Supporting End-User Understanding of Classification Errors

Emma Beauxis-Aussalet
CWI - Utrecht University
emalb@cwi.nl

Joost van Doorn
CWI - UvA
joost.van.doorn@gmail.com

Lynda Hardman
CWI - Utrecht University
lynda.hardman@cwi.nl

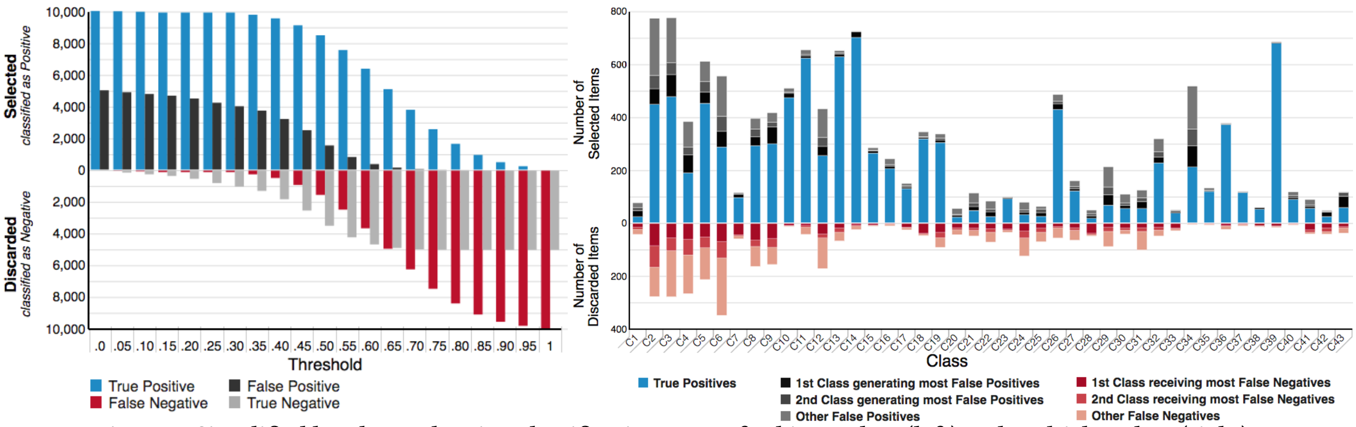

Figure 1: Simplified barcharts showing classification errors for binary data (left) and multiclass data (right).

ABSTRACT

Classifiers are applied in many domains where classification errors have significant implications. However, end-users may not always understand the errors and their impact, as error visualizations are typically designed for experts and for improving classifiers. We discuss a visualization design that addresses the specific needs of classifiers' end-users. We evaluate this design with users from three levels of expertise, and compare it with ROC curves and confusion matrices. We identify key difficulties with understanding the classification errors, and how visualization designs addressed or aggravated them. The main issues concerned confusions of the actual and predicted classes (e.g., confusion of False Positives and False Negatives). The machine learning terminology, complexity of ROC curves, and symmetry of confusion matrices aggravated the confusions. The end-user-oriented visualization reduced the difficulties by using several visual features to clarify the actual and predicted classes, and more tangible metrics and representation. Our results contribute to supporting end-users' understanding of classification errors, and informed decisions when choosing or tuning classifiers.

ECCE'18, September 5–7, 2018, Utrecht, Netherlands

ACM ISBN 978-1-4503-6449-2/18/09…$15.00
https://doi.org/10.1145/3232078.3232096

CCS CONCEPTS

• **Visualization → Empirical studies in visualization**; • **Machine learning → Classification**;

ACM Reference Format:
Emma Beauxis-Aussalet, Joost van Doorn, and Lynda Hardman. 2018. Supporting End-User Understanding of Classification Errors. In *ECCE'18: Proceedings of the 36th European Conference on Cognitive Ergonomics (ECCE2018), September 5–7, 2018, Utrecht, Netherlands.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3232078.3232096

1 INTRODUCTION

Classifiers are inherently imperfect but their errors are accepted in a wide range of applications. End-users may not fully understand the errors and their implications [21] and may mistrust or misuse classifiers [23]. Error assessment is not self-evident for end-users with no machine learning expertise. Yet they may need to understand the classification errors, e.g., to make fully-informed decisions when choosing between classifiers. End-users may also need to control the tuning parameters that can adjust the errors, e.g., to limit the errors for the most important classes. Although machine learning experts better understand the complexity of the algorithms and their parameters, end-users should take part in the final tuning decisions because they better understand the implications of errors for their application domain.

We aim at enabling end-users to choose among classifiers and tuning parameters, and to understand the errors to expect when applying classifiers (e.g., classes may be over- or under-estimated [3, 7]). Choosing and tuning classifiers allow to adjust the errors to specific use cases, e.g., to balance False Positives (FP) and False Negatives (FN, Table 1). For example, when detecting medical conditions, FN are critical (pathologies must not be missed) and FP to a lesser extent (although further procedures may be risky). Pre-defined tuning parameters may not fully address end-user needs. For example, parameters may minimize both FP and FN while users prefer to increase the FP if it reduces the FN. Cost functions can handle user requirements [9] but they are complex and weighing the cost of errors is not always straightforward (e.g., what is the cost of missed pathologies?). The metrics and visualizations of classification errors are also complex and may be misinterpreted by non-experts [21] as their underlying concepts are not common knowledge and do not easily convey the implications in end-usage applications.

