Int. J. Human–Computer Studies 196 (2025) 103438

Contents lists available at ScienceDirect

International Journal of Human - Computer Studies

journal homepage: www.elsevier.com/locate/ijhcs

"What are they not telling me?" Learning machine learning: Understanding the challenges for novices

Robert Cinca [a,*], Enrico Costanza [a], Mirco Musolesi [a,b], Muna Alebri [c,1]

[a] University College London, Gower Street, London, UK
[b] University of Bologna, Via Zamboni, Bologna, Italy
[c] United Arab Emirates University, Sheik Khalifa Bin Zayed Street, Abu Dhabi, United Arab Emirates

ARTICLE INFO

Keywords:
Learning machine learning
Machine learning
Explainable AI
Algorithms
Visualization
Black-box

ABSTRACT

Machine Learning (ML) is increasingly accessible to users with limited knowledge of its theoretical foundations. However, misapplying it can lead to negative consequences. This paper reports on a qualitative study designed to reveal challenges that novices encounter when learning about basic ML concepts and building their first models. Twenty participants were introduced to fundamental ML concepts for classification through an interactive tutorial involving an off-the-shelf GUI application, built their own ML model for a shape gesture dataset, and participated in a semi-structured interview. A thematic analysis revealed insights into these challenges, particularly around problem selection and multi-dimensionality, but also around what constitutes ML, algorithm selection, cross-validation, and interpreting visualizations. Despite these and other misconceptions, participants reflected on good model building practices, discussing that algorithm selection might require knowledge and context and that input features may introduce bias. We discuss the findings' implications for the design of ML tools for novices.

1. Introduction

Machine Learning (ML) is being used in an increasing number of domains, such as cyber-security, diagnosing disease, document classification, and language translation (Mitchell, 2019; Barreno et al., 2010; Koller and Sahami, 1997; Srividya et al., 2018). The proliferation of AutoML tools and GUI's such as KNIME (Berthold et al., 2009), RapidMiner (Hofmann and Klinkenberg, 2016), and Orange (Demšar et al., 2013) allow novices to apply their own models even with limited theoretical knowledge of ML (Carney et al., 2020). This opens them up to the risks of misapplying ML, for example, through building models with biased outcomes for marginalized groups (Angwin et al., 2016; Noble, 2018; Keyes, 2018). Although programs built by experienced ML practitioners can also fall foul of such outcomes, those without any formal training in ML are more likely to experience challenges when designing or using ML applications, as they have limited knowledge of the theoretical foundations and the practical implications of using these techniques (Patel et al., 2008a; Amershi et al., 2014; Yang et al., 2018).

We therefore present a qualitative study designed to shine light on the challenges novices in ML encounter when applying ML in an interactive environment, with implications to the design of ML tools for novices. Prior research investigating ML challenges identified challenges that more experienced ML users face (Patel et al., 2008b,a; Amershi et al., 2019; Yang et al., 2018; Veale et al., 2018), the difficulties a lay user might face when interacting with ML applications (Rader and Gray, 2015; Sanchez et al., 2021; Oh et al., 2020), or around children's understanding of ML concepts (Hitron et al., 2019; Touretzky et al., 2019). Instead, this work aims to introduce fundamental concepts required for building classification models to novice ML users.[2] It also examines the difficulties that these novices encounter as they learn about ML.

Twenty volunteers interested in learning and applying ML took part in our study. They were introduced to ML concepts through a take-home tutorial designed to be completed within two hours. The tutorial was iteratively designed and centered around a widely used graphical user interface (GUI) application for ML. This application served as a tool to uncover insights related to the practical implementation of ML concepts and provided interactivity. Participants demonstrated understanding through a series of questions and exercises; by building their

* Corresponding author.
E-mail address: robert.cinca.14@ucl.ac.uk (R. Cinca).

[1] The work was conducted when the author was at University College London.

[2] We refer to *novices in ML* as those who have no prior knowledge of ML but who have expressed an interest in learning or implementing ML (e.g., in their own domains) whilst possessing relevant skills (e.g., within statistics and data analysis).

https://doi.org/10.1016/j.ijhcs.2024.103438
Received 19 June 2024; Received in revised form 8 December 2024; Accepted 11 December 2024
Available online 17 December 2024

own ML model for a shape gesture dataset; and through a subsequent semi-structured interview that introduced participants to a real-world bike sharing scheme dataset. The researchers then analyzed participant responses using thematic analysis techniques to identify the difficulties encountered by participants. Specifically, we address the following research question: what challenges do ML novices encounter when learning and applying ML?

The thematic analysis revealed insights into ML challenges participants experienced, such as problem selection and multi-dimensionality of both algorithms and data, but also around what constitutes ML, algorithm selection, cross-validation, and interpreting visualizations. Despite challenges and misconceptions, participants reflected on good model building practices, discussing how algorithm selection requires knowledge and context, input features can introduce bias, and the importance of trust in an algorithm. We offer three key contributions. Our *first* contribution is a series of findings that illustrate the various challenges that novices in ML face when learning and applying ML. *Secondly*, a series of misconceptions of what ML can do, showing the risks of misapplying ML. For example, participants misunderstood ML models to be less biased than humans, or they stumbled on problem selection, expecting ML to solve hard problems rather than mundane, repetitive tasks. Our *third* contribution is a series of implications for the design of ML tools for novices. This includes guiding novices to solve simpler problems using simpler algorithms that can achieve similar performance to more complex algorithms, while being easier to understand and providing explainable outputs.

2. Related work

Relevant research exists on understanding the challenges faced by ML users, on how novice users engage with ML through interactive ML (IML) tools and on ML explanations that facilitate model building. We review this work below.

2.1. Understanding the challenges of ML for users

Previous studies on the use of ML by researchers identified three major challenges both when creating models and when interacting with them (Patel et al., 2008a,b; Amershi et al., 2019): (1) a difficulty in understanding and applying iterative exploration processes when creating models; (2) a lack of understanding of ML concepts, especially for those who are not mathematically trained; and (3), a difficulty in evaluating model performance. In addition, prior work has investigated the current approach of algorithmic fairness for ML practitioners in public sector decision-making, finding a disconnect between organizational needs and current ML building practices, such as how domain experts should be modifying their ML models to account for changes in the data over time (Veale et al., 2018). While this previous research focused on software developers already experienced with applying ML, our work aims to identify the challenges, both similar and different, that novice users without any hands-on ML experience face when learning and applying ML.

Substantial research has investigated the challenges faced by less experienced users when interacting with ML applications, such as how a general lay audience understands the Facebook news feed algorithm (Rader and Gray, 2015). Research conducted by Oh et al. (2020) looked at how different kinds of users (ML experts, domain experts and lay users) reason about AI algorithm results, showing that their understanding of the model depended on their own field of knowledge, and when users struggled to understand the steps of the algorithm, there was a wider gap between how users think the AI will predict and how it actually predicts (Oh et al., 2020). Work by Sulmont et al. (2019) found that novices could be taught how algorithms function, but that they struggled with higher-level design decisions when constructing models. However, the ML novices in these studies did not get to build their own models, therefore having no agency in the training process. Another challenge identified by prior work is that novices rely solely on accuracy measures of output to determine whether the selected algorithm is a good fit for the dataset (Krause et al., 2016; Yang et al., 2018), leading to the deployment of problematic models (Yang et al., 2018). This is partly due to the ease of using summary accuracy statistics that often obscure important information about a model's behavior, which creates a dissociation between performance and data, leading to practitioners taking a trial-and-error approach to model building (Ren et al., 2016). A proposed solution to debugging models beyond summary statistics is to visualize the error distributions, as it can reveal misleadingly high accuracies in skewed datasets (Ren et al., 2016). For instance, a practitioner can use a visualization from Squares (Ren et al., 2016) to observe that although a naive Bayes model and a random forest model both achieve the same accuracy, the random forest model may produce many predictions with nearly equal probabilities across multiple classes. This means a small change in the input could cause a true prediction to become false, suggesting the naive Bayes model may be more robust in this case. Sanchez et al. (2021) investigated lay users' fine-tuning of a neural network for classifying sketches, finding that participants adopted diverse strategies, but that understanding the fundamental properties of neural networks was a challenge. In contrast, our work exposes novice users to a wider array of ML concepts, such as train/test split, cross-validation and parameter selection. These users are then guided through the model building process using four commonly employed simple classification algorithms (Yang, 2018). This approach is taken because simpler algorithms often achieve comparable performance to more complex algorithms on structured data (Rudin, 2019).

2.2. Interactive ML and machine teaching

IML tools enable users to create their own ML models without requiring any programming knowledge or understanding of ML algorithms, thereby making ML accessible to novice users (Amershi et al., 2011). Fails and Olsen asserted the importance of human involvement in providing training data and proposed an interactive system that allows users to train, classify, and correct classifications in a real-time iterative loop (Fails and Olsen, 2003), known as IML (Interactive ML). Some examples of IML tools and research that have emerged in recent years include: Weka, which provides a GUI for ML (Hall et al., 2009); Wekinator (Fiebrink and Cook, 2010), an application that lets users record music gestures as input to an ML model; TensorFlow Playground, designed to teach novices about neural network behavior through hands-on interaction (Hohman et al., 2018); platforms like Lobe (Microsoft, 2020) and Elements of AI (Heintz and Roos, 2021), which are interactive tools enabling ML novices to create basic models; Simple ML for Sheets (Guillame-Bert et al., 2022), which integrates ML capabilities into Google Sheets without requiring ML knowledge or coding; the Teachable Machine project (Carney et al., 2020), which explains ML concepts to users as they build models and has been effective in teaching ML to children (Vartiainen et al., 2020); Marcelle (Françoise et al., 2021), a tool for creating IML toolkits; and visualization tools such as CNN Explainer for understanding neural networks (Wang et al., 2020), visualizations showing how recurrent neural networks store information (Madsen, 2019), and AI-designed user interfaces (Carter and Nielsen, 2017). We chose Weka's GUI (Hall et al., 2009) as a way to investigate the needs and challenges of ML novices in our study. This decision allowed us to create a tutorial introducing essential ML concepts, whilst Weka' interactive interface enabled participants to practically build models without requiring any programming knowledge.

Prior research has leveraged IML tools and GUIs to study user interaction with ML, providing insights for the design of these tools. Fiebrink et al. used Wekinator (Fiebrink and Cook, 2010) to explore how musicians train a model through recording audio samples and other gestures as input (Fiebrink et al., 2009). After manually labeling

a few input samples, the user-selected algorithm classified the rest of the inputs using the model trained on the training set created by participants. This research demonstrated the practical advantages of IML tools, such as the automated classification of audio samples which can save musicians significant time. While this study explores model building by novices, its topical focus on the transcription of musical sounds resulted in many fundamental ML concepts being left out, whilst the training set was small, given that it was manually created by participants. Researchers have used IML tools to observe how children engage with ML, with work by Touretzky et al. proposing five key ML concepts that children should understand as they interact with ML, including maintaining simple representations of the world and that ML algorithms learn from data (Touretzky et al., 2019). A study involving the widely used Scratch platform (Resnick et al., 2009) had children use the GUI to build their own applications. The researchers found that equipping users with data science skills enabled them to create analyses that better suited their needs (Dasgupta and Hill, 2017), as a significant portion of building ML models involves data processing. Another study around children's interaction with an IML tool found that the best understanding comes from direct experience in interacting with the system and that black-boxing too many processes is like black-boxing all of them (Hitron et al., 2019). There is also research being conducted around interactive machine teaching, which permits lay users or domain experts to build their own ML models by directly involving and guiding them in the process (Ramos et al., 2020). For example, Ramos et al. (2020) built an integrated teaching environment designed to observe how novice users can build an image recognition model for postal addresses, finding that employing intrinsic human capabilities (judgment, insight, foresight, and sensemaking) within the interactive machine teaching process helps domain experts build ML models. Similarly, work by Martins and Von Wangenheim (2023) found that high school students were able to successfully apply basic ML concepts using a hands-on approach to learning, confirming the benefits of IML tools for teaching. Our work explores learning by doing (Schank et al., 1999) with adult novices, observing how they apply core ML concepts using an IML tool, without abstracting key model building steps, to identify the challenges they face.

