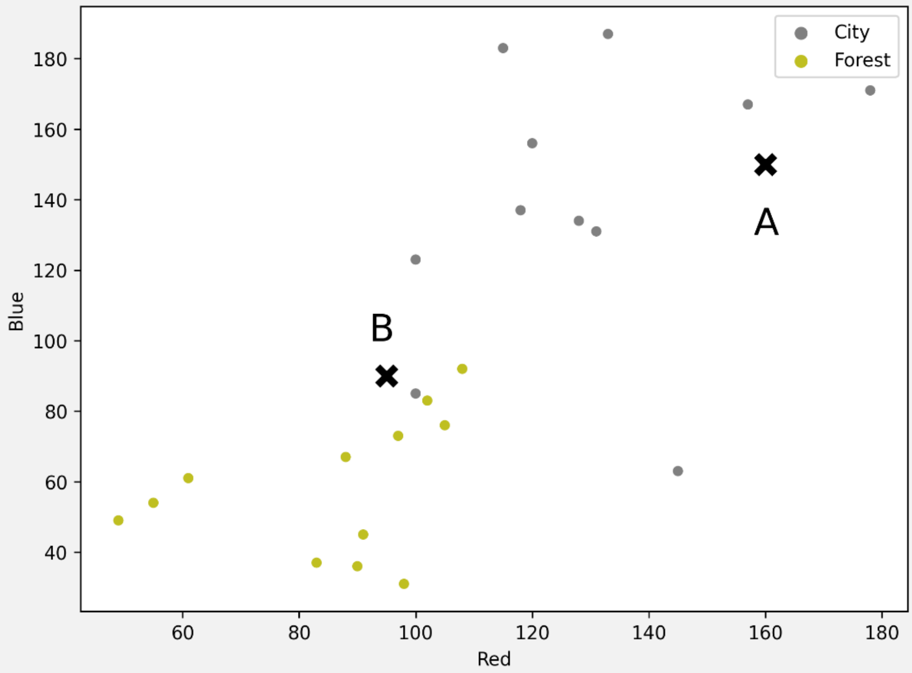
A common task for practicing the KNN algorithm is naturally a classification task of an unknown object based on a training set. One such exercise is the classification of Iris flowers based on four features of the flower: sepal length (SL), sepal width (SW), petal length (PL), and petal width (PW), using a well-known training dataset containing 150 examples of three types of Iris flowers (Fisher, 1936). Even with a small dataset of 150 flowers, this task might be too tedious for any learner to execute manually, and the calculation is, therefore, performed by a computer. Both methods, however, fail to produce the ritual required to support the learning of mathematical concepts, first as a process and then as an object, and so we designed a worksheet that guides learners to simulate the KNN algorithm manually and perform the required calculations themselves, manually (see Figure 3).

This worksheet guides a ritual execution of the mathematical process of classification using KNN and has two phases:

- The first phase requires visual identification of nearest neighbors in a two-dimensional case. It simulates the algorithm's search for the K nearest neighbors of a new instance and its classification according to the majority class. The given dataset was drawn in a way that enables to observe the classification easily without needing to actually calculate the distances.
- In the second phase, the students are asked to manually calculate the Pythagoras distances in the four-dimensional case for the Iris flower dataset, to find the nearest neighbors, and to classify a new instance according to the majority class. Since the given dataset is small (6 samples of flowers), the students can perform the calculations manually without using a computer program.

Part 1 – Image classification

A KNN algorithm is designed to classify images into two types: urban or forest; based on two features: Red - the mean level of the red color in the image, and Blue - the mean level of the blue color in the image.
The following graph presents the training dataset images (urban images in gray and forest images in green). The graph also shows two unknown images, A and B. Classify images A and B using a KNN algorithm, with K = 1 and K = 3.



1. For K = 1:
   a. The classification of image A is:
   Explain your answer: ____________________
   b. The classification of image B is:
   Explain your answer: ____________________
2. For K = 3:
   a. The classification of image A is:
   Explain your answer: ____________________
   b. The classification of image B is:
   Explain your answer: ____________________

Part 2 – Iris classification

A KNN algorithm is designed to classify Iris flowers into two classes: setosa and versicolor. Four features are given for each flower: sepal length (SL), sepal width (SW), petal length (PL), and petal width (PW). There are six samples of flowers in the training set.
A researcher found a new flower, U, with the following features: $U_{SL} = 5$, $U_{SW} = 3$, $U_{PL} = 2$, $U_{PW} = 2$.

1. Calculate the distance of the new flower from each sample in the training set:

Sample	sepal length (SL)	sepal width (SW)	petal length (PL)	petal length (PW)	Label	$d^{(i)}$
1	5.1	3.5	1.4	0.2	Setosa	
2	4.9	3	1.4	0.2	Setosa	
3	4.7	3.2	1.3	0.2	Setosa	
4	7	3.2	4.7	1.4	Versicolor	
5	6.4	3.2	4.5	1.5	Versicolor	
6	6.9	3.1	4.9	1.5	Versicolor	

2. Classify this new flower using the KNN algorithm with K = 1 and with K = 3.
For K = 1:
a. The indexes of the K closest training examples:
b. The labels of the K closest training examples:
c. The final classification is:
For K = 3:
a. The indexes of the K closest training examples:
b. The labels of the K closest training examples:
c. The final classification is:

Figure 3. The KNN algorithm worksheet

5. RESEARCH METHOD

This section describes the research method in detail. We start with the research goal and research question with respect to the current gap in the research literature regarding teaching ML to non-major data science students (subsection 5.1). Then, the research design and research population are presented (subsections 5.2 and 5.3, respectively). In subsection 5.4, we describe the KNN comprehension questionnaire, a tool we developed to collect data regarding students' understanding of the KNN algorithm and explain the theoretical considerations behind the development process. We conclude this section by laying out the method of data analysis and offering examples of it (subsection 5.5).