A simplified barchart visualization [2] has been designed to address the needs of end-users with no expertise in machine learning (Fig. 1). We analyse the user needs it addresses (Section 3) and discuss its design rationale (Section 4). We then evaluate it compared to ROC curves and confusion matrices (Section 5). The suitability for specific audiences was assessed with users having three kinds of expertise: machine learning; mathematics but not machine learning (as it may impact the understanding of error rates and ROC curves); none of machine learning, mathematics or computer science. We identified key factors that facilitated user understanding or added confusion (Section 6).

The main issues concerned confusions between the *actual* class and the *predicted* class assigned by the classifier (e.g., confusing FN and FP), misinterpretations of error rates and technical terms, and misunderstandings of the impacts of errors on end-results. The simplified visualizations facilitated user understanding by using simpler error metrics, and by distinguishing the *actual* and *predicted* classes with several visual features. Our findings contribute to understanding *"how (or whether) uncertainty visualization aids / hinders [...] reasoning"* about the implications of classification errors, and *"decisions"* when choosing or tuning classifiers [20].

Table 1: Definition of FP, FN, TP, TN.

False Positives (FP): objects classified as *Positive* (e.g., as the primary class to detect) while actually being *Negative* (e.g., the class to discard).
False Negatives (FN): objects classified as *Negative* while being *Positive*.
True Positives (TP): objects correctly classified as *Positive*.
True Negatives (TN): objects correctly classified as *Negative*.

$$\text{Error rates w.r.t. actual class size (e.g., ROC curves):} \quad \frac{n_{xy}}{n_{x.}} \tag{1}$$

$$\text{Error rates w.r.t. predicted class size (e.g., Precision):} \quad \frac{n_{xy}}{n_{.y}} \tag{2}$$

$$\text{Accuracy:} \quad \frac{\sum_x n_{xx}}{n_{..}} \qquad \text{e.g., for binary data:} \quad \frac{TP+TN}{TP+TN+FP+FN} \tag{3}$$

2 RELATED WORK

Recent work developed visualizations to improve classification models [10, 17, 19], e.g., using barcharts [1, 24]. They are algorithm-specific (e.g., applicable only to probabilistic classifiers or decision trees) but end-users may need to compare classifiers based on different algorithms. These comparisons are easier with algorithm-agnostic visualizations using the same representations for all algorithms, and limiting complex and unneeded information on the algorithms. Confusion matrices, ROC curves and Precision-Recall curves are well-established algorithm-agnostic visualizations [11] but they are intended for machine learning experts and simplifications may be needed for non-experts (e.g., understanding ROC curve's error rates may be difficult, especially for multiclass data). Cost curves [9] are algorithm-agnostic and investigate specific end-usage conditions (e.g., class proportions, costs of errors) but they are also complex, intended for experts, and do not address multiclass data. The non-expert-oriented visualizations in [16, 21] use simpler trees, grids, Sankey or Euler diagrams, but are illegible with multiclass data due to multiple overlapping areas or branches.

Different error metrics have been developed and their properties address different requirements [14, 25, 26]. Error metrics are usually derived from the same underlying data: numbers of correct and incorrect classifications encoded in confusion matrices, and measured with a *test set* (a data sample for which the actual class is known). These raw numbers provide simple yet complete metrics. They are easy to interpret (no formula involved) and address most requirements for reliable and interpretable metrics, e.g., they do not conceal the impact of class proportions on error balance, and have known values for *perfect, pervert* (always wrong) and *random* classifiers [25]. These values depend on the class sizes in the test set, which is not recommended in [25]. However, raw numbers convey the class sizes, omitted in rates, but needed to assess the class imbalance and statistical significance of error measurements. These are crucial for extrapolating the errors to expect in end-usage applications [3, 7].

Using raw numbers of errors, we focus on conveying basic error rates in equations (1)-(2) where n_{xy} is the number of objects actually belonging to class x and classified as class y (i.e., errors if $x \neq y$), $n_{x.}$ is the number of objects actually belonging to class x (actual class size), and $n_{.y}$ is the number of objects classified as class y (predicted class size). Accuracy is a widely used metric summarizing errors over all classes, as shown in (3) where n_{xx} is the number of objects correctly classified as class x, and $n_{..}$ is the total number of objects for all classes. We also consider conveying accuracy, and focus on overcoming its bias towards large classes and missing information on the error composition [14] (e.g., high accuracy can conceal significant errors for specific classes).