2.3. ML explanations

There is a significant body of work in explainable AI that focuses on providing users with interpretable outputs that help them improve a system's performance, and explanations that help users understand why they are getting a specific output. The widening gap between theory and practice in XAI, due to the diversity of methods and techniques, complicates method selection for novice users, as highlighted by Retzlaff et al. (2024), who provide design guidelines for method selection in their study. Prior research into the efficacy of explanations that help users understand a model's output has shown that answering *why* and *why not* questions can improve intelligibility for the end-user (Lim et al., 2009). Researchers tested how to provide explanations with Laksa (location, activity, connectivity, social awareness) (Lim and Dey, 2011), a context-aware mobile app that uses various features (e.g., calendar data, phone sensor data) to determine whether a work colleague is available for a meeting. The authors also found that, to reduce users' frustration, it was important to only provide explanations if users can leverage that information and change their behavior (Lim and Dey, 2011), a finding confirmed by Kulesza et al. (2015). However, Laksa itself is non-interactive: users are unable to modify any parts of the data pipeline, such as the algorithms used, the inputs, and the outputs. Past research has looked at the effect on mental model soundness of how an intelligent agent works when users are provided ML explanations in the form of a training program (Kulesza et al., 2012). The authors found a strong link between mental model soundness and user understanding, and that even a short training program helped improve participants' mental models of how the system works.

3. Study design

To address the research question on the challenges faced by ML novices when learning about ML, a remote qualitative study was designed. The method is outlined below.

3.1. Participants

The study recruited twenty participants using a two-pronged approach: an initial recruitment was done by distributing a call for participants through mailing lists in departments where learning and applying ML might be of interest, such as Information Studies, Geography, and Psychology. To recruit novices, departments where ML is a prominent topic were intentionally avoided. The participant pool was then expanded through snowball sampling, leveraging word-of-mouth referrals from previous study participants. In a bid to attract participants who were genuinely interested in learning about ML, we did not offer compensation. Given our recruitment strategy, we attracted participants who wanted to learn the basics of ML, and who wanted to apply ML to their own work. To ensure we recruited participants who were novices of ML, the consent form asked participants to confirm they did not fall under the exclusion criteria of the study. This meant participants could not have any hands-on experience with ML, including the use of GUI's like Weka that help build ML solutions. Specifically, participants had to confirm they had not taken any for-credit courses in ML, or had experience in applying ML algorithms, whether in a workplace setting or for personal use. Please refer to Table 1 for a summary of participants, including their education level, technical background and current occupation.

This cohort, though highly educated, had no prior knowledge of ML according to our exclusion criteria, whilst possessing relevant skills (e.g., within statistics and data analysis) that would make them suitable to learn and apply ML within a study of two hours. While the mathematical concepts covered in the tutorial were designed for individuals with a high school-level understanding, six participants discussed how their familiarity with statistical concepts like probability distributions, Euclidean distances, and false positives/negatives provided them with a grounding that eased their learning of ML in the study. Another two participants discussed how their prior work as data scientists meant they were familiar with analyzing data, allowing them to focus on understanding the ML algorithms presented in the study. However, not all participants had relevant backgrounds. Four participants mentioned that their non-mathematical backgrounds (such as social sciences and multimedia design) did not aid them in completing the study. For instance, two of these participants noted that the terminology used to explain the concepts differed from what they were accustomed to in their respective fields.

Referring back to our definition of novices in ML as described in the introduction, the exclusion criteria identified ML novices who did not have any prior knowledge of the fundamentals of ML, whilst the recruitment strategy attracted those interested in learning and applying ML, most of whom had backgrounds that would ease this learning process. Although the prior knowledge possessed by some participants may have given them an advantage in completing the study, this group is also more likely to recognize the potential of ML and incorporate it into their own work-related domains.

3.2. Apparatus

The study provided participants with a static tutorial that explained the fundamental components required to build ML classification models. Participants engaged with a series of inquisitive questions and exercises that required participants to try things practically in Weka's GUI, which were subsequently discussed in the follow-up interview. Some exercises were designed to encourage reflection on ML principles, while others aimed to facilitate practical experimentation and assess

Table 1
Information about the participants involved in the study, identified by their participant ID number, including details such as occupation, background, sex, education, and time spent completing the tutorial and interview.

ID	Occupation	Background	Sex	Education	Tutorial duration	Interview duration
01	Researcher in Data Visualization	Data Science	M	PhD	2 h	54 m
02	Researcher in Creativity Tools	HCI	F	PhD	4 h	40 m
03	Researcher in Digital Interruptions	HCI	F	PhD	2.5 h	65 m
04	Researcher in Virtual Reality	HCI	M	PhD	2 h	60 m
05	Psychology Student	Social Sciences	F	UG	2 h	49 m
06	HCI Student	Multimedia Design	F	M	3 h	47 m
07	Researcher in Autonomous Driving	Electrical Engineering	M	M	4 h	63 m
08	Researcher in Cyber Security	Electrical Engineering	F	M	2 h	59 m
09	Psychology Student	Social Sciences	F	M	3 h	58 m
10	Product Manager	Data Science	M	MBA	2 h	59 m
11	Researcher in Accessibility	HCI	M	PhD	3 h	51 m
12	Researcher in Cyber Security	Computer Science	F	PhD	1.5 h	40 m
13	Researcher in Information Studies	Archivist	F	PhD	2 h	75 m
14	Archivist	Social Sciences	F	UG	2 h	43 m
15	Researcher in Waste Management	Biotechnology	F	PhD	1 h	40 m
16	Information Studies Student	Law	F	M	2 h	65 m
17	Archivist	Conservation Science	F	PhD	2.5 h	54 m
18	Psychology Student	Social Sciences	F	M	2 h	46 m
19	Software Engineer	Computer Science	M	UG	2 h	61 m
20	Archives Student	Digital Marketing	M	M	4 h	42 m

participants' understanding of concepts through binary right-or-wrong tasks. A list of these exercises is presented in Table 2. The rationale behind the design of these exercises was to assess understanding and encourage participants to reflect on ML. Both the tutorial and exercises were developed iteratively, incorporating feedback from students and pilot participants, some of whom were experienced ML users. Taking inspiration from Rosson et al. (1990), the study employed a minimalist instructional approach (Carroll, 1990), which has been shown to help participants understand difficult concepts in a short amount of time (Rosson et al., 1990), by: (1) introducing all concepts through a simple use case (the Iris dataset Dua and Graff, 2017), (2) minimizing upfront instruction and using a spiral approach to incrementally reveal more complexity, (3) supporting reasoning and improvising by providing incomplete explanations and exercises that require the learner to figure things out on their own, (4) supporting error recognition and recovery by anticipating common errors, and (5) leveraging prior knowledge on statistics that participants may possess to aid in the explanation of ML concepts.

The iteratively designed tutorial contained four simple classification algorithms: kNN (Keller et al., 1985), naive Bayes (Joachims, 1998), decision trees (Dietterich, 2000), and random forests (Pal, 2005). These algorithms were chosen because their outputs directly correlate with inputs, the notion of discrete features is easily conveyed, and the results can be visualized using a classification output boundary visualizer. While neural networks constitute a classification approach with widespread adoption, their introduction was avoided in the tutorial as they necessitate extensive datasets and are frequently linked to deep learning, a subject intentionally omitted to maintain the tutorial's conciseness. Classification also provides real-world use-cases relevant to our participants, such as intrusion detection in cyber-security (e.g., classification of threats Barreno et al., 2010), document classification in archival research (e.g., classifying documents based on topic Koller and Sahami, 1997), and within psychology (e.g., predicting the onset of mental health conditions using classifiers such as decision trees, naive Bayes and kNN Srividya et al., 2018). A summary of the concepts covered in the tutorial is presented below, while the full material as presented in this study is available as supplementary material.

1. Familiarizing with and loading the Iris dataset (Dua and Graff, 2017) into Weka. This dataset contains 150 flower samples, four features (sepal length, sepal width, petal length, and petal width) and was split equally among the three output classes: Iris Setosa, Iris Versicolor, and Iris Virginica. This dataset was chosen due to the low number of input features and samples, given that simplification is commonly used in introductory ML material (Witten and James, 2013; Chollet, 2021; Géron, 2022).
2. Plotting flower features (e.g., petal length against petal width) on scatter plots so participants could visually identify rules that can separate the output into the three output classes. An example of a plotted scatter plot is shown in Fig. 1. This step was designed to prompt participants to reflect on how a classification algorithm might separate the output.
3. Learning about the process of feature selection and its importance. This concept was introduced to spur participants to reflect about possible sources of bias in the inputs.

Table 2
A list of questions that participants were asked in the tutorial.

Tutorial questions
Q1: How would you describe what ML is?
Q2: How would you use ML technology in your work?
Q3: Are there cases where we should not use ML?
Q4: Given this scatter plot of sepal width vs sepal length, could you define rules to classify new data items as belonging to either of these 3 classes?
Q5: Now click on the petal width vs petal length plot. How does this compare?
Q6: Based on the visualizations of features in the previous section, why do you think these features are most important?
Q7: Why is it useful to have a train and test set?
Q8: Do you understand how the kNN algorithm works?
Q9: Please briefly write about the algorithm's accuracy. Do you think the accuracy achieved is good? What would you say the threshold would be?
Q10: Please have a look at the confusion matrix and write down how you think it conveys information on the model accuracy.
Q11: Why is it better to run the model using cross-validation?
Q12: In terms of the dataset, imagine that fewer data points had been collected. What would you expect to change and why?
Q13: Can you think what happens to the training as the value of k in kNN is changed? What happens when k=1? What about when k is very large?
Q14: How does the accuracy change as you vary k?
Q15: Do you understand how the naïve Bayes algorithm works?
Q16: What observations can you make about the decision tree algorithm?
Q17: Why do you think a random forest performs better than a decision tree?
Q18: Please look back at the scatter plots you generated earlier for the Iris dataset. Where would you manually draw boundaries to separate the classes?
Q19: After running the boundary visualization for different algorithms and different features, please comment on the decision boundaries.
Q20: For the shapes dataset exercise, discuss the levels of misclassified samples.
Q21: What was the best performing algorithm?
Q22: Why do you think this algorithm performed best?
Q23: Do you think this model is 'good enough' to deploy for the application?

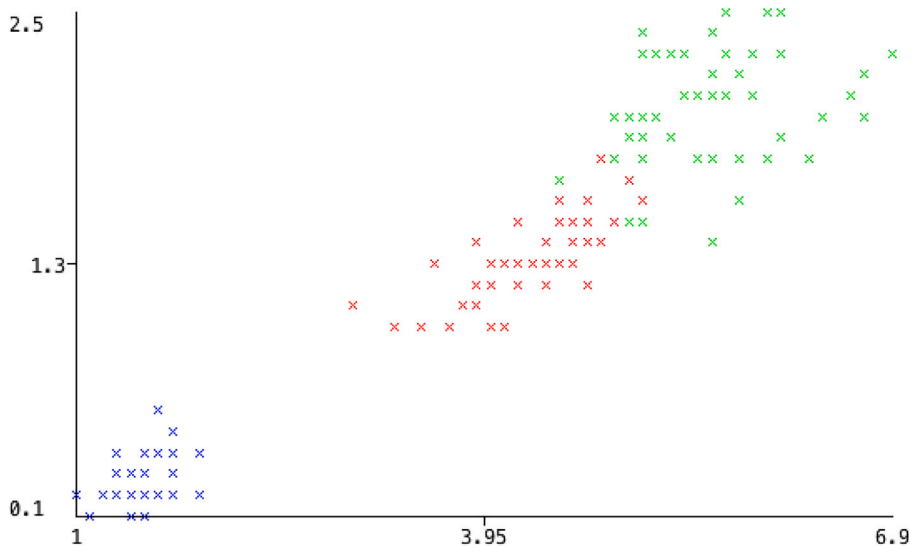

Fig. 1. An example of a scatter plot visualization that was used so that participants can think of 'rules' that would separate the outputs into 3 distinct classes. This visualization represents petal length against petal width, and the coloring of individual samples represents the output class.