5.1. RESEARCH GOAL AND RESEARCH QUESTION

On the one hand, the literature on ML education points to the need for a white box understanding of ML algorithms (Biehler & Schulte, 2018), also indicating that a) non-major data science students have difficulty achieving such an understanding and b) that educational tools have not yet been developed to achieve this desired level of understanding (Sulmont et al., 2019b). On the other hand, the mathematical education research community, which is a significantly more mature research community than the ML education research community, offers theories that explain the processes of mathematics learning and understanding of mathematical concepts along with pedagogical approaches for achieving a high-level understanding of mathematical concepts (Lavie et al., 2019).

Addressing this gap, our research goal was to examine whether theories and practices borrowed from mathematics education research can support non-major students' white box understanding of ML algorithms. From this research goal we derived the following research question: Can teaching methods, derived from the mathematical education theories of object-process conception and ritual exploration, support non-major students' white box understanding of ML algorithms? Specifically, our research focused on the hands-on task presented in Section 4.

5.2. RESEARCH DESIGN

To answer this question, we designed a learning module on the KNN algorithm that included a recorded lecture and the hands-on task. The lecture and the task are based on the following principles, borrowed from mathematical education:

(a) In the recorded lecture, the students learn all mathematical details of the KNN algorithm, using only mathematical objects the students are familiar with, that is, finding the minimum of a list and calculating Euclidian distance using addition, subtraction, squaring, and taking square roots (the recorded lecture in Hebrew can be found here).
(b) In the hands-on activity, the students manually simulate the operation of the KNN algorithm and calculate the required operations. This task allows the students to execute the algorithm ritually as a process (see Figure 3, part 2).

To measure the learning outcomes, we designed a KNN comprehension questionnaire (see Section 5.4). This questionnaire was administered to the students after watching the recorded lecture about the KNN algorithm and completing the KNN hands-on activity. Thus, this questionnaire measured the students' accumulated learning of the KNN algorithm from both the lecture and the hands-on task.

The learning module was integrated into the Introduction to Data Science for Psychological Science course in the 2021 Spring semester (see Appendix 1). After the students watched the recorded lecture and solved the hands-on activity, they answered the KNN comprehension questionnaire (see Figure 4 below).

5.3. RESEARCH POPULATION

The research population included 23 students who were enrolled in the course "Data Science for Psychology Science" in a large research university in Israel (see Appendix 1). Gender was balanced (12 man and 11 women). Approximately 75% of the students were graduate (MA) students and 25% were PhD students. Half of the students were studying and researching social psychology and the other half were studying and researching cognitive psychology. Two students were studying neural sciences.

An introductory questionnaire was administered to the students before the onset of the semester to collect data on the participants' motivation to study the course and on their previous knowledge. Seventeen students answered this questionnaire. Student motivation was measured by a closed question based on our previous study on the learning of ML by social science and digital humanities graduate students and researchers (Mike et al., 2021). Ninety-four percent of the students indicated that they were interested in learning new research tools and 77% said they were interested in learning Python.

With respect to students' previous knowledge, as expected, the students evaluated their background in statistics as being higher than their knowledge in computer science and ML, and their initial knowledge regarding ML as being lower relative to all other topics mentioned in the questionnaire (Table 1).

Table 1. Students' self-evaluation of prior knowledge (1 = low, 5 = high) (n = 17)

Topic	Mean (Variance)
Programing in any language	3.9 (0.6)
Python programing	2.9 (0.9)
Descriptive statistics	4.2 (0.9)
Statistical inference	4.2 (1.0)
Linear regression	4.1 (1.4)
Machine learning	2.3 (1.0)

5.4. RESEARCH TOOLS

Theories in mathematics education research distinguish between a process and object conception of mathematical concepts (Sfard, 1991). A *process* conception of a mathematical concept is reflected by the ability to follow the calculation of the output of the concept for a specific input. An *object* conception of a mathematical concept, on the other hand, is reflected by the ability to examine it according to its properties, without necessarily checking to see how it works. Similarly, a *ritual* solution of a mathematical task is characterized by meticulously following the entire process of the solution, step by step, while an *explorative* solution of a mathematical task is characterized by the flexible and goal-oriented execution of previously learned rituals that have been conceptualized as objects and can, therefore, be manipulated and examined according to their properties (Lavie et al., 2019).

Accordingly, we characterized students' understanding of the KNN algorithm as a process and as an object. Students' conception of the KNN algorithm as a *process* in problem-solving processes, whose topic was the KNN algorithm, is exhibited when students specifically address at least one of the three steps required for classification of an unseen sample (P1-P3 below, see Textbox 1) and/or the step of tuning the K hyperparameter in the performance optimization stage (P4 below). We note that classification of an unseen sample is executed several times during the ML life cycle (as part of the algorithm's validation, testing, and prediction stages):

(P1) Calculate distance from all samples
(P2) Pick the K nearest samples
(P3) Find the label of the majority
(P4) Tune the hyperparameter K to improve performance (to avoid underfitting and overfitting)

Students' conception of the KNN algorithm as an *object* in problem-solving processes, whose topic was the KNN algorithm, is exhibited when students examine the following seven properties of the KNN algorithm—the first three (O1-O3) address the classification, the fourth (O4) addresses the algorithm's performance and the last three (O5-O7) address the algorithm's complexity (determined by the number of distance calculations):

(O1) Classification depends on similarity
(O2) Classification is determined by distance
(O3) The classification of a specific unknown example depends on K
(O4) The performance of the KNN algorithm for a specific K depends on the distribution of data
(O5) The number of distance calculations depends on the number of training samples
(O6) The number of distance calculations depends on the number of features
(O7) The number of distance calculations does not depend on K

To examine students' conception of the KNN algorithm, we used the KNN comprehension questionnaire (see Figure 4). Each question enabled us to reveal the students' conception of the KNN algorithm as either a process or an object or both, by examining the entity(ies) the students used in their answers: the four process steps or the seven object properties of the KNN algorithm, respectively (see Table 2).