Table 2: Relationships between users, tasks, information needs, metrics and basic visualizations

		Task			Visualization		
		Improve Model & Algorithm	Tune Classifier	Extrapolate Errors in End-Results	Confusion Matrix	ROC/PR Curve	Classee
Target Audience	End-Users		X	X			X
	Software Providers	X	X		X	X	X
Low-Level Metric	Raw Numbers	X	X	X	X		X
	Error Rates in Equation (1)	X	X	X		X	X
	Error Rates in Equation (2)	X	X	if class proportions are equal [3]		X	X
	Accuracy	X	X				X
	AUC	X				X	variant
High-Level Information	Overall Error Magnitude	X	X		X	X	X
	Errors over Tuning Param.	X	X			X	X
	Errors over Object Features	X		used in extrapolation method [6]			if ≠ x-axis
	Error Composition	X	X	X	X	X	X
	Class Proportions		X	X	X		X
	Class Sizes		X	X	X		X

3 USER INFORMATION NEEDS

We identified key information needs through interviews of machine learning experts and end-users, conducted within the Fish4Knowledge and Classee projects [5, 12]. We found that the needs of software providers and end-users have key differences and overlaps (Table 2). Software providers often seek to optimise classifiers on all classes and all types of error (e.g., FP and FN). For example, they measure the Area Under the Curve (AUC) [11] to summarise all types of errors (FN and FP) over all possible values of a tuning parameter. This approach is irrelevant for end-users who apply classifiers tuned with fixed parameter values. Metrics that summarize all types of errors for all classes (e.g., AUC, Accuracy) fail to convey *"the circumstances under which one classifier outperforms another"* [9], e.g., for which classes, class proportions (e.g., rare or large classes), error composition (i.e., the breakdown of errors between all possible classes) and values of the tuning parameters. These characteristics are crucial for end-users: specific classes and types of errors can be more important than others; class proportions may vary in end-usage datasets; and optimal tuning parameters depend on the classes and errors of interest, and on the potential class proportions. End-users are also interested in extrapolating the errors in their end-usage datasets (e.g., within the objects classified as class Y how many truly belong to class X?). Such extrapolation depends on class sizes, class proportions and error composition [3, 7] and can be refined depending on the features of classified objects [6].

4 CLASSEE VISUALIZATION

The Classee project simplified the visualization of classification errors by using ordinary histograms and raw numbers of errors (Fig. 1). The *actual* class and the error types are differentiated with color codes: vivid colors if the *actual* class is positive (blue for TP, red for FN), desaturated colors if the *actual* class is negative (grey for TN, black for FP). The zero line distinguishes the *predicted* class (TP and FP are above the zero line, FN and TN are below).

For binary data (Fig. 1 left), objects from the same actual class are stacked in distinct bars: TP on FN for the positive class, and FP on TN for the negative class. Basic error rates can easily be interpreted visually. ROC curve's error rates in equation (1) are visualized by comparing the blocks within continuous bars: blue/red blocks for TP rate, black/grey blocks for FP rate. Precision-like rates in equation (2) are visualized by comparing adjacent blocks on each side of the zero line: blue/black blocks for Precision, red/grey blocks for False Omission Rate. Accuracy (3) can be interpreted by comparing blue and grey blocks against red and black blocks, which is more complex. However, it overcomes key issues with accuracy [14] by showing the error balance between FP and FN, and potential imbalance between large and small classes. The visualization also renders information similar to Area Under the Curve [11] as blue, red, black and grey areas can be perceived.

Perceiving ROC-like rates (1) implies comparing *divided* and *adjacent* blocks. It can lower perception accuracy [27] compared to unadjacent blocks in [24] (TP rates rendered with separated TP and FN blocks) or [1] (FP rates rendered with separated TN and FP blocks). However, Classee shows *part-to-whole* ratios while [27] researched *part-to-part* ratios, and suggests that perceiving *part-to-whole* is more intuitive and effective. Further, Classee lets users compare the positions of bar extremities to the zero line, and perceiving positions is more accurate than perceiving relative bar lengths [8]. Precision-like rates (2) are perceived using *aligned* and *adjacent* blocks. It supports more accurate perceptions [8, 27] compared to divided unadjacent blocks in [1, 24].

For multiclass data (Fig. 1 right), errors are shown for each class in a one-vs-all reduction, i.e., considering one class as the positive class and all other classes as the negative class, and so for all classes (e.g., for class x, FP = $\sum_{y \neq x} n_{yx}$ and TN = $\sum_{y \neq x} \sum_{z \neq x} n_{yz}$). TN are not displayed because they are typically of far greater magnitude, especially with large numbers of classes, which can reduce other bar sizes to illegibility. TN are also misleading as they do not distinguish correct and incorrect classifications (e.g., n_{zz} and $n_{yz, y \neq z}$). Without TN, FP are stacked on TP which shows the Precision for each class.

Compared to [24] stacking TP-FP-FN in this order, Classee stacking facilitates the interpretation of TP rates (1) and true class sizes by showing continuous blocks for TP and FN.

Compared to chord diagrams in [1] encoding error magnitudes with surface sizes, Classee uses bar length to support more accurate perceptions of quantities [8].

Accuracy can be interpreted by comparing all blue blocks against either all red blocks, or all black blocks (the sum of errors for all red blocks is the same for all black blocks, as each misclassified object is a FP for its predicted class and a FN for its actual class). Users can visualize the relative proportions of correct and incorrect classifications, but the exact equation of accuracy (3) is harder to interpret. Instead Classee focuses on conveying the error composition for each class while accuracy involves TN that do not distinguish errors from correct classifications [14].