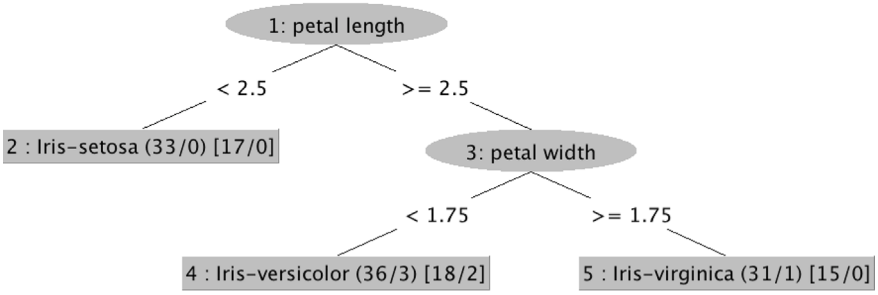

Fig. 2. An example of a simple decision tree visualization that participants computed themselves. This example only uses two features to separate the output classes, petal length and petal width.

4. Splitting the Iris dataset into two subsets: one for training a classifier, and the other one for testing it. This was needed for participants to evaluate model performance later on.
5. Understanding and training the k-Nearest Neighbor (kNN) algorithm (Keller et al., 1985), and evaluating its performance using accuracy. Participants began with kNN for the Iris dataset because the algorithm's outputs exhibit a clear relationship with the discrete input features, and was relatable to the previous task of charting the input features on a scatter plot.
6. Understanding and visualizing confusion matrices, and as a way of evaluating model performance, in addition to accuracy. An example of a confusion matrix can be seen in Fig. 3.
7. Introducing the concept of k-fold cross-validation (Refaeilzadeh et al., 2009), as a way for participants to ensure that a model is generalizable and not overfitting the training set.
8. Selecting parameters: introducing the parameter k for the kNN algorithm, a parameter used to determine the number of samples used to predict the output of a test sample. Participants were then invited to run the model multiple times, each time with different parameter values for k, as a way of tuning the model and improving performance. The effect of changing the parameter is described in the tutorial through the visualization shown in Fig. 5, and was employed as a way to introduce some additional complexity to model building.
9. Considering a small number of alternative classifiers, and comparing them in terms of performance. Participants were introduced to naive Bayes (Joachims, 1998), decision trees (Dietterich, 2000) and random forests (Pal, 2005). These were explained through short text snippets and visualizations such as the decision tree visualization in Fig. 2. The Iris dataset is commonly used to demonstrate decision tree algorithms, as the tree outputs align directly with the discrete input features. Random forests were chosen to add complexity to its simpler variant, the decision tree. Naive Bayes was added as an additional commonly employed yet simple classification algorithm (Yang, 2018).
10. Comparing and discussing the visual representations of different classifiers' output. Participants visualized classification outputs using the feature space boundary visualizer shown in Fig. 4. This was needed for random forests, which pose difficulties for direct visualization owing to their high-dimensional nature, unlike the relative ease of visualizing decision trees.

Prior work has shown that working with large, real-life datasets is challenging for ML practitioners (Rojas et al., 2017). As a result, we also tasked participants with building their own classification model for a shape gesture dataset that was much larger than the Iris dataset, to test knowledge acquired in the tutorial and provide a potential real-world application of ML. Drawing inspiration from Wekinator's gesture recognition toolkit (Fiebrink and Cook, 2010), shape gestures could provide a viable approach for users to execute commands on smartphones (for example, drawing a square gesture to instruct the phone to set an alarm for 8:00). This 10,000-sample dataset included ten different shape classifications as the output along with eight input features that measured coordinate positions of corners and their interior angle. The shapes used included several types of quadrilaterals and triangles, circles, and pentagons. The dataset was artificially generated, allowing for controlled design, while incorporating added noise to emulate real-world

```
=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
 998   2   0   0   0   0   0   0   0   0 |   a = square
 181 819   0   0   0   0   0   0   0   0 |   b = rectangle
   1   1 954  44   0   0   0   0   0   0 |   c = rhombus
   1   5 225 769   0   0   0   0   0   0 |   d = parallelogram
   0   0   0   0 1000   0   0   0   0   0 |   e = circle
   0   0   0   0   0 1000   0   0   0   0 |   f = pentagon
   0   0   0   0   0   0 901  68  16  15 |   g = up_triangle
   0   0   0   0   0   0  82 883  19  16 |   h = down_triangle
   0   0   0   0   0   0  20  29 907  44 |   i = right_triangle
   0   0   0   0   0   0  22  26  66 886 |   j = left_triangle
```

Fig. 3. A confusion matrix for a random forest model on the shapes dataset shows it struggles to distinguish triangles and quadrilaterals but correctly predicts circles and pentagons.

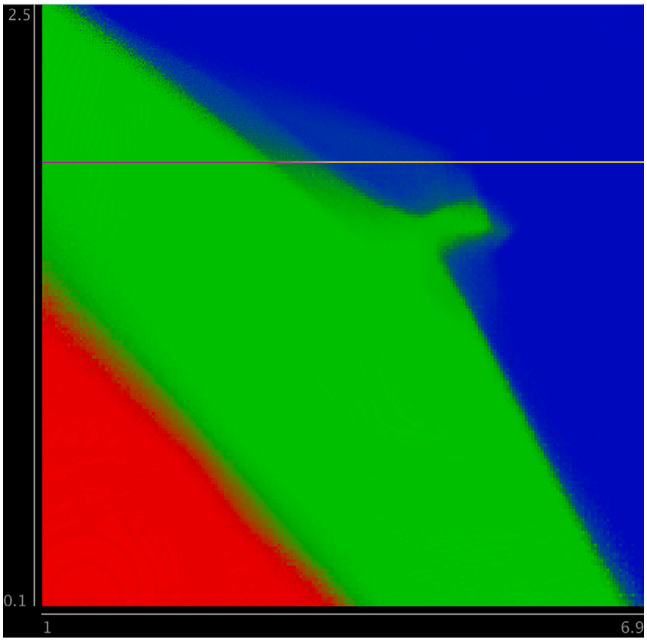

Fig. 4. An example of a boundary visualization of the kNN algorithm. The different colors represent the predicted output of the classifier, given an X (petal length) and Y (petal width) value. The blurred boundaries indicate a blend of predictive outputs.

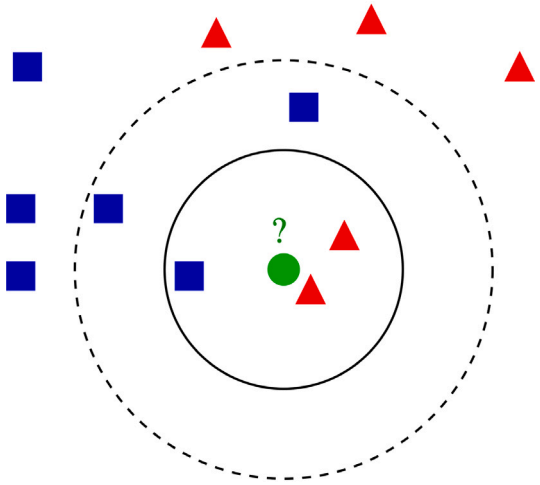

Fig. 5. The tutorial's illustration of kNN, aimed at demonstrating the impact of adjusting the parameter *k*, proved helpful to some participants but caused confusion for others. Sketch by Antti Ajanki [Public domain], via Wikimedia Commons (https://commons.wikimedia.org/wiki/File:KnnClassification.svg).

conditions, resulting in imprecise shape measurements. In addition, although participants still had access to the tutorial when building the gesture classifier, this dataset was much larger than the Iris dataset that participants had been experimenting with, both in terms of feature count and sample size, creating an extra challenge for participants as the results were more difficult to interpret. The aim of the exercise was for participants to use the concepts that they learnt during the tutorial to develop an accurate model that successfully classified shape gestures, encouraging participants to engage in self-directed exploration and problem-solving. Though practical applications often involve greater intricacy and broader scope than the examples employed by the tutorial, simplification is a common pedagogical approach employed in introductory ML resources like textbooks (Witten and James, 2013; Chollet, 2021; Géron, 2022), with previous research finding that low-dimensional examples are more effective at explaining ML concepts as it is challenging for users to conceptualize high-dimensional data (Liu et al., 2016). The tutorial was designed to cover key ML model building process using classical ML algorithms (kNN, naive Bayes, decision trees, random forests) in about two hours. In line with the minimalist instructional approach (Rosson et al., 1990; Carroll, 1990), ML concepts in the tutorial were contextualized around the Iris dataset due to the short period of time to run the study, and to improve participant engagement by making the study less theoretical. Although employing an alternative tutorial could potentially generate a different set of participant challenges and reflections, our selected tutorial served as a means to extract a series of insights and observations within the constraints of a two-hour session.

3.3. Study procedure

Participants devoted around two hours to the take-home tutorial detailed in Section 3.2, with no enforced time limit permitting them to spend longer if they wanted. After completing the tutorial, participants participated in a semi-structured interview that lasted approximately one hour. Table 1 shows the time participants spent being interviewed. To boost completion rates and ensure the interview took place within a week of the tutorial, keeping the material fresh in participants' minds, the interview was scheduled before the study material was released. Most participants completed the tutorial on the day of the interview, or a few days before. Participants were asked to share their answers in advance of the interview.

In the interview, participants were asked questions on their experiences with the tutorial and their understanding of ML concepts, specifically discussing and reflecting on some of the answers they gave to exercises, and whether any of the answers have changed since completing the tutorial. Participants were allowed to refer back to the tutorial and their written responses to the tutorial exercises. To further assess participants' comprehension and reflections on ML model construction, participants were tasked with describing their approach for building a model using a real-world bike sharing dataset. This was

a much larger dataset than the two datasets presented in the tutorial and was sent to participants during the interview in the form of two CSV files. Due to time constraints and the need to pre-process the dataset – an aspect beyond the scope of our study – we did not ask participants to build a model during the interview. Instead, we asked them to think aloud and describe the steps they would take to build a model for this more complex dataset, which contained 62,809 unique data samples and 46 features. Finally, participants were asked a series of questions around their own experiences with ML, including if they have interacted with end-user applications of ML and how they would apply ML to their own interests and domains. The full interview guide is available in the supplementary material.

3.4. Analysis

The interviews were audio-recorded, fully transcribed, and analyzed using inductive thematic analysis (Braun and Clarke, 2006), following the six-step data analysis process outlined by Braun and Clarke (Clarke and Braun, 2013). Once familiarized with the transcripts, the primary researcher generated an initial set of codes through open coding, with sample checks performed by the other researchers to ensure that codes were suitable in addressing the research question. Inter-coder reliability metrics were not applied because the open-ended nature of the responses suggests that there is no single "correct" way to interpret the data. The open coding approach involved condensing participant statements into a set of concise, representative keywords, with the goal of organizing the data into tags that could help address the research question. A combination of descriptive and interpretive codes was applied. For instance, descriptive codes like "accuracy" and "output" were used to summarize participants' comments on the performance of ML models, while interpretive codes such as "challenge" and "confusion" sought to capture the emotional responses participants had, particularly their difficulties in understanding ML. These codes were added to an expanding list of used codes, which was employed to help the researcher maintain consistency in the coding process. Even so, an initial list of codes was whittled down to 72 codes (full list available in the paper supplementary material), based on merging codes that were misspelled or that had the same meaning. Codes that had fewer than five quotes were checked for novelty, and whether they were entirely covered by other codes. The final list of codes was deliberated among the researchers, who then independently searched for themes by identifying patterns across the codes and data, clustering the codes into groupings of four to eight categories. These were then reviewed and discussed, resulting in a collective decision to define four themes, which are outlined in the findings.

4. Findings

Participants were motivated by the study's hands-on approach, which involved applying learnt knowledge using the graphical tool and completing a series of short exercises. As shown in Table 1, eight participants spent longer than the required two hours, with three of these participants allocating approximately four hours. These eight participants were highly motivated; they all successfully completed the study's exercises and provided valuable insights. Twelve participants spent up to the recommended time on completing exercises, with seven of them either unable to move further and stopping, or not managing to finish all the exercises within the allocated time. Overall, thirteen participants completed the tutorial in its entirety, including all the exercises, and were able to explain how they would build their own ML models. Of those who did not finish, two participants had technical issues, including problems installing the tool or their computer being too slow. The other five participants faced challenges grasping ML concepts; three of them discontinued the study after two hours, while the remaining two became completely stuck and abandoned the study midway. Table 3 presents a summary of participants' responses to the practical questions in the tutorial, highlighting that some exercises were more challenging than others. The findings were grouped into four themes, each presented in one of the following subsections.