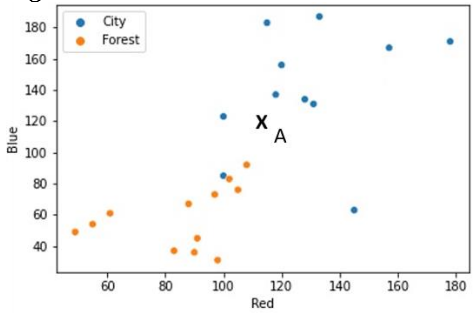
Question 1. The students are asked to explain the KNN algorithm to a friend. The aim of the request to explain the algorithm to a friend instead of simply defining it is to guide the students *not* to recall the KNN definition they learned in the lecture, but rather, to explain it in their own words. The formulation of this question also implies that the explanation should avoid jargon, but rather, should use only simple and understandable language.

Questions 2 and 3. The students are asked to indicate whether each of six statements on the prediction of the KNN algorithm, made by six hypothetical students (Alice, Bob, Carol, Dave, Eve, and Frank) is correct or incorrect, and to speculate how the hypothetical student explained his or her answer. These are critical thinking questions, which represent a higher order of thinking based on Bloom's taxonomy (Anderson et al., 2001; Bloom et al., 1956).

Question 4. The students are asked to examine the computational complexity of the KNN algorithm. While for 4(a) and 4(b), a process conception of the KNN algorithm is sufficient, an object conception is required in order to draw conclusions regarding its complexity in different situations (in 4(c), 4(d) and 4(e)). We note that graduate psychological students are not familiar with the concept of computational complexity and that it is not part of the course curriculum. Specifically in Question 4, the students are asked to indicate, for K = 5 and K = 11, the number of times the square operator had to be calculated in a specific classification problem using a KNN algorithm. Although the K values are different, the square operator must be calculated to find the Euclidian distance between the unknown instance and each of the training examples. In other words, 4,000 calculations are required in both cases, regardless of the value of K.

KNN Questionnaire

1. How would you explain to a friend what the KNN algorithm is?
2. Students were asked to classify Example A in the figure below, using the KNN algorithm for K = 5.



Alice claims that A's classification is Forest.

a. Is Alice, right?

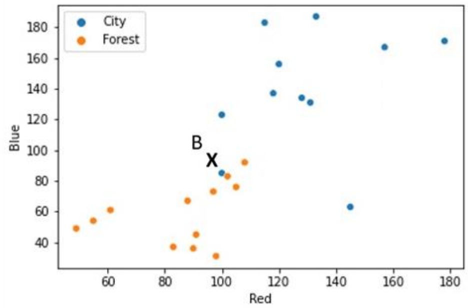
b. In your opinion, how did Alice explain her answer?

Bob claims that A's classification is City.

c. Is Bob, right?

d. In your opinion, how did Bob explain his answer?

3. Students were asked to classify Example B in the figure below, using the KNN algorithm.



Carol claims that for K=5, B's classification is City.
a. Is Carol, right?
b. In your opinion, how did Carol explain her answer?

Dave claims that for K=5, B's classification is Forest.
c. Is Dave, right?
d. In your opinion, how did Dave explain his answer?

Eve claims that for any K, B's classification is City.
e. Is Eve, right?
f. In your opinion, how did Eve explain her answer?

Frank claims that for any K, B's classification is Forest.
g. Is Frank right?
h. In your opinion, how did Frank explain his answer?

4. In order to classify dogs as Poodle or Labrador, four characteristics were selected: height, weight, tail length, and ear length. The training set included 1,000 dogs, 500 of each kind. Based on this data set, we wish to classify an unknown dog using the KNN classifier.
a. For K = 5: How many times is the square operation executed?
b. For K = 11: How many times is the square operation executed?
c. What conclusion can you draw from your answers to the above two questions?
d. In your opinion, when are the chances of a correct classification higher?
    I. K = 5
    II. K = 11
    III. It is impossible to decide
    IV. I do not know
e. Please explain your answer.

Figure 4: KNN comprehension questionnaire

Table 2. Mapping the questions of the comprehension questionnaire according to the KNN process steps and object properties each question elicits

Type of conception	Expression of conception	Question 1	2	3	4
Process conception	(P1) Calculate distance from all samples	X			X
	(P2) Pick the K nearest samples	X	X	X	
	(P3) Find the label of the majority	X	X	X	
	(P4) Tune the hyper parameter K to improve performance				X
Object conception	(O1) Classification depends on similarity	X			
	(O2) Classification is determined by distance	X			
	(O3) The classification of a specific unknown example depends on K	X		X	
	(O4) The performance of the KNN algorithm for a specific K depends on the distribution of the data				X
	(O5) The number of distance calculations depends on the number of training samples				X
	(O6) The number of distance calculations depends on number of features				X
	(O7) The number of distance calculations does not depend on K				X

5.5. DATA ANALYSIS METHODS

Students' answers to all questions in the comprehension questionnaire were analyzed to extract the students' conception of the KNN algorithm as either a process or an object or both. Specifically, for each student explanation provided in the comprehension questionnaire, we examined what process steps and/or object properties it includes and determined the student's conception accordingly. The coding scheme used is presented and illustrated in Appendix 2.