Inspecting the error composition is crucial for understanding the impact of errors in end-results. Users need to assess the errors between specific classes and their *directionality* (i.e., errors *from* an actual class are misclassified *into* a predicted class). If errors between two classes are of significant magnitudes, it creates biases in the end-results [3, 7]. For example, errors from large classes can result in FP of significant magnitude for small classes that are thus over-estimated. Such biases can be critical for end-users' applications.

Hence Classee visualization details the error composition between actual and predicted classes. The FP blocks are split in sub-blocks representing objects from the same actual class. The FN blocks are also split in sub-blocks representing objects classified into the same predicted class. To avoid showing too many unreadable sub-blocks, Classee shows the 2 main sources of errors in distinct sub-blocks and merges the remaining errors in the same sub-block. The FP sub-blocks show the 2 classes from which most FP actually belong, and the remaining FP as a 3rd sub-block. The FN sub-blocks show the 2 classes into which most FN are classified, and the remaining FN as a 3rd sub-block. Future implementations could let users control the number of sub-blocks to display, and the *boxes* in [24] may improve their rendering.

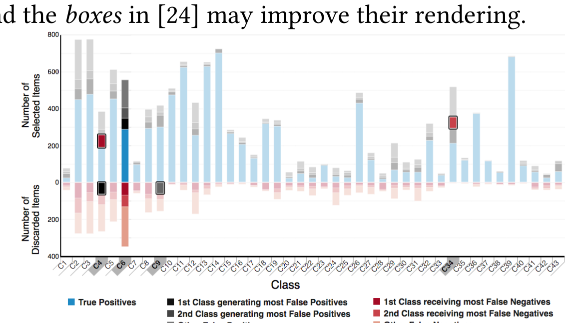

Figure 2: Rollover detailing the errors for a specific class.

Users can select a class to inspect its errors (Fig. 2). It shows which classes receive the FN and generate the FP. The FN sub-blocks of the selected class are highlighted within the FP sub-blocks of their predicted class. The FP sub-blocks are highlighted within the FN sub-blocks of their predicted class. Users can identify the error *directionality*, i.e., they can differentiate *Class X objects misclassified into Class Y* and *Class Y objects misclassified into Class X* (e.g., in Fig. 2, objects from class C6 are misclassified into C34, but not from C34 into C6). Future implementations could also highlight the remaining FN and FP merged in the 3rd sub-blocks.

Large classes (with long bars) can hinder the perception of smaller classes (with small bars). Thus we propose a normalised view that balances the visual space of each class (Fig. 3). Errors are normalised on the TP of their actual class as n_{xy}/n_{xx} (i.e., dividing ${}^{FN}/_{TP}$ and reconstructing the FP blocks using the normalised errors ${}^{FN}/_{TP}$). Although unusual, this approach aligns all FP and FN blocks to support easy and accurate visual perception [8, 27]. It also reminds users of the impact of varying class proportions: the magnitude of errors change between normalised and regular views, as they would change if class proportions differ between test datasets (from which errors were measured) and end-usage datasets (to which classifiers are applied). It is also the basis of the Ratio-to-TP method that extrapolates errors in end-usage applications [3].

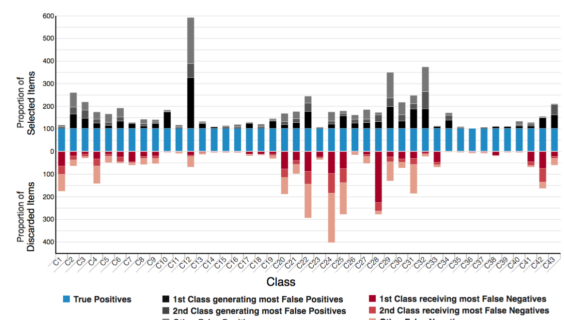

Figure 3: Normalized view with errors proportional to TP

Color choices - Classee uses blue rather than green as in [1] to address colorblindness [28] while maintaining a high contrast opposing warm and cold colors. Compared to class-specific colors in [24] which can clutters the visualization to illegibility (e.g., with more than 7 classes [22]), Classee colors can handle large numbers of classes. Following the *Few Hues, Many Values* design pattern [28], sub-blocks of FN and FP use the same shades of red and black. The shades of grey for FP may conflict with the grey used for TN in binary classification. The multiclass barchart does not display TN and its shades of grey remain darker. Thus color consistency issues are limited, and we deemed that Classee colors are a better tradeoff than adding a color for FP (e.g., yellow in [1]). As a result, the identification of *actual* and *predicted* classes is reinforced by the interplay of three visual features: position (below or above the zero line for the predicted class, left or right bar for the actual class), color hues (blue/red if the actual class is positive), and color (de)saturation (black/grey if the actual class is negative).

5 USER EXPERIMENT

We evaluated Classee and investigated the factors supporting or impeding the understanding of classification errors. We conducted in-situ semi-structured interviews with a think-aloud protocol to observe users' *"activity patterns"* and *"isolate important factors in the analysis process"* [18]. We focused on evaluating the *Visual Data Analysis and Reasoning* rather than *User Performance* [18] as our primary goal is to ensure a correct understanding of classification errors and their implications. We conducted a qualitative study that informs the design of end-user-oriented visualization, and is preparatory to potential quantitative studies. We included a user group of mathematicians to investigate how mathematical thinking impacts the understanding of ROC curves and error metrics. Such prior knowledge is a component of the *Demographic Complexity* interacting with the *Data Complexity*, and thus impacting user cognitive load [15]. This section sumarizes the study setup. The exact tasks, tutorials, datasets and visualizations are specified in [4].