4.1. Understanding ML algorithms

This theme arose from participants' understanding of specific algorithms presented in the tutorial. Codes also emerged from their reflections on the bigger picture, including doubts and challenges regarding what constitutes ML. This section focuses on participants' demonstrated understanding of ML, as evidenced by measurable exercises and questions from the tutorial and interview, rather than their self-reported understanding.

4.1.1. Specific algorithms

Participants valued learning about ML algorithms and demonstrated an understanding of what is often considered an abstract concept: "*an algorithm is just a way of building something, it is just a series of steps that you go through, it's not alchemy, it's not magic, and I think I find that quite reassuring*" [P13]. Participants found some algorithm concepts easier to understand if they already had a mental model of how it might work, a finding consistent with prior research on mental models (Lim et al., 2009). A number of participants' comments were specific to the four different classification algorithms that were introduced in the tutorial, namely kNN (Keller et al., 1985), naive Bayes (Joachims, 1998), decision trees (Dietterich, 2000) and random forests (Pal, 2005). With kNN, seven participants were unsure whether the algorithm bases its output on input samples within a certain distance (incorrect) or simply several sample points that are deemed closest (correct). Meanwhile, participants with a recent statistical background found naive Bayes intuitive due to how a Gaussian naive Bayes classifier will model the probability distribution of the input features to follow a normal probability distribution. Other participants, especially those less familiar with statistics, preferred the more visual explanation of naive Bayes using histogram sketches of the input data: "*it was very useful to have the graph and be able to see that OK at five Sepal length it is definitely Setosa because you can see that the blue line is a lot bigger so that kind of helps you process how the algorithm works*" [P02]. This shows the importance of explanatory visualizations for understanding ML algorithms.

Unsurprisingly, the decision tree algorithm was the most intuitive algorithm for most participants, as they felt that it was similar to how a human would approach the problem: "*the decision tree is easy because it's more how we would work [...]. It's more rules based, you have definite rules, and you say if it's X, go this way, if it's Y, then go that way*" [P13]. After being given a visualization of a scatter plot, as seen in Fig. 1, and a decision tree visualization, as seen in Fig. 2, participants saw a direct link between the exercise that asked them how they would split the data on a scatter plot and the way the decision tree algorithm works: "*I really like the decision tree because I answered one of the first questions where I stated that if this width was less than that and was greater than this, [...] kind of like an explanation of how a computer classifies this dataset*" [P06].

Regarding random forests, participants provided several reasons in their written responses during the take-home tutorial for why a random forest outperforms a decision tree. Table 4 provides a list of participant responses, ordered by the frequency of the answer. However, challenges persisted, as eight participants in the follow-up interview had difficulty understanding how the random forest algorithm works, struggling to visualize its multi-dimensional nature due to its composition of multiple decision trees. In summary, ML algorithms that are easily visualized or resemble a novice's approach to the problem were easier to understand, whilst more advanced ML algorithms were not, even if they were based on the simpler ones.

Three participants questioned whether understanding the theoretical details of ML algorithms is even needed, if one can easily apply them using trial-and-error: "*if the cost of testing out each algorithm would have been much higher, like an hour, then probably I would have stepped back and thought, maybe I need to understand what I'm doing, but because for me the cost is just clicking on an algorithm like clicking on a button then I thought well I can work out which one is the most accurate just by trying it*

Table 3
Accuracy and completion rates analyzed by individual questions and participants.

Question	Accuracy	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20
Q4	**78**	1	1	1	0.5	1	1	1	0.5	1	0	0.5	1	1	0	–	–	1	1	0.5	1
Q5	**100**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	–	–	1	1	1	1
Q6	**84**	1	1	1	0	1	1	1	1	–	–	1	1	0.5	1	–	–	1	0.5	1	0.5
Q7	**100**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	–	1	1	1	1	1
Q8	**47**	0.5	0.5	0.5	0.5	0.5	0	–	1	0.5	0.5	0.5	–	0.5	0	–	0	1	0	1	0.5
Q9	**86**	1	1	1	1	0.5	1	1	1	1	1	1	0.5	0.5	0.5	–	–	1	0.5	1	1
Q10	**68**	1	1	1	1	1	1	0	1	1	0	0.5	1	0	0	–	–	0	–	1	1
Q11	**76**	0.5	1	1	1	0.5	0.5	1	–	0.5	–	0.5	0.5	0.5	1	–	1	1	1	0.5	1
Q12	**75**	–	–	–	–	1	1	0.5	–	1	1	1	1	0	0.5	–	1	1	1	0.5	0
Q13	**62**	0.5	1	0.5	0.5	0	0.5	1	–	1	–	1	0	0.5	0.5	–	0.5	1	1	0.5	0.5
Q14	**97**	1	1	1	1	1	1	1	–	1	0.5	1	–	1	1	–	–	1	1	1	1
Q15	**83**	0.5	1	1	1	0.5	1	0.5	–	0.5	–	0.5	–	1	1	–	1	1	–	1	1
Q16	**93**	1	1	1	1	1	1	1	–	1	–	1	–	1	0.5	–	0.5	1	–	1	1
Q17	**80**	1	1	0.5	0.5	0.5	0.5	1	–	1	–	1	–	0.5	0.5	–	1	1	–	1	1
Q18	**83**	1	1	0.5	1	1	0.5	0	–	1	–	1	–	1	1	–	0.5	1	–	1	1
Q19	**92**	0.5	1	0.5	1	1	1	1	–	1	–	1	–	1	–	–	–	1	–	1	1
Q20	**81**	1	1	–	1	1	0	1	–	1	–	0	–	1	–	–	0.5	1	–	1	1
Q21	**92**	1	1	1	1	1	1	1	–	1	–	1	–	0	–	–	–	1	–	1	1
Q22	**68**	0.5	–	1	0.5	–	1	1	–	1	–	0	–	0.5	–	–	–	1	–	0	1
Q23	**89**	1	1	1	1	1	0.5	0.5	–	1	–	1	–	0.5	–	–	1	1	–	1	1
Score		**16**	**17.5**	**15.5**	**15.5**	**15.5**	**15.5**	**15.5**	**6.5**	**17.5**	**5**	**15.5**	**7**	**13**	**9.5**	**–**	**8**	**19**	**8**	**17**	**17.5**
Score %		**84**	**97**	**86**	**82**	**82**	**78**	**82**	**93**	**92**	**63**	**78**	**78**	**65**	**63**	**–**	**73**	**95**	**80**	**85**	**88**

Table 4
Participants' written explanations for why random forests outperform decision trees.

Why does a random forest perform better than a decision tree?	Participants
Average of many trees	6
Greater amount of data	4
Looks at the problem in many different ways	3
It can take more complex variability into account because it simplifies less	3
Uses decision trees and probability	1
More branches to allow for more fine-grained rules	1

Table 5
Summary of participants' written responses to what constitutes ML.

How would you describe what ML is?	Participants
Inferring information from input data	3
Recognizing patterns	3
Automatic learning of rules on a dataset, applied to new data	3
Machines that try to behave like humans	2
Training a system using data	2
Combines statistical and programming concepts together	1
Aiding decision making by providing data	1
Method of using data to make predictions on unseen data	1
Method of processing data to automate routine tasks	1

really" [P04]. This implies that novices might believe they do not need to grasp the algorithms if they can experiment with them practically. Yet, this poses another risk: novices might unknowingly introduce biases if they assume they can apply ML without comprehending the theoretical foundations. For instance, they might not thoroughly test their model, especially regarding edge cases. Even more worryingly, some participants proposed excluding outliers from the model if they affect performance, even though they may not be anomalies, but rather valid input samples: "*the outliers, you should always exclude them because then that would really mess your dataset*" [P06]. On the other hand, one participant did acknowledge the risk of misclassifying edge cases: "*my concern is that when I come up with a set of algorithms or formulas, what if there are outliers and the machine actually does not know how to detect the alliance?*" [P15].

4.1.2. Reflecting on what constitutes ML

This sub-theme focuses on participants' reflections of ML after completing the tutorial. Participants broadly characterized ML as a system that recognizes patterns in the data and learns rules to make decisions: "*in order to create artificial intelligence, you have to train this machine to see the patterns of the data*" [P06], and "*learning the rules to making some decisions on their own*" [P05]. Participants also described the notion of feeding the model new data to predict an output: "*give it new data that I haven't fed it with yet, for the computer to make decisions based on what it knows, based on the information it has, based on the structures it has built*" [P09]. Table 5 presents the written responses participants provided about what ML is before completing the tutorial, indicating that their understanding of ML remained largely unchanged. To thirteen participants, it was challenging to conceptualize the concepts of models and algorithms, and what constitutes ML. For example, three participants were unsure how classification relates to the definition of ML, whilst five participants were unsure how to even define it, giving vague responses: "*it's using data to make human lives easier. Functions that speed up processes that are long-winded*" [P16]. One participant erroneously thought that ML was able to adapt beyond learning from the data, without human input: "*it can adapt and change itself if it needs too, so it's more like if it finds that it's not working, what it's trying to solve, then it would try to find a way to adapt and change*" [P03]. P15 had the erroneous view that models which incorporate ML will perform better than humans, which is not necessarily the case. As P16 pointed out, this viewpoint might be reasonable in practice, as an ML model is more likely to be employed if the accuracy matches or even exceeds that of a human.

Two participants tried to differentiate between ML algorithms and other types of algorithms: "*With respect to ML you're training a model in order to be as accurate as possible. Whereas a normal algorithm you may not be training a model, it is just to perform routine activities*" [P10]. However, participants were often unsure whether a real-world application uses ML, struggling to differentiate between normal computer programs and ones that use ML: "*I've done some Watson, natural language learning*

stuff like that kind of stuff. That was an API, but is that an AI? Or is it just using the computing power of Watson?" [P01]. Many participants' impression was that ML is more complex than what was shown in the study: "*it's complicated concepts and it explained it in a really clear and easy way, almost too easy, like I was a bit suspicious, like what are they not telling me?*" [P05]. This can be partially attributed to the study's lack of in-depth coverage of the mathematics behind the algorithms. However, it may also stem from the common perception among non-ML experts that ML is challenging (Yang et al., 2020; Amershi et al., 2019). Indeed, most participants believed that ML is primarily designed to tackle complex problems.

The results in this section indicate that while most participants could outline ML in general terms, they struggled with grasping the abstract concepts of models and algorithms. This hindered their ability to understand the capabilities of ML and its most effective applications.

4.2. Applying ML within and beyond the study

This theme encapsulates participants' reflections of applying ML, either in the study or externally. It also describes how participants reflected on the ethical ramifications of applying ML.

4.2.1. Applying ML

During the interview, participants were introduced to a real-world bike sharing scheme dataset. Similarly to the shape dataset introduced at the end of the tutorial, this dataset was large, containing 62,809 unique data samples and 46 features. Participants were asked to describe how they would build their own model on this dataset, applying knowledge learnt in the tutorial. The model's goal was to predict whether a trip will be short, medium, or long. Overall, fifteen participants were able to successfully complete this task by describing a sound model building process, thus demonstrating an ability to apply concepts from the study. A model building process was considered sound if it touched across all of the following points: dataset size and ideas to clean it; selecting relevant features, including ones to avoid; splitting the data into test and train sets; running the training with cross-validation; experimenting with different algorithms; optimizing algorithm parameters; and comparing the evaluation metrics of different algorithms to determine the best model. Only three participants did not complete the task, struggling to apply learnt concepts, whilst two participants were partially able to explain what they would do. As part of the data collection process, two participants suggested that they might want to collect different types of data: "*have you got the right dataset here to make the comparisons with, that's what I would be questioning*" [P16]. Other participants talked about data processing techniques, such as joining the bike sharing scheme dataset comprising of separate tables of bike and weather data on a column and possibly creating new features from the raw features. This shows that participants thought of the whole model building process when discussing the bike sharing scheme, applying concepts they had learnt in the study along with their own reflections on what data is needed for building a good model.