5.6. RESEARCH LIMITATIONS

This exploratory research aims to present preliminary results regarding the usability of the learning module (a lecture and hands-on task) to support non-major data science students' white box understanding of ML algorithms. The module's current design has several limitations. We mention three:

A) Student understanding was tested only once: Since the comprehension questionnaire (Figure 4) was given to the students after watching a recorded lecture about the KNN algorithm and solving the hands-on task (Figure 3), we cannot separate the effect of the lecture from that of the hands-on task.

B) We have no observation data on how the students solved the hands-on task and completed the comprehension questionnaire.

C) One data collection tool: Students' understanding was analyzed only by a written questionnaire. In future, to further validate the data analysis of this preliminary research, we intend to gather additional data about students' comprehension of the KNN algorithm and other algorithms using additional tools, such as interviews.

6. RESULTS AND DATA ANALYSIS

In this section, we present our analysis regarding students' conception of the KNN algorithm from the perspective of the process-object duality, first from the student point of view and then from the algorithmic perspective. Specifically, in Section 6.1, we categorize the students according to their conception of the KNN algorithm as either a process, object, or procept. Then, in Section 6.2, we delve into the details, analyzing the process steps of object properties the students addressed in their answers and what can be learned from the analysis. In the next section (Section 7) we further delve into the details of these conceptions and their implications for the pedagogy of machine learning.

6.1. STUDENTS' CONCEPTION FROM THE PROCESS-OBJECT DUALITY PERSPECTIVE

Reviewing the students' coded answers (see Table 3) revealed that:

i) About half of the students (5 out of 12, light green in Table 3) conceived the KNN algorithm as a process, using in their explanations at least half of its process steps but less than half of its object properties. Clearly, other criteria could be used to determine this conception, but for the sake of simplicity, we set this simple criterion.
ii) About half of the students (7 out of 12, dark green in Table 3) conceived the KNN algorithm as a procept, using in their explanations at least half of its process steps and at least half of its object properties. Again, we used this criterion for the sake of simplicity.
iii) None of the students used object properties exclusively in their explanations, and accordingly, none of the students' conception was classified as an object.

6.2. PROCESS STEPS AND OBJECT PROPERTIES CONCEPTION

We present three observations about the students' conception of each process step and object property, by their prevalence:

i) *Similarity*: The students best conceptualized the steps of *picking the K nearest samples* (P2), *finding the label of the majority* (P3), and the property *classification is determined by distance* (O2). This result may be explained by the facts that a) steps (P2) and (P3) are the most practiced steps in the hands-on task (Figure 3) and b) this understanding of (P2) and (P3) supports their understanding of (O2). This may explain the fact that all students gave correct answers to all of the closed items of Questions 2 and 3 (See Figure 4, Questions 2 a, c and 3 a, c, e, g). We note that only one third of the students mentioned that *classification depends on similarity* (O1). This can be explained by the fact that students used the terms *similarity* and *distance* interchangeably, due to the graphic representation in which similarity and distance are visually the same.
ii) *Complexity:* Eight students mentioned at least one of the properties: *The number of distance calculations depends on the number of training samples* (O5), *the number of distance calculations depends on the number of features* (O6), and *the number of distance calculations does not depend on K* (O7). Three students consistently mentioned all three properties (students 10, 11 and 12), and five students mentioned one or two of them. This is consistent with the fact that half of the students conceptualize the first process step *calculate the distance from all samples* (P1). Indeed, all students who mentioned this process step also mentioned two properties regarding complexity (number of calculations). This result might be due to the technique of solving visual tasks, i.e., Part 1 of the KNN hands-on activity (Figure 2) and Questions 2 and 3 on the KNN comprehension questionnaire (Figure 3). In these cases, the human mind can find the neighbors instantly, using only its visual processing capabilities, without needing to calculate the distances from all samples in the training set numerically.
iii) *K hyperparameter:* One third of the students mentioned the property that the *classification of a specific unknown example depends on K* (O3) and one half of the students mentioned the property that the *performance of the KNN algorithm for specific K depends on the distribution of the data* (O4). There is partial overlap (of 2 students) between the students who mentioned these properties regarding K and the students who mentioned the performance optimization phase in which the K hyperparameter is tuned (P4). This might indicate that more students conceptualized the role of K correctly but did not mention the tuning step, as the question formulation asked them merely to compare the possible performance of two different Ks, and not to describe how to maximize performance. This explanation is also supported by the fact that only one quarter of the students mentioned the step of tuning the hyperparameter K to optimize the algorithm performance (P4).

Table 3. Coded answers

			Student number											
Type of conception	Expression of conception	Total	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	9	10	11	12
Process conception	(P1) Calculate distance from all samples	6												
	(P2) Pick the K nearest samples	12												
	(P3) Find the label of the majority	12												
	(P4) Tune the hyperparameter K to improve performance	3												
Object conception	(O1) Classification depends on similarity	4												
	(O2) Classification is determined by distance	11												
	(O3) The classification of a specific unknown example depends on K	4												
	(O4) The performance of the KNN algorithm for specific K depends on the distribution of the data	6												
	(O5) The number of distance calculations depends on the number of training samples	6												
	(O6) The number of distance calculations depends on the number of features	6												
	(O7) The number of distance calculations does not depend on K	6												

7. DISCUSSION: PEDAGOGICAL IMPLICATIONS AND RESEARCH CONTRIBUTION

From a wider perspective, the process-object duality of the mental representation of mathematical concepts is associated with the phenomenon of reducing the abstraction level when learning abstract mathematical concepts (Hazzan, 1999). In general, students who need to learn mathematical concepts that are too abstract for their current mental representation, i.e., their conception, of the concepts, use several mechanisms to reduce the level of abstraction. One of these is based on the dual process-object representation of mathematical concepts, according to which students conceive abstract concepts that are too abstract as processes or procepts rather than as objects, which are considered a more abstract conception. This statement has been generalized to include the learning of computer science concepts as well (Hazzan, 2003a, 2003b; Hazzan & Hadar, 2005), and so the theory of process-object representation of mathematical concepts can be used also to describe students' understanding of ML algorithms on different levels of abstraction.