The 3 user groups represented three types of expertise: 1) practitioners of machine learning (4 developers, 2 researchers), 2) practitioners of mathematics but not machine learning (5 researchers, 1 medical doctor), and 3) practitioners of neither machine learning, mathematics nor computer science (including 1 researcher). A total of 18 users and 2 users per condition (3 groups x 3 visualizations x 2 users) was sufficient to yield significant observations, as we repeatedly identified key factors impacting user understanding.

The 3 experimental visualizations compared Classee to ROC curves and confusion matrices, applied to the same datasets. ROC curves are preferred to Precision-Recall curves which exclude TN thus do not convey the same information as Classee. Users interacted with only one kind of visualization, to account for their learning curve. After interacting with a first visualization, users gain expertise that would bias the results with a second visualization.

The 15 user tasks were in two parts, for binary and multiclass data. Each part started with a tutorial explaining the visualization and the technical concepts. This could be displayed anytime during the tasks. The explanations of the technical concepts were the same for all users and visualizations. Only the explanations of the visualization differed.

6 QUALITATIVE ANALYSIS

To identify the factors influencing user understanding of classification errors, we analysed user comments and behaviours by transcribing written notes of the interviews. To let the factors emerge from our observations, we first proceeded with *grounded* coding (no predefined codes). We then organized our insights into themes and proceeded to *a priori* coding (predefined codes). We identified 3 key difficulties that are independent of the visualizations: 1) The terminology (e.g., TP, FN, FP, TN are confusing terms); 2) The error directionality (e.g., considering both FN and FP); 3) The extrapolation of error impact on end-usage application (e.g., a class may be over-estimated). We report these difficulties and how the visualizations aggravated or addressed them.

Terminology - The basic terms TP, FN, FP, TN were difficult to understand and remember (*"In 30 minutes I'll have completely forgotten"*). Twelve users (66%) mentioned difficulties with these terms, including machine learning experts. The terms *Positive/Negative* were often misunderstood as the actual class (instead of the predicted class) especially when not matching their applied meaning (*"Cancer is the positive class, that's difficult semantically"*). Users were also confused by the unusual syntax (*"Positive and Negative are usually adjectives but here they are nouns, it's confusing"*) and the association of antonyms (e.g., False and Positive in FP, *"False is for something negative"*) and synonyms (e.g., *"The words are so close"* with True and Positive in TP, *"I understand that FN are not errors"* because Negative and False is a logical association). Users misinterpreted the terms *True* and *False* as representing the actual or predicted class, and both are incorrect. Some users suggested adverbs to avoid such confusion (*"Falsely"*, *"Wrongly"*). To cope with the semantic issues, users translated the technical terms into more tangible terms, using concrete examples (*"Falsely Discarded"*, *"False face"*). A machine learning expert requested short acronyms (e.g., *TP* for *True Positive*). A non-expert suggested icons as another form of abbreviation (*"like a smiley"* Fig. 4). This user preferred labels mentioning the actual class first (using *Negative/Positive*) then the errors (using *True/False*).

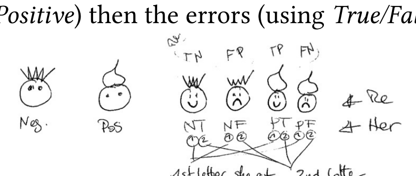

Figure 4: User-suggested icons for TP, FN, FP, TN. ***Drawn by the interviewer following user's instructions in post-experiment discussions. User-suggested labels are below the icons. Usual labels were later added above.***

The terminology of legends and explanations can yield difficulties (*"You could make the text more clear"*). The terms *Select* and *Discard* in our tutorials and legends can be at odds with their application (*"Discarding objects may be confusing if both classes are equally important"*). The term *true* in its common meaning ("*true* class", "*truly* belong to [class x]") conflicts with its meaning in TP, TN and must be avoided.

Math experts were often familiar with TP, FN, FP, TN as these are involved in statistical hypothesis testing. Machine learning experts knew the technical terms well, except a self-taught practitioner who was only familiar to terms used in daily tasks, e.g., *Accuracy* but not *ROC Curve* or *Confusion Matrix*. This user mentioned *"Clients only ask for accuracy"* but did not recall its formula. Two other machine learning experts were unfamiliar with either Precision-Recall or ROC curves, as their daily tasks involved only one of these. Hence machine learning practitioners may not recall the meaning and formula of unused metrics, or even metrics used regularly. Some metrics are not part of their routines, but may be relevant for specific use cases or end-users. Hence experts too can benefit from Classee since i) remembering error rate formulae is not needed as rates are visually reconstructed; ii) both ROC-like or Precision-like rates can be visualized (1)-(2); and iii) accuracy can also be interpreted.