In addition to the bike-sharing dataset exercise, an intriguing aspect that emerged was how participants perceived applying ML and whether their experiences with ML had been influenced by the tutorial. Overall, fourteen participants reported gaining a better understanding of what ML is about: "*It doesn't seem as scary, I guess. It is complicated, but I think the way you've explained it's really clear, so it doesn't seem as complicated, as scarily complicated*" [P14]. Participants were prompted to recall their interactions with ML-driven applications. They cited various examples, including medical diagnosis for cancer, image recognition, and recommender systems like those found in Netflix and Spotify. The study heightened participants' awareness of real-world ML applications, with two participants reflecting on how social media platforms might employ ML to retain users. This indicates that education and training that conceptualize ML's capabilities does change novices' perception of the field. In addition, almost all participants mentioned how, beyond the study, they would want to apply ML to their own work, or how it could be applied. One participant viewed incorporating ML to make their work more time-efficient, allowing them to focus on more engaging tasks like interpreting results [P17]. Six participants discussed how they now want to apply ML to their own domain-specific datasets, discussing potential difficulties: P16 mentioned that for classifying documents, it might be difficult to distinguish between nuances of certain words, or contextual information might be lacking. Other types of applications within participants' own work that could or do implement ML include: intrusion detection in a system; object recognition; driverless cars; predictive forecasting; and re-categorizing mental health diseases. In terms of problem selection, only P20 shows an understanding of where ML is most useful: "*you could use it to automate some mundane repetitive tasks that require a large volume of data*" [P20]. This is contrast to other participants, who thought ML was designed to solve difficult problems.

4.2.2. Ethical ramifications

Most participants spoke of the ethical ramifications of ML, including fairness, accountability and transparency. For example, participants' trust in a model is strongly linked to accuracy: the higher a model's accuracy, the more it was trusted by participants: "*if it is consistently wrong and people would start using it, that may affect their confidence in using the system*" [P20]. Three participants mentioned the importance of explainability from a matter of transparency, as it aims to provide information about how and why an ML algorithm produced its output, suggesting that higher transparency might lead them to place higher trust in the algorithm: "*I certainly think that my faith in such a system would increase if it was possible to see an audit trail around this, if there had been some testing or evaluation over time that demonstrated to me that the algorithm mostly got it right over time, and then there was this constant process of flagging up where it got it wrong*" [P13]. This confirms prior research that found that higher accuracy and transparency both increase trust (Tintarev and Masthoff, 2011). In terms of accountability, an issue brought up by participants is the risk that ML systems are over-promising what they can do. Participants queried what happens when real-world models get things wrong: "*From what I saw [in the tutorial] it looks like the models can be really accurate, but you still you wouldn't want to be that [incorrectly predicted] 1%, would you?*" [P14]. This raises an interesting ethical concern: at what level of accuracy can one deploy a ML model? Along the same lines, P04 suggested that the explainability of algorithms would show users what ML can and cannot do, thus creating better-defined uses and limitations of ML, and avoiding issues of over-promising what ML can do.

Participants showed a sensitivity towards bias, considering ethical considerations when designing a ML system, including questions of accuracy, bias, and confidence. Participants suggested that one way to avoid bias is to use a ML model: "*it kind of makes it more objective as well, because if you have a person and maybe they like someone more but not for the reason they should like someone more*" [P02]. In reality, ML models can still have biases and participants highlighted the importance of trying to avoid introducing biases to the model, such as data bias: "*I think that's the limitation that we need a good at reliable dataset to start with. Because if we introduce biases or mistakes then they might get magnified by the analysis and give a completely wrong answer*" [P17]. Even so, participants mentioned that biases may still be inevitable due to human input in the model building process: "*I think at the end of day, it's being trained in a certain way. There are going to be biases in it and errors that will come up*" [P14]. Through the study participants became aware that datasets can contain proxy features, promoting bias unintentionally. Participants mentioned issues like racial or gender bias that need to be considered in relation to the bike sharing scheme dataset:"*I just wouldn't want to make any gender based assumptions, you know, in terms of looking at bike trips, because I feel like I'd be building in some biases into it or something*" [P01] and "*even if you don't tell it race, there's going to be like other variables that might be connected to race*" [P05]. Eight participants emphasized the significance of having a large dataset with reliable data.

They noted that if this isn't the case, the model might exhibit lower accuracy and could potentially introduce biases: "*we need a good and reliable dataset to start with. Because if we introduce biases or mistakes then they might get magnified by the analysis and give a completely wrong answer*" [P17]. One participant even suggested that ML models should be built to reflect an idealized version of society, rather than promoting underlying biases: "*maybe consult with some liberal social psychologists and see what could help build a future that we like* [P05]. This vision implies counterfactual fairness, which states that outcomes are fair for individuals if they would receive the same outcome if they belonged to a different cohort in a counterfactual world (Kusner et al., 2017)".

4.3. Visualization

Visualizations were integral to the study, with many codes relating to the various visuals in the tutorial. Participants readily interpreted these visualizations, except for the histogram used to describe the naive Bayes algorithm, as three participants had not come across this type of visualization before. In some cases, the availability of a visualization for a specific algorithm, such as the decision tree algorithm, was reported as a reason to choose it for a model [P14]. Scatter plots were used to familiarize participants with the data and get them to think about how ML algorithms might separate the data in a 2-dimensional feature space (an example of this can be seen in Fig. 1). This process helped participants understand that some features are more important than others. Participants also linked the rules to separate classes on a scatter plot to how the decision tree algorithm works: "*I think I put my own set of rules in one of the first questions. Later on, when I saw that with the tree it kind of made sense to me, I kind of linked it. Like it's something I kind of did before that, and I understood why the computer did that*" [P06]. Two participants correctly noted that the scatter plot is useful in visualizing two features at a time (in two dimensions), but the model may use more features than that, a limitation for this kind of visualization. The study incorporated confusion matrices, a popular way of visually identifying false positives and false negatives in the output. An example of a confusion matrix derived from the shape dataset can be seen in Fig. 3. While the visualization effectively conveyed where the algorithm faltered, many participants emphasized that the confusion matrix lacked insights into the underlying reasons behind incorrect predictions and guidance on enhancing the algorithm's performance: "*I think it's definitely useful. However, it doesn't explain why [...] the system got it wrong*" [P17]. Indeed, prior work on intelligibility has found that users want to know *why* an application behaved in a certain way and *how to* modify it (Lim et al., 2009; Lim and Dey, 2010).

One of the tutorial sections introduced the notion of classification boundaries: participants were able to visualize this on a two-dimensional feature space within the graphical tool (an example of this visualization can be seen in Fig. 4). The aim of this visualization was to show participants how the algorithms might have different output boundaries. Overall, participants enjoyed this visualization, referring to it as a way to *"see inside"* the algorithm: "*I like the boundary visualizer as well, because it gives you a sense of how the algorithm actually works*" [P04]. One participant made the link between the boundary visualization and the data points on the scatter plot: "*in the scatter plot there was this one little part that overlapped a bit between the two flowers: that green part (in the boundary visualization) is just a really detailed visualization of that overlapping with the best way to split it up*" [P05]. Similarly to the confusion matrix, the boundary visualization does not provide participants with explanations regarding the type of output, nor does it show users how to action on the output, something that prior work has shown to frustrate users (Lim and Dey, 2011). As a result, the boundary visualization did not help participants understand complex algorithms such as random forests. Unlike the decision tree visualization, participants understood that visualizing the implementation of the random forest model as a collection of trees would be impractical due to its multi-dimensional design, hence they could only see an output of a random forest model through the boundary visualizer. This might explain why eight participants struggled to understand random forests when shown a visualization of its output through the classification boundaries of the Iris dataset: "*even though I can see exactly what's going on, it doesn't make the same sort of sense that the decision tree made to me visually*" [P16]. Participants became aware that visualizing features in a multi-dimensional space is not possible, making it hard for them to conceptualize how multi-dimensional features interact with each other: "*I think it's also because it's all happening in geometric space almost and that is quite hard to sort of visualize, particularly when it's a multi-dimensional geometric space that I find quite hard to comprehend*" [P13].

In summary, although participants found visualizations helpful for explaining concepts, they did not find them particularly helpful in determining how to enhance a model. In addition, understanding multi-dimensional data and algorithms proved to be difficult.

4.4. ML process

A final theme emerged regarding fundamental model-building practices. The following sub-sections outline the findings within this theme.

4.4.1. Feature selection

After loading the Iris dataset and observing the data, the next model-building step for participants was feature selection: "*a good next step would be to identify the relevant features. The ones that are most discriminating. So, you would go through a feature selection phase*" [P19]. Participants were intuitively aware that some features are more useful than others. At a rudimentary level, participants based their feature selection on their own intuition: "*first of all, I would try and eliminate the features that I think are useless from a qualitative or high-level idea*" [P08]. Some participants took a more rigorous approach by plotting pairs of features against each other on a scatter plot. Other participants, when employing Weka's feature selection algorithm, were perplexed as it frequently pinpointed different key features than they had anticipated. However, four participants brought arguments against performing feature selection. Firstly, they did not see feature selection as a step that ML practitioners would undertake, as to them this concept would be abstracted away in an ML algorithm's logic, in that the algorithm does the feature selection instead by assigning different weights to different features. Secondly, some of these participants expressed doubt around the feature selection process: namely which features to keep in the model and why the model sometimes performed better when using all the features, instead of the just the ones recommended by the feature selection algorithm: "*When I'm applying these seven attributes, my random forest algorithm is not working as well as it was working with 30 attributes when no selection was done*" [P07].

4.4.2. Algorithm selection

Most participants had the right approach to algorithm selection in ML: a process of trying different algorithms and comparing the evaluation metrics to decide on the best algorithm. However, algorithm selection was described as a challenge by many participants: "*having to make that decision towards the end of choosing which algorithm are you going to go for, that's where I was a bit, you know, confused what would be best*" [P06]. Participants thought there was more to the process of choosing an algorithm than just trial-and-error and accuracy comparison. They thought that context and knowledge on how the algorithm works dictates why one algorithm might be better suited over another. Participants felt like they did not know enough about algorithm differences to know which one is best suited in which situation. The tutorial does not cover direct comparisons between algorithms. Rather, participants are left to discover this themselves by trying out the different classifiers covered in the study and observing the output both through visualizations and evaluation metrics. In addition, eight participants would have liked a more detailed explanation of the algorithms, with three participants resorting to external resources

for further guidance when they completed the tutorial. These three participants were searching for why something was not working when completing the tutorial [P13], or to get further clarification on ML concepts [P09, P12]. These participants felt that by having a more in-depth understanding, they would be better equipped at selecting which algorithm is most suitable, beyond relying only on accuracy: "*in my mind, there are some reasons as to why you would choose an algorithm over the other one based on how it suits the data itself. I imagine that the accuracy would be good, and it would also be a suitable algorithm for this type of data*" [P03].

4.4.3. Parameter optimization

The only example of parameter optimization covered in the study was for the parameter *k* in kNN. This parameter, defined by the user, instructs the algorithm on how many of the closest training data points' outputs to consider when predicting the output of an unclassified input sample. Participants liked the combination of short explanations followed by the practical nature of parameter selection (i.e., the minimalist explanation model Rosson et al., 1990), helping them develop a mental model of how the kNN algorithm worked. In the exercises, most participants found a value for *k* that optimized the accuracy of kNN, reflecting on the practical nature of the exercise: "*to see what happens when you change the k factor that was very interesting, to understand that actually there is an optimal range, and if your k is too low or too high, you won't get as good results*" [P17]. The visualization, shown in Fig. 5, helped four participants understand the effect of changing the parameter. However, there were seven other participants who were not sure what *k* referred to, thinking that it refers to a fixed distance rather than the number of points that the algorithm should consider.

4.4.4. Model evaluation

In the final stage of the ML process, participants evaluated the model's output. They assessed model performance by considering the percentage of correctly classified instances and by examining the confusion matrix, which was viewed as a visual tool for interpreting model evaluation. During their evaluation of models, participants simply chose the model that provided the highest accuracy, despite their prior reflections on bias risks in Section 4.2.2. In the interview, participants were asked to reflect on the accuracy of the models they created. P12 acknowledged that there might be limits to how high the accuracy can be, whilst P01 suggested a minimum accuracy threshold on their personal opinion: "*maybe the 80/20 rule, you can say that if it's right four times out of five, that is good*" [P01]. Reflecting on the datasets used in the study, eleven participants suggested that the acceptable threshold might depend on the context. For example, they related it to the bike sharing exercise introduced during the interview: a bike hiring scheme does not require high accuracy because it is not perceived as life changing. Finally, P07 mentioned that their confidence in model building was based not only on the accuracy of the results but also on the systematic process of model building, which suggests that a thorough ML process increases trust in what has been built.