Within this context, the hands-on ritual-based tasks presented in this paper (see Figure 2) may support students' learning by reducing the level of abstraction (in intermediate learning stages) in several ways:

(a) Data have specific features that are more concrete (less abstract) than abstract features; images with features "red" and "green" are more concrete than objects with features "x_1" and "x_2".

(b) Data have specific values that are more concrete than abstract values; an image with features [107, 83, city] is more concrete than an object with features [x_1, x_2, x_3].
(c) Data are more concrete than just numbers, as they have meaning in the real world; "the level of red in the image is 107" is more concrete than the fact "the value of feature x_1 is 107".
(d) A small training dataset that learners can iterate manually is more concrete than the huge datasets typically used to demonstrate ML algorithms.
(e) The manual execution of a task is more concrete than running a simulation, such as a computer program, because attention must be paid to every detail, regardless of its role in the procedure.

Thus, the hands-on activities presented in this paper support learning not only by allowing the students to practice a ritual involving the underlying mathematical concepts, but also by intentionally reducing the level of abstraction of ML algorithms in intermediate learning stages.

Reducing the level of abstraction, however, should be done very carefully, since reducing the level of abstraction too much, may lead learners to conceive the specific as the general case and this, in turn, may lead to a process or a procept conception. For example, the graphical representation of the KNN, as described in the first part of the KNN hands-on task (see Figure 3) in which the students can find the neighbors using visual capabilities without needing to calculate the distance from *all* samples in the training set (since the algorithm does not really calculate the distance), might lead to partial conceptualization of the complexity of the KNN algorithm (see Section 6.2). Another example is the small number of training samples in the second part of the KNN hands-on task (see Figure 2), in which the students had to calculate the distances of a new example from too small a training data set (of only six samples), and so the first step of the KNN algorithm, calculating the distance from *all* the training samples, was not sufficiently clear.

From the abstraction perspective, the three categories of process steps and object properties presented in Section 6.2—*Similarity, Complexity, and K as hyperparameter*—can be organized by the level of abstraction they represent. *Similarity* may be explained visually and, thus, requires the lowest level of abstraction. *Complexity* may need to be explained using large tables and, thus, requires a higher level of abstraction, and *K as a hyperparameter* requires that the classification process be conceived as an object in order to optimize performance, therefore, requiring an even higher level of abstraction.

The above discussion highlights both the practical-pedagogical and theoretical contributions of this research. Practically, we present a learning module that supports non- major data science students' white box understanding of ML (at least as a process). Theoretically, we introduce a data analysis method to evaluate students' conceptions of ML algorithms. Combining these two facets of contribution may guide the teaching process of ML algorithms. For example, a teacher who notices that the majority of his or her class conceives specific ML algorithms as a process, should consider whether more advanced algorithms should be introduced to the class or alternatively, and most probably, additional practice of the specific algorithms is required. An illustrative example for this assertion is the perceptron and neural networks. A teacher should verify that the class conceptualizes the perceptron algorithm as an object before moving forward using it as a building block of a neural network.

8. CONCLUSION

The research presented in this paper addressed a major challenge of data science education: white box understanding of ML algorithms by non-major data science students who lack a sufficient mathematical background for this kind of understanding. Specifically, to support the learning process that leads to white box understanding of the mathematical concepts learned in an ML course, we presented a pedagogical method that is based on a theory according to which mathematical concepts can be comprehended as processes and as objects. We propose that teaching algorithms based on a) mathematical concepts that are already known to the learners and b) hands-on tasks, can improve learners' understanding of algorithms as processes also in the case of complex algorithms which students may not be able to understand as white boxes, and thus are commonly learned as black boxes. We illustrated our claims with the results of an exploratory research conducted on graduate psychology students who learned ML as part of an introduction level data science course. This method should be further investigated and modified to fit a variety of learner populations of data science to promote a deeper understanding of ML algorithms. Such an investigation is needed due to the crucial role that data play in present times and the importance attributed to the responsible and critical use of data in many real-life domains and situations. We also propose to further study the comprehension and conceptualization of ML algorithms from the process-object duality perspective. Such research may lead to the invention and adaption of other pedagogical methods to teach ML algorithms to non-major data science learners.

9. ACKNOWLEDGMENTS

This work was supported by generous funding from the Technion's Machine Learning & Intelligent Systems (MLIS) Center. We would also like to thank to Prof. Galit Yovel from the School of Psychological Sciences in Tel Aviv University and Prof. Eran Yahav from the Faculty of Computer Science in the Technion for their cooperation and support in this research. We would also like to acknowledge Prof. Rolf Biehler and the reviewers of this paper for their insightful comments, which helped improve this paper dramatically.

REFERENCES

Adams, J. C. (2020). Creating a balanced data science program. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (pp. 185–191). https://doi.org/10.1145/3328778.3366800

Anderson, L. W., Bloom, B. S., & others. (2001). *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. Longmans.