Error Directionality - Users need to distinguish the actual and predicted classes of errors, and the direction of errors *from* an actual class classified *into* a predicted class. Ten users (56%) from all profiles had difficulties with error directions, e.g., confusing FP and FN (*"Oh my FP were FN, why did I switch!"*). With binary data, users may not understand how the tuning parameter influence errors in both directions, e.g., decreasing FN but increasing FP (*"I put a high threshold so that there's no error [FP, FN] in the results"*, *"High threshold means high TP and TN"*). With multiclass data, users may not understand that FN for one class are FP for another, and that errors for class x concern both errors with predicted class x and actual class x (e.g., not considering both FN and FP).

Terminology issues complicated user understanding of error directionality, e.g., the terms *Positive/Negative* could mean both the actual or predicted class. Some users intuitively interpreted these terms as the predicted class, others as the actual class. Users often used metaphors and more tangible terms to clarify the error directionality (*"The destination class"*, *"We steal [the FP] from another class"*). The terms *Selected* and *Discarded*, although using a tangible metaphor, can be misunderstood as the actual class (*"The class that must be selected"*) yielding misinterpretations of error directionality.

Extrapolation of Errors in End-Usage Applications - Users needed additional information to extrapolate the classification errors in end-usage applications (*"It's impossible to deduce a generality"*, *"How can I say anything about the rest of the data?"*). More information on the consequences of error was needed to decide which errors are tolerable (*"There can be risks in allowing FP, additional tests have further health risks"*, *"No guidance on how to make the tradeoff"*). Users questioned whether the error measurements are representative of end-usage conditions, regarding potential changes in class sizes and error magnitudes (*"Assuming class proportions are equal"*, *"This is a sample data, another sample could have some variations"*). They also wondered about additional sources of uncertainty, such as changes in object features or the presence of other classes (*"Will it contain only paintings and photographs?"*) and their impact on the algorithm (*"How does the classifier compute the problem"*). The lack of context information decreased user confidence, e.g., when assessing if a class is likely to be over- or under-estimated.

ROC Curve - It is unusual to visualize line charts where both x- and y-axes represent a rate, and where thresholds are a third variable encoded on the line. It is more intuitive to represent thresholds on the x-axis and rates on the y-axis, with distinct lines for each rate (as a user suggested). Non-experts primarily relied on text explanations to perform the tasks (e.g., reading that low thresholds reduce FP, then checking each dot's threshold to find the lowest). Only machine learning and math experts were comfortable with interpreting the data visually (*"My background makes me fluent in reading ROC curves visually"*, *"I don't use formulas, I compare the dots with each other without reading the values"*).

Error rate formulae were difficult to understand and remember, even for experts (*"Formulas are still confusing, and still require a lot of thinking"*). All users but one needed to reexamine the equations and their meaning many times during the tasks. It increased their response time and impacted their confidence (*"To be sure I'll need to read it again"*). Some users interpreted the rates as numbers of errors, for a simpler surrogate metric. Otherwise, without the numbers of errors, class sizes and potential imbalance are unknown, and it aggravates the difficulties with extrapolating the errors in end-results, e.g., it is impossible to assess the balance of errors between large and small classes (*"Unknown ratio of Positive/Negative"*,*"Assuming class proportions are equal"*). The error composition (how many objects from class X are confused with class Y) is unavailable for multiclass data. Some users noticed the lack of information (*"There's not enough information, errors can come from one class or another"*, *"Assuming the destination class is random"*) but others failed to notice, even for one task that was impossible to answer without knowing the error composition.

Error rates' ambiguous labels aggravated the terminology issues. The rates have actual class sizes as denominators (1) but the term *Positive* in *TP* and *FP rate* refers to the predicted class. It misled users in considering that both rates have the predicted class size as denominator, e.g., misinterpreting TP rate (1) as Precision (2). This is consistent with [16] where misinterpretations were more frequent with denominators than numerators, and with [13] where a terminology specifying the denominator of probabilistic metrics improved user understanding. A user suggested to replace TP rate by the opposite FN rate (1 - TP rate). It is more intuitive that both rates focus on errors (rather than on correct TP), and by mentioning both *Positive* and *Negative* labels, it may indicate that the denominators differ. Yet the terminology remains confusing as it fails to indicate the rate's denominator. Longer labels could clear ambiguities but may be tedious to read.

Thus ROC curves aggravated the difficulties with the terminology and error directionality, because error rate labels are ambiguous and fail to clarify the denominator. They also aggravated the difficulties with extrapolating errors in end-results because their rates fail to provide the required information, and end-users may fail to notice this limitation.

Confusion Matrix - It is unusual to interpret rows and columns as in confusion matrices, e.g., tables are usually read row per row. Users needed to reexamine the meaning of rows and columns many times during the tasks. It was difficult to remember if they represent the actual or predicted class, which aggravated the difficulties with error directionality. By confusing the meaning of rows and columns, all users but one confused FN and FP. By reading the table either row by row, or column by column, users did not consider both FN and FP (including 2 machine learning experts). The experimental visualization included large labels *Actual Class* and *Automatic Classification* to specify the meaning of rows and columns, but further clarification was needed. Row and column labels showed only the class names (e.g., *Class A*, *Class B*). It was confusing because the list of labels was identical for rows and columns. Labels could explicitly refer to the actual or predicted class, e.g., *Actual Class A*, *Classified as Class B*. One user suggested icons to provide concise indications of the meaning of rows and columns. Another suggested animations to show the relationships of rows or columns and the error directionality, e.g., a rollover on a cell shows an arrow connecting it *from* its actual class *to* its predicted class.