5. Discussion

From the findings, we have identified three main discussion points: (1) (mis)understanding ML and what it can do, (2) ML challenges for novices, and (3) broader reflections on ML.

5.1. (Mis)understanding ML and what it can do

Our findings in Sections 4.1.2, 4.2.1, and 4.4.1 show that most participants had exaggerated expectations of what ML could do. Due to this discrepancy in how participants viewed problem formulation in ML, participants questioned whether the content presented in our study can be considered ML, as they found it easier than expected. This finding supports previous research that identified a widespread perception in society that ML is challenging and intended to solve difficult problems (Yang et al., 2020; Amershi et al., 2019; Kozyrkov, 2018), even though ML is best suited for simpler, repetitive applications (Weiner, 2020). In fact, one of the main reasons attributed to why 87% of ML projects never get released (Venture Beat Staff, 2019) is that project managers ask for applications that are too complex (Weiner, 2020).

Our findings in Section 4.1.2 show that although most participants were able to anecdotally define ML as learning a set of rules "to see the patterns of the data", they struggled to grasp the abstract concepts of models and algorithms, often using the terms interchangeably despite their distinct meanings. Not only did participants frequently struggle to distinguish between ML models and other types of algorithms, but they also found it challenging to understand what constitutes ML, and what ML can and cannot do. For example, in Section 4.1.2, participants' expectations were that a ML model would be able to "update itself" and "adapt on their own, without human input", which is not the case in current ML model building practices. This corroborates findings by Sulmont et al. (2019), where ML instructors described higher-level design decisions as the biggest challenge when teaching novices about ML. Previous research has identified strong connections between the robustness of mental models and user understanding (Kulesza et al., 2012, 2013). Our findings, such as those in Sections 4.1 and 4.3, support this, revealing that participants without a recent mathematical background struggled with abstract ML concepts due to their lack of a foundational mental model for such concepts. This finding corroborates prior work by Patel et al. on software engineers that have applied ML in their work (Patel et al., 2008b), which found a lack of understanding of ML concepts among those who are not mathematically trained and extends these prior findings to novices, implying that despite additional experience of working with ML, it remains difficult to use and apply ML without a formal background and training in the field.

A related challenge, highlighted in our findings (e.g., in Sections 4.1.1, 4.1.2 and 4.2.1), is the risk of novices misapplying ML. Some participants who successfully completed the study and found ML easier than expected misinterpreted certain ML concepts. For example, Section 4.1.1 highlighted that instead of wanting to actively include edge cases, or to collect more data around edge cases, some participants wanted to remove anomalies as they can "mess your dataset". In addition, our findings in Section 4.1.2 show participants' misconceptions around the performance of ML models, with novices expecting ML to outperform humans, which is not always the case in practice. These misconceptions and superficial attitudes can lead to the deployment of flawed models. Indeed, prior research found that novice users often deploy problematic models, such as by relying solely on accuracy measures when selecting models (Yang et al., 2018).

In summary, despite most participants having completed the tutorial, they still had misunderstandings on the capabilities of ML that were resilient to the learning, including problem selection, understanding abstract concepts, and the risk of misapplying ML. These misunderstandings around the capabilities of ML and where it can be applied could be problematic if users bring them into the process of building their own models. As suggested by P20 in Section 4.2.1, and extending prior work on ML problem formulation (Weiner, 2020; Kozyrkov, 2018), these challenges imply a need for ML tools and applications to encourage novices to employ ML on simpler problems, such as automating repetitive tasks on large datasets, and to employ simpler algorithms, such as decision trees, as they are inherently explainable. Prior work has shown that simpler algorithms often perform similarly to more complex algorithms on structured data, despite popular belief that more complex models are more accurate (Rudin, 2019).

5.2. ML challenges for novices

Participants experienced difficulties with ML concepts throughout the study, such as algorithm selection (e.g., in Sections 4.1.1 and 4.4.2).

Although participants successfully selected algorithms solely on the metrics, confirming prior research (Yang et al., 2018) on ML experts and extending it to novices, they did, however, question whether algorithm selection also relied on context and knowledge, beyond the use of metrics, demonstrating a broader reflection on ML. The degree to which an ML algorithm should be made transparent has been a subject of interest (Hamilton et al., 2014). Previous studies on lay users suggest that a comprehensive grasp of an algorithm's underlying mechanisms is not essential for effective interaction with models that incorporate such algorithms (Rader and Gray, 2015). Our findings in Sections 4.1.1 and 4.4.2 support such a claim, showing that participants were able to select algorithms for a model, with participants reflecting that they may not need context and algorithmic knowledge if they can just empirically test them. This finding aligns with common ML practices: although the study did not provide an in-depth mathematical understanding of classification algorithms, practitioners often use empirical approaches to algorithm selection when tuning a model (Ali and Smith, 2006), with many IML tools (e.g., Fails and Olsen, 2003; Fiebrink and Cook, 2010; Demšar et al., 2013) abstracting the model building process altogether. We believe this might go against novices' expectations concerning the practice of ML: in other domains there might be a prescribed way of doing things, which may have led to them questioning the potential need for context and further knowledge. For example, in statistical hypothesis testing, there are clear boundaries for when to use a parametric or non-parametric test (Harwell, 1988). In terms of algorithm selection based on the output's context, there have been growing calls to identify ways for lay users to measure model performance beyond accuracy (Veale et al., 2018) and reduce novices' over-reliance of summary statistics such as accuracy (Krause et al., 2016), echoing our participants' desire for context around the output. Our findings (e.g., in Section 4.1.1) also confirm this need, as another pitfall was identified: participants would exclude outliers from the training dataset simply because they lower overall performance. However, when the study presented participants with context around the model output through the confusion matrix, participants found it frustrating that the matrix did not show them *why* the model reached an output or *how to* modify the model to change the output, confirming prior research which found that explanations addressing users' questions to *why* and *how to* supports intelligibility (Lim and Dey, 2010; Lim et al., 2009). This implies a need for novice ML tools to not only provide more context around the output, but to provide actionable explanations which help novices understand why the model reached a certain output and how to improve on this outcome.

Our findings in Section 4.3 show that although participants found visualizations helpful, these were sometimes misinterpreted. For example, participants described the kNN visualization (shown in Fig. 5) as helpful with explaining the algorithm, but our findings in Section 4.4.3 show that some participants misinterpreted the role of the parameter k, thinking that it refers to a fixed distance, rather than a fixed number of points. This finding implies that although there is potential for leveraging visualizations in ML tools for novices, there is a need to design them in ways that avoid multiple interpretations, perhaps using focus techniques (Ajani et al., 2021), which are shown to enhance clarity and improve memory retention, may help in the redesign of this common depiction of kNN (Witten and James, 2013). An alternative reason for the confusion could be linked to the concept of parameters and their optimization process, as participants in the study were tasked with tuning the value of k in kNN. This is corroborated by prior research conducted by Fiebrink et al. (2009), which found that novices were confused by parameter tuning, with the researchers suggesting an abstraction of parameter selection through a slider control from "very fast training" to "very accurate training". However, this would at the expense of a novice user's ability to fine-tune their models, which may frustrate them if they are not able to leverage new information to change the output (Lim and Dey, 2011; Kulesza et al., 2015). Indeed, participants did not find visualizations useful if they could not leverage the information to modify and improve the model, echoing our findings on model output. This finding resonates with prior research on designing explanations, where Tintarev and Masthoff (2011) found that users prefer explanations that describe what might help the system learn faster, and iterates our implication that explanations, whether visual or textual, need to be actionable. The results presented in Sections 4.3 and 4.1.1 indicate that participants who struggled to comprehend the functioning of the random forest due to its high dimensionality also exhibited difficulties understanding its predictions, as observed through the output displayed on the boundary visualizer. This finding corroborates previous research by Oh et al. (2020), which revealed a mismatch between users' expectations and a model's actual output when users lacked understanding of the underlying algorithm. This disconnect persisted even though the algorithm was based on the principles of lower-dimensional decision trees—a concept the participants had grasped, as evidenced by their ability to directly visualize the algorithm. One implication from participants' experiences with random forests is that novices may struggle to understand more complex algorithms simply by building on their foundational knowledge of simpler ones, as the lack of transparency and the difficulty of explaining these algorithms due to their highly dimensional data hinders comprehension (Abdul et al., 2018; Lipton, 2018). This challenges the common approach of using simplified examples in introductory ML material (Witten and James, 2013; Chollet, 2021; Géron, 2022).

Despite the challenges of algorithm selection, parameter selection, and dimensionality, our findings (e.g., from Sections 4.1, 4.2, and 4.4) indicate that it is possible for novices to apply basic ML concepts on their own, as evidenced by thirteen participants completing the study exercises and building a model from scratch on a shape dataset. This aligns with findings by Martins and Von Wangenheim (2023), who discovered that high school students could apply basic ML concepts using active learning strategies, with our study extending this finding to an adult novice population. For instance, our findings (e.g., in Section 4.2.1) indicate that participants, despite being complete novices to ML, could describe a sound model-building process, including experimenting with different algorithms and feature combinations. This contrasts with prior work which found that participants' reasoning about a model's expected output using various examples did not always lead to them developing a clear mental model of the system (Oh et al., 2020), and that even more experienced ML users struggled with understanding and applying iterative exploration processes when creating models (Patel et al., 2008a,b; Amershi et al., 2019). Although primarily used to extract insights, the tutorial highlights that brief, hands-on training can effectively help novices create mental models of ML. This aligns with prior findings showing that a short amount of ML training enabled novices to build models comparable to those of experts (Ramos et al., 2020). The *implication*, then, is that there is a case for training and education about ML to reach a broader audience. Earlier research has advocated for educating children about key ML concepts (Touretzky et al., 2019): our work extends this call to all novices who want to apply ML.

5.3. Broader reflections on ML

Although the study offered a brief introduction to classical ML concepts and applications, participants' written answers to the exercises and subsequent interviews revealed that they considered algorithm selection beyond statistical measures (e.g., in Section 5.2), however, findings in Section 4.4.4 show that they still selected models based on the highest accuracy. Findings in Section 4.2.2 illustrate how participants responded to bias concerns, linking these issues to the input features of the three study datasets (Iris, shapes, and bike sharing) and considering the impact on their own domains. Regarding input features, participants emphasized the importance of avoiding bias by excluding gender or age when building an ML model on the bike-sharing dataset. This aligns with the fairness through unawareness definition, which

posits that a model is fair if it does not use sensitive features (Grgic-Hlaca et al., 2016). By completing the tutorial exercises, participants became aware that biases could arise from using sensitive features in the dataset and could also be introduced by ML practitioners through the way they build and validate their models, such as through human evaluation biases (Srinivasan and Chander, 2021). Participants also discussed the abstract concept of trust. Findings in Sections 4.2.2 and 4.4.4 indicate that participants were more likely to trust models with higher accuracy and better explainability. This confirms prior research showing that users have higher acceptance of results when accuracy is high and accompanied by explanations (Tintarev and Masthoff, 2011), and extends this understanding to novice ML users. However, these discussions included some misconceptions. For instance, Section 4.2.2 found that participants believed trust could be increased by using ML algorithms instead of human judgment, perceiving them as "more objective" based on the tutorial. They also suggested that bias must stem from the underlying data or from practitioners' biases when building the model. This impression is not entirely correct: models can suffer from algorithmic bias (Danks and London, 2017), where the bias is added purely by the algorithm (Baeza-Yates, 2018). This perception among novices that ML models can be trusted poses a challenge for them, confirming prior research which found that novices are more likely to trust models than ML experts (Yang et al., 2018), as it increases the risk of inadvertently developing biased models (Yang et al., 2018).