Anderson, P., Bowring, J., McCauley, R., Pothering, G., & Starr, C. (2014). An undergraduate degree in data science: Curriculum and a decade of implementation experience. *Proceedings of the 45th ACM Technical Symposium on Computer Science Education - SIGCSE '14* (pp. 145–150). https://doi.org/10.1145/2538862.2538936

Berman, F., Rutenbar, R., Hailpern, B., Christensen, H., Davidson, S., Estrin, D., Franklin, M., Martonosi, M., Raghavan, P., Stodden, V., & Szalay, A. S. (2018). Realizing the potential of data science. *Communications of the ACM*, *61*(4), 67–72. https://doi.org/10.1145/3188721

Biehler, R., & Schulte, C. (2018). *Paderborn symposium on data science education at school level 2017: The collected extended abstracts*. Universitätsbibliothek.

Biggs, J. B., & Collis, K. F. (2014). *Evaluating the quality of learning: The SOLO taxonomy (Structure of the Observed Learning Outcome)*. Academic Press.

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). *Taxonomy of educational objectives. Vol. 1: Cognitive domain.* Longmans.

Bryant, C., Chen, Y., Chen, Z., Gilmour, J., Gumidyala, S., Herce-Hagiwara, B., Koures, A., Lee, S., Msekela, J., Pham, A. T., & others. (2019). A middle-school camp emphasizing data science and computing for social good. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 358–364).

Buckley, J., Brown, M., Thomson, S., Olsen, W., & Carter, J. (2015). Embedding quantitative skills into the social science curriculum: Case studies from Manchester. *International Journal of Social Research Methodology*, *18*(5), 495–510.

Carter, J., Brown, M., & Simpson, K. (2017). From the classroom to the workplace: How social science students are doing data analysis for real. *Statistics Education Research Journal*, *16*(1), 80–101. https://doi.org/10.52041/serj.v16i1.218

Cassel, L. N., Dicheva, D., Dichev, C., Goelman, D., & Posner, M. (2016). Data science for all: An introductory course for non-majors; in flipped format (Abstract Only). *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (p. 691). https://doi.org/10.1145/2839509.2850558

Conway, D. (2010). The data science venn diagram. *http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram*

Crooks, N. M., Bartel, A. N., & Alibali, M. W. (2019). Conceptual knowledge of confidence intervals in psychology undergraduate and graduate students. *Statistics Education Research Journal*, *18*(1), 46–62. https://doi.org/10.52041/serj.v18i1.149

Danyluk, A., Leidig, P., Cassel, L., & Servin, C. (2019). ACM task force on data science education: Draft report and opportunity for feedback. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 496–497). https://doi.org/10.1145/3287324.3287522

De Veaux, R. D., Agarwal, M., Averett, M., Baumer, B. S., Bray, A., Bressoud, T. C., Bryant, L., Cheng, L. Z., Francis, A., Gould, R., Kim, A. Y., Kretchmar, M., Lu, Q., Moskol, A., Nolan, D., Pelayo, R., Raleigh, S., Sethi, R. J., Sondjaja, M., … Ye, P. (2017). Curriculum guidelines for undergraduate programs in data science. *Annual Review of Statistics and Its Application*, *4*(1), 15–30. https://doi.org/10.1146/annurev-statistics-060116-053930

Demchenko, Y., Belloum, A., Los, W., Wiktorski, T., Manieri, A., Brocks, H., Becker, J., Heutelbeck, D., Hemmje, M., & Brewer, S. (2016). EDISON data science framework: A foundation for building data science profession for research and industry. *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)* (pp. 620–626). https://doi.org/10.1109/CloudCom.2016.0107

Dichev, C., & Dicheva, D. (2017). Towards data science literacy. *Procedia Computer Science*, *108*, 2151–2160. https://doi.org/10.1016/j.procs.2017.05.240

Dryer, A., Walia, N., & Chattopadhyay, A. (2018). A middle-school Module for Introducing data-mining, big-data, ethics and privacy using RapidMiner and a Hollywood theme. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 753–758). https://doi.org/10.1145/3159450.3159553

Elad, M. (2017). Deep, deep trouble: Deep learning's impact on image processing, mathematics, and humanity. *SIAM News*, *50*(4). https://sinews.siam.org/Details-Page/deep-deep-trouble-4

Fillebrown, S. (1994). Using projects in an elementary statistics course for non-science majors. *Journal of Statistics Education*, *2*(2). https://doi.org/10.1080/10691898.1994.11910470

Fisher, N., Anand, A., Gould, R., Hesterberg, J. B. ans T., Bailey, J., Ng, R., Burr, W., Rosenberger, J., Fekete, A., Sheldon, N., Gibbs, A., & Wild, C. (2019, September). *Curriculum frameworks for introductory data science*. http://www.idssp.org/files/IDSSP_Data_Science_Curriculum_Frameworks_for_Schools_Edition_1.0.pdf

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*(2), 179–188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1). MIT Press.

Gould, R., Suyen, M.-M., James, M., Terri, J., & LeeAnn, T. (2018). Mobilize: A data science curriculum for 16-year-old students. In M. A. Sorto, White, & L. Guyot, L. (Eds.), *Looking back, looking forward. Proceedings of the 10th International Conference on the Teaching of Statistics* (ICOTS10), Kyoto, Japan, July 8–14. International Statistics Institute.

Gray, E. M., & Tall, D. O. (1994). Duality, ambiguity, and flexibility: A "proceptual" view of simple arithmetic. *Journal for Research in Mathematics Education*, *25*(2), 116–140. https://doi.org/10.5951/jresematheduc.25.2.0116

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

Haqqi, S., Sooriamurthi, R., Macdonald, B., Begandy, C., Cameron, J., Pirollo, B., Becker, E., Choffo, J., Davis, C., Farrell, M., & others. (2018). Data jam: Introducing high school students to data science. *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (pp. 387–387).