Thus confusion matrices aggravated the difficulties with error directionality because the visual features do not differentiate actual and predicted class. Users must rely on row and column labels, and terminology issues can arise (e.g., if the labels only mention the class names). Color codes and heatmaps can help differentiating FP from FN, but only when a class is selected (errors are FP or FN from the perspective of a specific class) and heatmaps support less accurate perceptions of magnitudes [8]. Difficulties with extrapolating the errors in end-results were also aggravated because errors are not easy to compare, i.e., users need to relate cells at different positions in the matrix.

Classee - The histograms were intuitive and quickly understood, especially for binary problems (*"This you could explain to a 5-year-old"*). For multiclass problems, it was unusual to interpret histograms where two blocks can represent the same objects. Indeed errors are represented twice: in red FN blocks for their actual class, and in black FP blocks for their predicted class. When a class is selected (Fig. 2), highlighting the related FP and FN blocks helped users to understand the error directionality (*"Highlight with rollover helps understanding how the classifier works"*) but clarifications were requested (*"You could use an arrow to show the correspondence between FP and FN"*, Fig. 5). Animations may better show the related FN and FP (e.g., FN blocks moving to the position of their corresponding FP blocks).

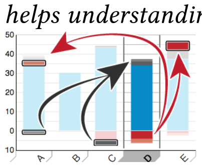

Figure 5: User suggestion

Once users familiarized with the duplicated blocks, Classee supported a correct understanding of error directionality, and answers were rarely wrong (*"It's something to get trained on"*, *"Once you get used to it, it's obvious"*). Difficulties remained with confusion matrices and ROC curves, as misunderstandings of FP and FN remained frequent. Classee better clarified the error directionality with visual features that clearly distinguish actual and predicted classes (*"I like the zero line, it makes it more visual"*). These also reduced the difficulties with the technical terminology and its explanation (*"Explanations are more difficult to understand than the graph"*, *"We usually say it's easier said than done, but here it's the opposite: when you look at the graph it's obvious"*) even though multiclass legends were unclear (*"What do you mean with 1st class and 2nd class?"*). Classee was more tangible and self-explanatory (*"I see an object that contains things"*) and non-experts were more confident than they expected (*"I am absolutely sure but I should be wrong somewhere, I'm not meant for this kind of exercise"*, *"It sounds so logical that I'm sure it's wrong"*).

Extrapolating the errors in end-results was also easier with Classee. Using numbers of errors provides complete information while ROC curves conceal the class sizes (*"You get more insights from the barchart"*). Confusion matrices also use numbers of errors, but are more difficult to interpret (cell values are difficult to compare, rows or columns can be omitted or misinterpreted). Class sizes and error balance were easier to visualize with Classee (*"Here the grey part is more important than here"*, *"Histograms are more intuitive"*).

Thus Classee limited the difficulties with extrapolating errors in end-results because its metrics and visual features are more tangible and intuitive, and they provide complete information (including class sizes and error balance). Classee also limited the difficulties with the terminology and error directionality by using visual features that clearly distinguish actual and predicted classes. Yet error directionality can be further clarified for multiclass data by adding interactive features to reinforce the correspondence of FP and FN (e.g., animations) and choose the details to display (e.g., error composition for more than 2 classes, or for specific classes).

After the experiment, we introduced the alternative visualizations. Most users preferred Classee, especially after using the other graphs (*"It's easier, I can see what I was trying to do"*, *"This is what I did in my mind to understand the threshold"*). Two machine learning experts preferred Classee, others preferred the familiar confusion matrix or ROC curve (*"You get more insights from the barchart, but ROC curve I read it in a glimpse"*) or would use both confusion matrix and Classee as they complement each other with overview and details.

7 CONCLUSION

We identified issues with the terminology, the error directionality (objects *from* an actual class are misclassified *into* a predicted class) and the extrapolation of error impacts in end-usage applications. To address these issues, labels and visual features must reinforce the identification of actual and predicted classes, e.g., using domain terminology and tangible representations (animations, icons). The third issue requires information on the error composition, and additional information to assess the validity of the error measurements w.r.t. the end-usage conditions (e.g., if test sets are representative of end-usage datasets). End-users need to investigate the statistical validity of error measurements (e.g., with variance visualization [4] that consider the class sizes of end-usage datasets [3]), and additional factors to take into account (e.g., changes in object features, class number or class sizes).

Error metrics have crucial impacts on user cognitive load. With error rates, users may overlook missing information (e.g., class sizes) and misinterpret the denominators, which is worsened by terminology issues. Raw numbers of errors are simpler to understand, but are difficult to analyse with confusion matrices.

Classee successfully addressed these issues. Its use of numbers of errors encoded in histograms is more tangible and self-explanatory, and supports accurate perceptions of error magnitudes and class sizes. The combination of 3 visual features that distinguish the actual and predicted class (position, color hue, color saturation) clarified the error directionality. It helped overcome the terminology issues while providing complete information for choosing and tuning classifiers, and for extrapolating errors in end-usage applications.