6. Implications

The findings and discussion have implications for designing ML tools for novices. Prior research in explainable AI suggests that ML tools need to be designed in a way that is understandable to novices and helps them avoid common pitfalls (Abdul et al., 2018). Our findings (e.g., Section 4.1.1) and discussion (e.g., Section 5.2) support such a call, suggesting that novices need ML tools that encourage them to thoroughly test their models on a variety of datasets and subsets of given datasets. For example, it might be helpful to include features that visualize and characterize subsets of data where models fail and examine edge cases. Participants' comments suggest a risk for novices to misapply ML by removing anomalous values or over-relying on the "trial-and-error" testing of ML algorithms offered by such tools. Perhaps a checklist for novices to follow during model-building could reduce the risk of deploying problematic models. Prior work has demonstrated the effectiveness of such a simple job aid in reducing errors due to the limitations of human memory (Hales and Pronovost, 2006), and it could help bridge the gap in novices' awareness of what is needed for a successful ML model.

Further, our findings (e.g., Sections 4.1.1, 4.2.1 and 4.4.1) and discussion (e.g., Section 5.1) suggest that ML tools for novices should encourage the application of simpler models to straightforward, repetitive tasks. Our discussion in Section 5.2 highlights that novices predominantly viewed ML as useful for solving difficult problems, whereas ML can also be effective in automating repetitive tasks (Weiner, 2020; Kozyrkov, 2018). Improving access to ML by designing ML tools that guide novices towards automating simpler, mundane, yet valuable tasks could alleviate their fear of using it. This approach could also enhance novices' success in implementing ML models that meet stakeholders' needs while minimizing the risks of bias and other unwanted consequences (Kozyrkov, 2018; Hume, 2017; Mitchell, 2019).

Our discussion (e.g., Section 5.2) also highlights the need for actionable visual and textual explanations for novices. Tools designed for ML practitioners that provide actionable context around model outputs already exist, with a few examples spanning different parts of the model building process given below. Squares (Ren et al., 2016) is an example of a ML tool for experts that enhances confusion matrices, helping practitioners understand model outputs and compare algorithms. INFUSE (Krause et al., 2014) ranks predictive features across algorithms, folds, and classifiers, enabling domain experts to identify key features for model selection. IForest (Zhao et al., 2018) is a tool for interpreting random forests through visual explanations. However, these visualizations may not be suitable for novices, as they assume prior knowledge of fundamental ML concepts such as cross-validation (Berrar, 2019) and confusion matrices (Susmaga, 2004), unless supplemented by additional training.

In our discussion (e.g., Section 5.2), we emphasize the need for actionable visual and textual explanations tailored for novices. While there are existing tools designed for ML practitioners that provide actionable context around model outputs, a few examples covering different stages of the model-building process are listed below:

1. Squares (Ren et al., 2016) enhances confusion matrices, helping experts understand model outputs and compare algorithms.
2. INFUSE (Krause et al., 2014) ranks predictive features across algorithms, folds, and classifiers, aiding domain experts in identifying important features for model selection.
3. IForest (Zhao et al., 2018) offers visual explanations for interpreting random forests, such as allowing users to trace how individual predictions are made by the forest and displaying the relative importance of features in the decision-making process.

However, although these visualizations aim to simplify ML concepts through intuitive interfaces, they may still be unsuitable for novices without additional training, as they assume prior knowledge of fundamental ML concepts, such as cross-validation (Berrar, 2019) and confusion matrices (Susmaga, 2004).

7. Limitations and future work

It is likely that participants' high level of education made them more capable of understanding the ML content presented in the study. Some participants' familiarity with statistics may have better equipped them to grasp concepts. To broaden the range of individuals represented, alternative recruitment strategies, such as using online research platforms or organizing community workshops, could be implemented to help diversify the participant pool. Even so, ML is clearly a challenging topic that cannot be mastered in a few hours: our findings Sections 5.1 and 5.2 revealed a series of challenges and misconceptions which persisted even after participants completed the take-home tutorial. Additionally, we believe that this participant pool, with its diverse backgrounds and expertise ranging from Electrical Engineering to Social Sciences and Multimedia Design, is representative of individuals who are both willing and realistically capable of applying ML within the context of a two-hour study.

8. Conclusion

This paper presented a qualitative study designed to understand the challenges ML novices encounter when building simple ML models for classifications problems. The study included twenty participants who engaged with fundamental ML concepts through an interactive take-home tutorial, completed various exercises, and were subsequently interviewed. Our findings indicated that although participants reflected on good model building practices, discussing how ML should be applied, they encountered a variety of conceptual challenges, such as interpreting visualizations, problem selection and the multi-dimensionality of both algorithms and data. Finally, based on these findings, we have identified a series of implications for designing effective ML tools for novices, including providing actionable insights and directing novices towards simpler problems. The growing interest in ML by novices raises challenges in developing tools that help them correctly and efficiently apply ML whilst helping them avoid pitfalls. This is a timely challenge for the HCI community, as the misapplication of ML might potentially lead to biased or unfair results.

CRediT authorship contribution statement

Robert Cinca: Writing – original draft, Visualization, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Enrico Costanza:** Writing – review & editing, Validation, Supervision, Resources, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Mirco Musolesi:** Writing – review & editing, Resources, Methodology, Conceptualization. **Muna Alebri:** Writing – review & editing, Resources, Investigation, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/R513143/1 for the University College London Interaction Centre (UCLIC). Our study was approved by the UCLIC Ethics Committee (UCLIC/1617/017).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ijhcs.2024.103438.

Data availability

All data supporting this study is provided as supplementary information accompanying this paper.

References

Abdul, A., Vermeulen, J., Wang, D., Lim, B.Y., Kankanhalli, M., 2018. Trends and trajectories for explainable, accountable and intelligible systems: An hci research agenda. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. CHI '18, ACM, United States, pp. 1–18. http://dx.doi.org/10.1145/3173574.3174156.

Ajani, K., Lee, E., Xiong, C., Knaflic, C.N., Kemper, W., Franconeri, S., 2021. Declutter and focus: Empirically evaluating design guidelines for effective data communication. IEEE Trans. Vis. Comput. Graphics 1. http://dx.doi.org/10.1109/TVCG.2021.3068337.

Ali, S., Smith, K.A., 2006. On learning algorithm selection for classification. Appl. Soft Comput. 6 (2), 119–138. http://dx.doi.org/10.1016/j.asoc.2004.12.002.

Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B., Zimmermann, T., 2019. Software engineering for machine learning: A case study. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). In: ICSE-SEIP '19, IEEE, United States, pp. 291–300. http://dx.doi.org/10.1109/ICSE-SEIP.2019.00042.

Amershi, S., Cakmak, M., Knox, W.B., Kulesza, T., 2014. Power to the people: The role of humans in interactive machine learning. Ai Mag. 35 (4), 105–120.

Amershi, S., Fogarty, J., Kapoor, A., Tan, D., 2011. Effective end-user interaction with machine learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 25, No. 1. pp. 1529–1532. http://dx.doi.org/10.1609/aaai.v25i1.7964.

Angwin, J., Larson, J., Mattu, S., Kirchner, L., 2016. Machine bias. ProPublica 23 (1), 139–159, URL https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

Baeza-Yates, R., 2018. Bias on the web. Commun. ACM 61 (6), 54–61.

Barreno, M., Nelson, B., Joseph, A.D., Tygar, J.D., 2010. The security of machine learning. Mach. Learn. 81, 121–148.

Berrar, D., 2019. Cross-validation. Encycl. Bioinform. Comput. Biol. 1 (1), 542–545. http://dx.doi.org/10.1016/B978-0-12-809633-8.20349-X.

Berthold, M.R., Cebron, N., Dill, F., Gabriel, T.R., Kötter, T., Meinl, T., Ohl, P., Thiel, K., Wiswedel, B., 2009. KNIME-the Konstanz information miner: version 2.0 and beyond. AcM SIGKDD Explor. Newsl. 11 (1), 26–31.

Braun, V., Clarke, V., 2006. Using thematic analysis in psychology. Qual. Res. Psychol. 3 (2), 77–101. http://dx.doi.org/10.1191/1478088706QP063OA.

Carney, M., Webster, B., Alvarado, I., Phillips, K., Howell, N., Griffith, J., Jongejan, J., Pitaru, A., Chen, A., 2020. Teachable machine: Approachable web-based tool for exploring machine learning classification. In: Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems. CHI '20, ACM, United States, pp. 1–8. http://dx.doi.org/10.1145/3334480.3382839.

Carroll, J.M., 1990. The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill. MIT Press, USA.

Carter, S., Nielsen, M., 2017. Using artificial intelligence to augment human intelligence. Distill 2 (12), e9.

Chollet, F., 2021. Deep Learning with Python. Simon and Schuster, United States.

Clarke, V., Braun, V., 2013. Teaching thematic analysis: Overcoming challenges and developing strategies for effective learning. The Psychologist 26 (2), 120–123.

Danks, D., London, A.J., 2017. Algorithmic bias in autonomous systems.. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Vol. 17. IJCAI-17, International Joint Conferences on Artificial Intelligence Organization, Germany, pp. 4691–4697. http://dx.doi.org/10.24963/ijcai.2017/654.

Dasgupta, S., Hill, B.M., 2017. Scratch community blocks: Supporting children as data scientists. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. CHI '17, ACM, United States, pp. 3620–3631. http://dx.doi.org/10.1145/3025453.3025847.

Demšar, J., Curk, T., Erjavec, A., Gorup, Č., Hočevar, T., Milutinovič, M., Možina, M., Polajnar, M., Toplak, M., Starič, A., et al., 2013. Orange: data mining toolbox in Python. J. Mach. Learn. Res. 14 (1), 2349–2353.

Dietterich, T.G., 2000. Ensemble methods in machine learning. In: Proceedings of the First International Workshop on Multiple Classifier Systems. MCS '00, Springer, United States, pp. 1–15, URL https://link.springer.com/chapter/10.1007/3-540-45014-9_1.

Dua, D., Graff, C., 2017. UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences, URL http://archive.ics.uci.edu/ml.

Fails, J.A., Olsen, Jr., D.R., 2003. Interactive machine learning. In: Proceedings of the 8th International Conference on Intelligent User Interfaces. ACM, USA, pp. 39–45.

Fiebrink, R., Cook, P.R., 2010. The wekinator: a system for real-time, interactive machine learning in music. In: Proceedings of the Eleventh International Society for Music Information Retrieval Conference, Vol. 3. ISMIR 2010, Utrecht, International Society for Music Information Retrieval, Citeseer, Canada, 2–1.

Fiebrink, R., Trueman, D., Cook, P.R., 2009. A meta-instrument for interactive, on-the-fly machine learning. In: New Interfaces for Musical Expression. NIME '09, ACM, United States, pp. 280–285. http://dx.doi.org/10.1145/1518701.1519023.

Françoise, J., Caramiaux, B., Sanchez, T., 2021. Marcelle: composing interactive machine learning workflows and interfaces. In: The 34th Annual ACM Symposium on User Interface Software and Technology. ACM, United States, pp. 39–53.

Géron, A., 2022. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, United States.

Grgic-Hlaca, N., Zafar, M.B., Gummadi, K.P., Weller, A., 2016. The case for process fairness in learning: Feature selection for fair decision making. In: NIPS Symposium on Machine Learning and the Law, Vol. 1. ACM, United States, p. 2.

Guillame-Bert, M., Pfeifer, J., Stotz, R., Gustavo Martins, L., Oldacre, A., Becker, J., Cameron, G., 2022. Simple ML for Sheets. Google, URL https://simplemlforsheets.com.

Hales, B.M., Pronovost, P.J., 2006. The checklist—a tool for error management and performance improvement. J. Crit. Care 21 (3), 231–235.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2009. The WEKA data mining software: an update. SIGKDD Explor. Newsl. 11 (1), 10–18. http://dx.doi.org/10.1145/1656274.1656278.

Hamilton, K., Karahalios, K., Sandvig, C., Eslami, M., 2014. A path to understanding the effects of algorithm awareness. In: CHI'14 Extended Abstracts on Human Factors in Computing Systems. ACM, United States, pp. 631–642.

Harwell, M.R., 1988. Choosing between parametric and nonparametric tests. J. Couns. Dev. 67 (1), 35–38. http://dx.doi.org/10.1002/j.1556-6676.1988.tb02007.x.

Heintz, F., Roos, T., 2021. Elements of AI-teaching the basics of AI to everyone in Sweden. In: Proceedings of the 13th International Conference on Education and New Learning Technologies. EDULEARN21, IATED, Spain, pp. 2568–2572.

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: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. CHI '19, ACM, United States, pp. 1–11. http://dx.doi.org/10.1145/3290605.3300645.