Havill, J. (2019). Embracing the liberal arts in an interdisciplinary data analytics program. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 9–14).

Hazan, B., Zhang, W., Olcum, E., Bergdoll, R., Grandoit, E., Mandelbaum, F., Wilson-Doenges, G., & Rabin, L. (2018). Gamification of an undergraduate psychology statistics lab: Benefits to perceived competence. *Statistics Education Research Journal*, *17*(2), 255–265. https://doi.org/10.52041/serj.v17i2.169

Hazzan, O. (1999). Reducing abstraction level when learning abstract algebra concepts. *Educational Studies in Mathematics*, *40*(1), 71–90.

Hazzan, O. (2003a). How students attempt to reduce abstraction in the learning of mathematics and in the learning of computer science. *Computer Science Education*, *13*(2), 95–122.

Hazzan, O. (2003b). Reducing abstraction when learning computability theory. *Journal of Computers in Mathematics and Science Teaching*, *22*(2), 95–117.
Hazzan, O., & Hadar, I. (2005). Reducing abstraction when learning graph theory. *Journal of Computers in Mathematics and Science Teaching*, *24*(3), 255–272.
Hazzan, O., Ragonis, N., & Lapidot, T. (2020). *Guide to teaching computer science: An activity-based approach*. Springer.
Heinemann, B., Opel, S., Budde, L., Schulte, C., Frischemeier, D., Biehler, R., Podworny, S., & Wassong, T. (2018). Drafting a data science curriculum for secondary schools. *Proceedings of the 18th Koli Calling International Conference on Computing Education Research - Koli Calling '18* (pp. 1–5). https://doi.org/10.1145/3279720.3279737
Heyd-Metzuyanim, E., & Graven, M. (2019). Rituals and explorations in mathematical teaching and learning: Introduction to the special issue. *Educational Studies in Mathematics*, *101*(2), 141–151. https://doi.org/10.1007/s10649-019-09890-x
Immekus, J. C. (2019). Flipping statistics courses in graduate education: Integration of cognitive psychology and technology. *Journal of Statistics Education*, *27*(2), 79–89.
Khuri, S., VanHoven, M., & Khuri, N. (2017). Increasing the Capacity of STEM Workforce: Minor in bioinformatics. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 315–320). https://doi.org/10.1145/3017680.3017721
Kolaczyk, E. D., Wright, H., & Yajima, M. (2021). Statistics practicum: Placing 'practice' at the center of data science education. *Harvard Data Science Review*. https://hdsr.mitpress.mit.edu/pub/twyc748y/release/4
Lavie, I., Steiner, A., & Sfard, A. (2019). Routines we live by: From ritual to exploration. *Educational Studies in Mathematics*, *101*(2), 153–176. https://doi.org/10.1007/s10649-018-9817-4
Leron, U., & Dubinsky, E. (1995). An abstract algebra story. *The American Mathematical Monthly*, *102*(3), 227–242. https://doi.org/10.1080/00029890.1995.11990563
Mike, K., Hartal, G., & Hazzan, O. (2021). Widening the shrinking pipeline: The case of data science. *2021 IEEE Global Engineering Education Conference (EDUCON)* (pp. 252–261).
Mike, K., & Hazzan, O. (2022). Interdisciplinary CS1 for non-majors: The case of graduate psychology students. *2022 IEEE Global Engineering Education Conference (EDUCON)* (pp. 86–93), https://doi.org/10.1109/EDUCON52537.2022.9766516
Moore, D. S. (1997). New pedagogy and new content: The case of statistics. *International Statistical Review*, *65*(2), 123–137.
Neumann, D. L., Hood, M., & Neumann, M. M. (2013). Using real-life data when teaching statistics: Student perceptions of this strategy in an introductory statistics course. *Statistics Education Research Journal*, *12*(2), 59–70. https://doi.org/10.52041/serj.v12i2.304
Páez, A. (2019). The pragmatic turn in explainable artificial intelligence (XAI). *Minds and Machines*, *29*(3), 441–459.
Pfaff, T. J., & Weinberg, A. (2009). Do hands-on activities increase student understanding? A case study. *Journal of Statistics Education*, *17*(3). https://doi.org/10.1080/10691898.2009.11889536
Prodromou, T., & Dunne, T. (2017). Statistical literacy in data revolution era: Building blocks and instructional dilemmas. *Statistics Education Research Journal*, *16*(1), 38–43. https://doi.org/10.52041/serj.v16i1.212
Rabin, L., Fink, L., Krishnan, A., Fogel, J., Berman, L., & Bergdoll, R. (2018). *A measure of basic math skills for use with undergraduate statistics students: The MACS. Statistics Education Research Journal*, *17*(2), 179–195. https://doi.org/10.52041/serj.v17i2.165
Raj, R. K., Parrish, A., Impagliazzo, J., Romanowski, C. J., Ahmed, S. A., Bennett, C. C., Davis, K. C., McGettrick, A., Pereira, T. S. M., & Sundin, L. (2019). Data science education: Global perspectives and convergence. *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, pp. 265–266. https://doi.org/10.1145/3304221.3325533
Rosenberg-Kima, R. B., & Mike, K. (2020). Teaching online teaching: Using the task-centered instructional design strategy for online computer science teachers' preparation. In *Teaching, Technology, and Teacher Education during the COVID-19 Pandemic: Stories from the Field* (pp. 119–123). Association for the Advancement of Computing in Education.