Multiclass problems remain particularly difficult to visualize. All three experimental visualizations involve unusual representations in otherwise common graphs. ROC curves have rates on both axes, confusion matrices are read both column- and row-wise, and Classee has duplicated blocks representing the same errors (as FN or FP). In our evaluation, Classee was the easiest to learn and familiarize with, but its legends and interactions should be improved (e.g., with animations highlighting the error directionality).

REFERENCES

[1] B. Alsallakh, A. Hanbury, H. Hauser, S. Miksch, and A. Rauber. 2014. Visual methods for analyzing probabilistic classification datasets. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014).

[2] E. Beauxis-Aussalet and L. Hardman. 2014. Simplifying the Visualization of Confusion Matrix. In *Benelux Conferenceon Artificial Intelligence.*

[3] E. Beauxis-Aussalet and L. Hardman. 2017. Extended Methods to Handle Classification Biases. In *IEEE International Conference on Data Science and Advanced Analytics (DSAA).*

[4] E. Beauxis-Aussalet, J. Van Doorn, and L. Hardman. 2018. *Study of User Understanding of Classification Errors.* Technical Report IA-1801. CWI.

[5] E. Beauxis-Aussalet, J. Van Doorn, M. Welling, and L. Hardman. 2016. Classee tool and D3 components. (2016). http://classee.project.cwi.nl

[6] B.J. Boom, E. Beauxis-Aussalet, L. Hardman, and R.B. Fisher. 2016. Uncertainty-Aware Estimation of Population Abundance using Machine Learning. *Multimedia Systems* 22, 6 (2016).

[7] J.P. Buonaccorsi. 2010. *Measurement Error: Models, Methods and Applications.* CRC Press, Taylor and Francis.

[8] W.S. Cleveland and R. McGill. 1984. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *J. Amer. Statist. Assoc.* 79, 387 (1984).

[9] C. Drummond and R.C. Holte. 2006. Cost Curves: An improved method for visualizing classifier performance. *Machine Learning* 65, 1 (2006).

[10] S. van den Elzen and J. J. van Wijk. 2011. Baobabview: Interactive construction and analysis of decision trees. In *IEEE Visual Analytics Science and Technology (VAST).*

[11] T. Fawcett. 2006. An introduction to ROC analysis. *Pattern Recognition Letters* 27, 8 (2006).

[12] R.B. Fisher, Y.-H. Chen-Burger, D. Giordano, L. Hardman, and F.-P. Lin (Eds.). 2016. *Fish4Knowledge: Collecting and Analyzing Massive Coral Reef Fish Video Data.* Spinger.

[13] U. Hoffrage, S. Krauss, L. Martignon, and G. Gigerenzer. 2015. Natural frequencies improve Bayesian reasoning in simple and complex inference tasks. *Frontiers in Psychology* (2015).

[14] M. Hossin and M. N. Sulaiman. 2015. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining and Knowledge Management Process (IJDKP)* 5, 2 (2015).

[15] W. Huang, P. Eades, and S.H. Hong. 2009. Measuring effectiveness of graph visualizations: A cognitive load perspective. *Information Visualization* (2009).

[16] A. Khan, S. Breslav, M. Glueck, and K. Hornbæk. 2015. Benefits of visualization in the mammography problem. *Int. J. Human-Computer Studies* (2015).

[17] J. Krause, A. Dasgupta, J. Swartz, Y. Aphinyanaphongs, and E Bertini. 2017. A Workflow for Visual Diagnostics of Binary Classifiers using Instance-Level Explanations. In *IEEE Conference on Visual Analytics Science and Technology.*

[18] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. 2012. Empirical studies in information visualization: Seven scenarios. *IEEE Transactions on Visualization and Computer Graphics* 18, 9 (2012).

[19] S. Liu, X. Wang, M. Liu, and J. Zhu. 2017. Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics* 1, 1 (2017).

[20] A.M. MacEachren. 2015. Visual Analytics and Uncertainty: It's Not About the Data. In *EuroVis Workshop on Visual Analytics.*

[21] L. Micallef, P. Dragicevic, and J.D. Fekete. 2012. Assessing the Effect of Visualizations on Bayesian Reasoning through Crowdsourcing. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012).

[22] G.M. Murch. 1984. Physiological Principles for the Effective Use of Color. *IEEE Computer Graphics and Applications* 48, 55 (1984).

[23] R. Parasuraman and V. Riley. 1997. Humans and automation: Use, misuse, disuse, abuse. *Human Factors* 39, 2 (1997).

[24] D. Ren, S. Amershi, B. Lee, J. Suh, and J.D. Williams. 2017. Squares: Supporting interactive performance analysis for multiclass classifiers. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017).

[25] F. Sebastiani. 2015. An axiomatically derived measure for the evaluation of classification algorithms. In *International Conference on The Theory of Information Retrieval.*

[26] M. Sokolova and G. Lapalme. 2009. A systematic analysis of performance measures for classification tasks. *Information Processing and Management* 45, 4 (2009).

[27] J. Talbot, V. Setlur, and A. Anand. 2014. Four experiments on the perception of bar charts. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014).

[28] J. Tidwell. 2010. *Designing interfaces: Patterns for effective interaction design.* O'Reilly Media, Inc.