Hofmann, M., Klinkenberg, R., 2016. RapidMiner: Data Mining Use Cases and Business Analytics Applications. CRC Press, United States.

Hohman, F., Kahng, M., Pienta, R., Chau, D.H., 2018. Visual analytics in deep learning: An interrogative survey for the next frontiers. IEEE Trans. Vis. Comput. Graphics 25 (8), 2674–2693. http://dx.doi.org/10.1109/TVCG.2018.2843369.

Hume, K., 2017. How to spot a machine learning opportunity, even if you aren'ta data scientist. Harv. Bus. Rev..

Joachims, T., 1998. Text categorization with support vector machines: Learning with many relevant features. In: Proceedings of the 10th European Conference on Machine Learning. ECML '98, Springer, United States, pp. 137–142, URL https://link.springer.com/chapter/10.1007/BFb0026683.

Keller, J.M., Gray, M.R., Givens, J.A., 1985. A fuzzy k-nearest neighbor algorithm. IEEE Trans. Syst. Man Cybern. SMC-15 (4), 580–585. http://dx.doi.org/10.1109/TSMC.1985.6313426.

Keyes, O., 2018. The misgendering machines: Trans/HCI implications of automatic gender recognition. Proc. ACM Hum.-Comput. Interact. 2 (CSCW), 1–22.

Koller, D., Sahami, M., 1997. Hierarchically classifying documents using very few words. In: ICML, Vol. 97. ACM, United States, pp. 170–178.

Kozyrkov, C., 2018. Advice for finding AI use cases. Hacker Noon, URL https://hackernoon.com/imagine-a-drunk-island-advice-for-finding-ai-use-cases-8d47495d4c3f.

Krause, J., Perer, A., Bertini, E., 2014. INFUSE: Interactive feature selection for predictive modeling of high dimensional data. IEEE Trans. Vis. Comput. Graphics 20 (12), 1614–1623. http://dx.doi.org/10.1109/TVCG.2014.2346482.

Krause, J., Perer, A., Ng, K., 2016. Interacting with predictions: Visual inspection of black-box machine learning models. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. CHI '16, ACM, United States, pp. 5686–5697. http://dx.doi.org/10.1145/2858036.2858529.

Kulesza, T., Burnett, M., Wong, W.-K., Stumpf, S., 2015. Principles of explanatory debugging to personalize interactive machine learning. In: Proceedings of the 20th International Conference on Intelligent User Interfaces. ACM, USA, pp. 126–137.

Kulesza, T., Stumpf, S., Burnett, M., Kwan, I., 2012. Tell me more?: the effects of mental model soundness on personalizing an intelligent agent. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '12, ACM, United States, pp. 1–10. http://dx.doi.org/10.1145/2207676.2207678.

Kulesza, T., Stumpf, S., Burnett, M., Yang, S., Kwan, I., Wong, W.-K., 2013. Too much, too little, or just right? Ways explanations impact end users' mental models. In: 2013 IEEE Symposium on Visual Languages and Human Centric Computing. IEEE, USA, pp. 3–10.

Kusner, M., Loftus, J., Russell, C., Silva, R., 2017. Counterfactual fairness. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS '17, Curran Associates Inc., United States, ISBN: 9781510860964, pp. 4069–4079. http://dx.doi.org/10.5555/3294996.3295162.

Lim, B.Y., Dey, A.K., 2010. Toolkit to support intelligibility in context-aware applications. In: Proceedings of the 12th ACM International Conference on Ubiquitous Computing. Ubicomp '10, ACM, United States, pp. 13–22. http://dx.doi.org/10.1145/1864349.1864353.

Lim, B.Y., Dey, A.K., 2011. Design of an intelligible mobile context-aware application. In: Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services. MobileHCI '11, ACM, United States, pp. 157–166. http://dx.doi.org/10.1145/2037373.2037399.

Lim, B.Y., Dey, A.K., Avrahami, D., 2009. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '09, ACM, United States, pp. 2119–2128. http://dx.doi.org/10.1145/1518701.1519023.

Lipton, Z.C., 2018. The mythos of model interpretability. Queue 16 (3), 31–57. http://dx.doi.org/10.1145/3236386.3241340.

Liu, S., Maljovec, D., Wang, B., Bremer, P.-T., Pascucci, V., 2016. Visualizing high-dimensional data: Advances in the past decade. IEEE Trans. Vis. Comput. Graphics 23 (3), 1249–1268. http://dx.doi.org/10.1109/TVCG.2016.2640960.

Madsen, A., 2019. Visualizing memorization in RNNs. Distill 4 (3), e16.

Martins, R.M., Von Wangenheim, C.G., 2023. Findings on teaching machine learning in high school: A ten-year systematic literature review. Inform. Educ. 22 (3), 421.

Microsoft, 2020. Machine Learning Made Easy. Microsoft, URL https://www.lobe.ai.

Mitchell, M., 2019. Artificial Intelligence: A Guide for Thinking Humans. Penguin, United Kingdom.

Noble, S.U., 2018. Algorithms of Oppression: How Search Engines Reinforce Racism. NYU Press, United States.

Oh, C., Kim, S., Choi, J., Eun, J., Kim, S., Kim, J., Lee, J., Suh, B., 2020. Understanding how people reason about aesthetic evaluations of artificial intelligence. In: Proceedings of the 2020 ACM Designing Interactive Systems Conference. DIS '20, ACM, United States, pp. 1169–1181. http://dx.doi.org/10.1145/3357236.3395430.

Pal, M., 2005. Random forest classifier for remote sensing classification. Int. J. Remote Sens. 26 (1), 217–222. http://dx.doi.org/10.1080/01431160412331269698.

Patel, K., Fogarty, J., Landay, J.A., Harrison, B.L., 2008a. Examining difficulties software developers encounter in the adoption of statistical machine learning. In: Proceedings of the 23rd AAAI Conference on Artificial Intelligence. DIS '17, AAAI, United States, pp. 1563–1566, URL https://www.aaai.org/Papers/AAAI/2008/AAAI08-263.pdf.

Patel, K., Fogarty, J., Landay, J.A., Harrison, B., 2008b. Investigating statistical machine learning as a tool for software development. In: Proceedings of the 2008 SIGCHI Conference on Human Factors in Computing Systems. CHI '08, ACM, United States, pp. 667–676. http://dx.doi.org/10.1145/1357054.1357160.

Rader, E., Gray, R., 2015. Understanding user beliefs about algorithmic curation in the facebook news feed. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. CHI '15, ACM, United States, pp. 173–182. http://dx.doi.org/10.1145/2702123.2702174.

Ramos, G., Meek, C., Simard, P., Suh, J., Ghorashi, S., 2020. Interactive machine teaching: a human-centered approach to building machine-learned models. Hum.-Comput. Interact. 35 (5–6), 413–451.

Refaeilzadeh, P., Tang, L., Liu, H., 2009. Cross-validation. Encycl. Database Syst. 5, 532–538.

Ren, D., Amershi, S., Lee, B., Suh, J., Williams, J.D., 2016. Squares: Supporting interactive performance analysis for multiclass classifiers. IEEE Trans. Vis. Comput. Graphics 23 (1), 61–70. http://dx.doi.org/10.1109/TVCG.2016.2598828.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., Kafai, Y., 2009. Scratch: Programming for all. Commun. ACM 52 (11), 60–67. http://dx.doi.org/10.1145/1592761.1592779.

Retzlaff, C.O., Angerschmid, A., Saranti, A., Schneeberger, D., Roettger, R., Mueller, H., Holzinger, A., 2024. Post-hoc vs ante-hoc explanations: xAI design guidelines for data scientists. Cogn. Syst. Res. 86, 101243.

Rojas, J.A., Kery, M.B., Rosenthal, S., Dey, A., 2017. Sampling techniques to improve big data exploration. In: 2017 IEEE 7th Symposium on Large Data Analysis and Visualization. LDAV '17, IEEE, United States, pp. 26–35. http://dx.doi.org/10.1109/LDAV.2017.8231848.

Rosson, M.B., Carrol, J.M., Bellamy, R.K., 1990. Smalltalk scaffolding: a case study of minimalist instruction. In: Proceedings of the 1990 CHI Conference on Human Factors in Computing Systems. CHI '90, ACM, United States, pp. 423–430. http://dx.doi.org/10.1145/3173574.3174156.

Rudin, C., 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat. Mach. Intell. 1 (5), 206–215.

Sanchez, T., Caramiaux, B., Françoise, J., Bevilacqua, F., Mackay, W.E., 2021. How do people train a machine? Strategies and (mis) understandings. Proc. ACM Hum.-Comput. Interact. 5 (CSCW1), 1–26. http://dx.doi.org/10.1145/3449236.

Schank, R.C., Berman, T.R., Macpherson, K.A., 1999. Learning by doing. In: Instructional-Design Theories and Models: A New Paradigm of Instructional Theory, vol. 2, (no. 2), pp. 161–181.

Srinivasan, R., Chander, A., 2021. Biases in AI systems: A survey for practitioners. Queue 19 (2), 45–64.

Srividya, M., Mohanavalli, S., Bhalaji, N., 2018. Behavioral modeling for mental health using machine learning algorithms. J. Med. Syst. 42, 1–12.

Sulmont, E., Patitsas, E., Cooperstock, J.R., 2019. What is hard about teaching machine learning to non-majors? Insights from classifying instructors' learning goals. ACM Trans. Comput. Educ. (TOCE) 19 (4), 1–16.

Susmaga, R., 2004. Confusion matrix visualization. In: Intelligent Information Processing and Web Mining. Springer, United States, pp. 107–116.

Tintarev, N., Masthoff, J., 2011. Designing and evaluating explanations for recommender systems. In: Recommender Systems Handbook. Springer, United States, pp. 479–510.

Touretzky, D., Gardner-McCune, C., Martin, F., Seehorn, D., 2019. Envisioning AI for K-12: What should every child know about AI? In: Proceedings of the AAAI Conference on Artificial Intelligence. PKP, Canada, pp. 9795–9799. http://dx.doi.org/10.1609/aaai.v33i01.33019795.

Vartiainen, H., Tedre, M., Valtonen, T., 2020. Learning machine learning with very young children: Who is teaching whom? Int. J. Child-Comput. Interact. 25 (2), 1–11. http://dx.doi.org/10.1016/j.ijcci.2020.100182.

Veale, M., Van Kleek, M., Binns, R., 2018. Fairness and accountability design needs for algorithmic support in high-stakes public sector decision-making. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. CHI '18, ACM, United States, pp. 1–14. http://dx.doi.org/10.1145/3173574.3174014.

Venture Beat Staff, 2019. Why do 87% of data science projects never make it into production. URL: https://venturebeat.com/2019/07/19/why-do-87-of-data-scienceprojects-never-make-it-into-production.

Wang, Z.J., Turko, R., Shaikh, O., Park, H., Das, N., Hohman, F., Kahng, M., Chau, D.H.P., 2020. CNN explainer: learning convolutional neural networks with interactive visualization. IEEE Trans. Vis. Comput. Graphics 27 (2), 1396–1406.

Weiner, J., 2020. Why AI/data science projects fail: how to avoid project pitfalls. Synth. Lect. Comput. Anal. 1 (1), i–77.

Witten, D., James, G., 2013. An Introduction to Statistical Learning with Applications in R. Springer, United States.

Yang, F.-J., 2018. An implementation of naive bayes classifier. In: 2018 International Conference on Computational Science and Computational Intelligence. CSCI, IEEE, USA, pp. 301–306.

Yang, Q., Steinfeld, A., Rosé, C., Zimmerman, J., 2020. Re-examining whether, why, and how human-AI interaction is uniquely difficult to design. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. CHI '20, ACM, United States, pp. 1–13. http://dx.doi.org/10.1145/3313831.3376301.

Yang, Q., Suh, J., Chen, N.-C., Ramos, G., 2018. Grounding interactive machine learning tool design in how non-experts actually build models. In: Proceedings of the 2018 Designing Interactive Systems Conference. DIS '18, ACM, United States, pp. 573–584. http://dx.doi.org/10.1145/3196709.3196729.

Zhao, X., Wu, Y., Lee, D.L., Cui, W., 2018. iforest: Interpreting random forests via visual analytics. IEEE Trans. Vis. Comput. Graphics 25 (1), 407–416.