Sfard, A. (1991). On the dual nature of mathematical conceptions: Reflections on processes and objects as different sides of the same coin. *Educational Studies in Mathematics*, *22*(1), 1–36. https://doi.org/10.1007/BF00302715

Sfard, A., & Lavie, I. (2005). Why cannot children see as the same what grown-ups cannot see as different? Early numerical thinking revisited. *Cognition and Instruction*, *23*(2), 237–309. https://doi.org/10.1207/s1532690xci2302_3

Skiena, S. S. (2017). *The data science design manual*. Springer.

Srikant, S., & Aggarwal, V. (2017). Introducing data science to school kids. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 561–566). https://doi.org/10.1145/3017680.3017717

Sulmont, E., Patitsas, E., & Cooperstock, J. R. (2019a). Can You Teach Me to Machine Learn? *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 948–954). 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), 1–16. https://doi.org/10.1145/3336124

Tartaro, A., & Chosed, R. J. (2015). Computer scientists at the biology lab bench. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (pp. 120–125). https://doi.org/10.1145/2676723.2677246

Wiberg, M. (2009). Teaching statistics in integration with psychology. *Journal of Statistics Education*, *17*(1). https://doi.org/10.1080/10691898.2009.11889509

Wickham, H., & Grolemund, G. (2016). *R for data science: Import, tidy, transform, visualize, and model data*. O'Reilly Media.

Koby Mike
Faculty of Education in Science and Technology
Technion – Israel Institute of Technology
Haifa, Israel
mike@campus.technion.ac.il

APPENDIX 1. DATA SCIENCE FOR PSYCHOLOGY SCIENCE

The “Data Science for Psychology Science” specialization is a new initiative of the school of psychology sciences at a large research university in Israel (Mike & Hazzan, 2022). The motivation to develop such a specialization was threefold:

(a) To expose graduate psychology research students to new research methods that are emerging from the new discipline of data science.

(b) To expose graduate psychology research students, in general, and cognitive psychology research students, in particular, to state-of-the-art ML models that are inspired by the human brain and human thinking.

(c) To support the market demand for data scientists with a confirmed background in social sciences, in general, and in psychology science, in particular.

The specialization is divided into two courses. The first course, Computer Science for Psychology Science, was designed to fill the computer science knowledge gap (see Table A-1). The second course, Computer Science for Psychology Science, was designed to develop students’ ability to complete the data analysis cycle, that is, to explore, analyze, model, and predict using data they collected in Course 1 (see Table A-2). The courses were designed in a flipped classroom format, with a-synchronous pre-recorded lectures and weekly Zoom online meetings devoted to answering students’ questions and working on advanced exercises in small groups (Rosenberg-Kima & Mike, 2020).

Table A-1. Computer science for graduate psychology students - Topics and number of hours

Topic	Hours
Computational thinking	4
Website design and HTML	8
Website design and JavaScript	8
Python programming	16
Web scraping with Python	8
Computerized experiments with JavaScript	8
TOTAL	52

Table A-2. Data science for graduate psychology students - Topics and number of hours

Topic	Hours
The data science workflow	4
Table manipulation with Python	4
Visualization with Python	4
Statistical inference with Python	4
Principles of machine learning	8
Supervised machine learning	16
Unsupervised machine learning	4
Text analysis	8
TOTAL	52

APPENDIX 2. CODING SCHEME OF ANSWERS TO THE KNN COMPREHENSION QUESTIONNAIRE

Type of conception	Expression of conception	Coding scheme	Example
Process conception	(P1) Calculate distance from all samples	The text specifies the operation of calculating the distance from all training samples.	*"The algorithm checks the distance to all examples."* (Answer to Q.4)
	(P2) Pick the K nearest samples	The text specifies the operation of picking the K nearest neighbors.	*"If, for example, we defined that the number of neighbors (K) is five - the algorithm will check which five examples are the nearest."* (Answer to Q.1) *"Of the five images closest (most similar) to the new image, most (4) are a city."* (Answer to Q.2)
	(P3) Find the label of the majority	The text specifies the operation of choosing the label of the majority of samples.	*"… and according to the majority, will determine which type it belongs to."* (Answer to Q.1) *"… most (4) are a city."* (Answer to Q.2)
	(P4) Tune the hyperparameter K to improve performance	The text specifies the operation of tuning K to maximize performance.	*"It very much depends on the data; you should check the cross validation and decide accordingly. There is no one correct K."* (Answer to Q.4)
Object conception	(O1) Classification depends on similarity	The text specifies that classification is based on similarity.	*"[the decision] will be determined … by proximity."* (Answer to Q.1)
	(O2) Classification is determined by distance	The text specifies that classification is based on distance.	*"The similarity is expressed by a quantitative distance (squared) from the existing data."* (Answer to Q.1)
	(O3) The classification of a specific unknown example depends on K	The text specifies that different K may lead to different classifications.	*"[the decision] will be determined by the number of nearest items we defined (neighbors)."* (Answer to Q.1) *"He [Frank] missed k=1 and looked only at larger Ks."* (Answer to Q.3)
	(O4) The performance of the KNN algorithm for a specific K depends on the distribution of the data	The text mentions the property that accuracy depends on the properties of the data.	*"You need to know how the data is distributed."* (Answer to Q.4)
	(O5) The number of distance calculations depends on the number of training samples	The text mentions the property that the number of operations depends on the number of training samples.	*"… the number of square operations depends on the number of samples and not on the number of neighbors."* (Answer to Q.4)
	(O6) The number of distance calculations depends on number of features	The text mentions the property that the number of operations depends on the number of features.	*"You need to calculate 1000 distances; each distance has 4 characteristics which is 4000 square operations."* (Answer to Q.4)
	(O7) The number of distance calculations does not depend on K	The text mentions the property that the number of operations does not depend on the number of neighbors.	*"… the number of square operations depends on the number of samples and not on the number of neighbors."* (Answer to Q.4)