Supporting End-User Understanding of Classification Errors: Visualization and Usability Issues

EMMA BEAUXIS-AUSSALET, CWI, Utrecht University
JOOST VAN DOORN, CWI, Universiteit van Amsterdam
LYNDA HARDMAN, CWI, Utrecht University

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 the specific needs of classifiers' end-users and a simplified visualization, called Classee, designed to address them. 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 visualizations 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 Classee 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.

Interaction Science Key Words: Case-Based Research, Visualization, Classification, Error and Bias.

DOI: 10.24982/jois.1814019.003

1 INTRODUCTION

Classifiers are inherently imperfect but their errors are accepted in a wide range of applications. However, end-users may not fully understand the errors and their implications [25] and may mistrust or misuse classifiers [27]. 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 investigate how to enable end-users to choose among classifiers and tuning parameters, and to understand the errors to expect when applying classifiers, e.g., as class sizes 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 formalize such tradeoff by assigning costs to FP and FN [11] 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 [25] as their underlying concepts are not common knowledge and do not easily convey the implications in end-usage applications.

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

Abbr.	*Correctness*	*Prediction*	*Definition*
FP	**False**	**Positive**	Object classified into the Positive class (i.e., as the class of interest) while actually being Negative (i.e., belonging to a class other than the Positive class).
TP	**True**	**Positive**	Object correctly classified into the Positive class.
FN	**False**	**Negative**	Object classified into the Negative class while actually belonging to the Positive class.
TN	**True**	**Negative**	Object correctly classified into the Negative class.

We discuss end-users' specific requirements, and identify information needs that pertain to either end-users or developers (Section 2). We then discuss existing visualizations of classification errors and the end-users' or developers' needs they address (Section 3).We introduce a simplified barchart visualization [4], named Classee (Figures 2, 6), that aims at addressing the specific needs of end-users (Section 4). We evaluate Classee compared to ROC curves and confusion matrices (Section 5). The suitability for specific audiences is assessed with users having three kinds of expertise: i) machine learning; ii) mathematics but not machine learning (as it may impact the understanding of error rates and ROC curves); iii) none of machine learning, mathematics or computer science. From the quantitative results, we discuss users' performance w.r.t. the type of visualization and users' level of expertise (Section 6). From the qualitative results, we identify key difficulties with understanding the classification errors, and how visualizations address or aggravate them (Section 7).

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 terminology (e.g., terms in Table 1), 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 [24].

2 INFORMATION NEEDS AND REQUIREMENTS

We identified key information needs through interviews of machine learning experts and end-users, conducted within the Fish4Knowledge and Classee projects [2, 8, 15] We found that the needs of developers and end-users have key differences and overlaps (Table 2). Their tasks require specific information and metrics which may not be provided by all visualizations.

End-users are particularly interested in estimating the magnitudes of errors to expect in specific classification end-results (e.g., within the objects classified as class *Y* how many truly belong to class *X*?). Such estimations depend on class sizes, class proportions and error compositions (i.e., the magnitude of errors between all possible classes) and can be refined depending on the features of classified objects [8, Chapter 5, Section 5.7.2] [5].

End-users also expressed concerns regarding error variability, i.e., random variance due to random differences among datasets, as well as systematic error rate differences due to lower data quality. Users' concerns are justified, as random and systematic differences among datasets significantly impact the magnitude of errors to expect in classification end-results [3].

Developers often seek to optimize classifiers on all classes and all types of error (e.g., limiting both FP and FN). They often use metrics that summarize the errors over all classes, e.g., accuracy shown in equation (3). For example, for binary classification, they measure the Area Under the Curve (AUC) to summarise all types of errors (FN and FP) over all possible values of a tuning

parameter [14]. This approach is irrelevant for end-users who apply classifiers that are already tuned with fixed parameter values.

Furthermore, metrics that summarize all types of errors for all classes (e.g., Accuracy, AUC) fail to convey *"the circumstances under which one classifier outperforms another"* [11], e.g., for which classes, class proportions (e.g., rare or large classes), types of errors (i.e., errors between specific 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 class sizes and proportions in the datasets to classify.

Class sizes and proportions (i.e., the relative magnitudes of class sizes) directly impact the magnitudes of errors. One class's size impacts the magnitude of its False Negatives, i.e., objects that actually belong to this class but are classified into another class. The larger the class, the larger the False Negatives it generates. These misclassified False Negatives are also False Positives from the perspective of the class into which they are classified. The transfer of objects *from* their actual class (as False Negatives) *into* their predicted class (as False Positives) is the core mechanism of classification errors.

To understand the impact of classification errors, it is crucial to assess the *error directionality*, i.e., the actual class *from* which errors originate, and the predicted class *into* which errors are classified. Error directionality reflects the two-fold impact of classification errors: objects are *missing* from their actual class, and are *added* to their predicted class.

Finally, to support end-users' understanding of classification errors, visualizations must provide accessible information requiring little to no prior knowledge of classification technologies. The information provided must be relevant for end-users' data analysis tasks, e.g., clarifying the practical implications of classification errors without providing unnecessary details.

Hence we identified 5 key requirements for end-user-oriented visualizations of classification errors:

- **R1: Provide the magnitude of errors for each class.**
- **R2: Provide the magnitude of each class size**, from which class proportions can be derived.
- **R3: Detail the error composition and directionality**, i.e., the errors' actual and predicted classes, and the magnitude of errors for all combinations of true and predicted classes.
- **R4: Estimate how the errors measured in test sets may differ from the errors that actually occur when applying the classifier to another dataset**, e.g., considering random error rate variance, and bias due to lower data quality or varying feature distributions.
- **R5: Omit unnecessary technical details**, e.g., about the underlying classification technologies, and information unrelated to estimating the errors to expect in classification end-results (such as the AUC metric).

Table 2: Relationships among users, tasks, information needs, metrics and visualizations.

	Task			Visualization		
	Improve Model and Algorithm	Tune Classifier	Estimate Errors in End-Results	Confusion Matrix	Precision-Recall and ROC curves	Classee
Target Audience						
End-Users		X	X			X
Developers	X	X		X	X	X
Low-Level Metric						
Raw Numbers	X	X	X	X		X
ROC-like Error Rates in equation (1)	X	X	X		X[1]	X
Precision-like Error Rates in equation (2)	X	X	X[2]		X[1]	X
Accuracy in equation (3)	X	X				X
Area Under the Curve (AUC)	X				X	X[3]
High-Level Information						
Total Number of Errors	X	X		X	X	X
Errors over Tuning Parameter	X	X			X	X
Errors over Object Features	X		X[4]			X[5]
Error Composition for Each Class	X	X	X	X	X[6]	X
Class Proportions		X	X	X		X
Class Sizes		X	X	X		X

[1] ROC curves show two error rates defined by equation (1). Precision-Recall curves show one error rate defined by equation (2), and one error rate defined by equation (1).
[2] If class proportions vary across datasets, i.e., between test and target sets, error estimation methods based on these error rates are biased [3].
[3] Barcharts' areas show information similar to AUC (Section 4).
[4] Features distributions can be used to refine error estimates [5] or identify issues with the validity of error estimation methods under varying feature distributions [3].
[5] Objects' features can be used as the x-axis dimension.
[6] Binary classification only.

Table 3: Basic of error rates, i.e., equations (1)-(3), and notation.

$\frac{n_{xy}}{n_{x.}}$	(1)	Error rates w.r.t. actual class size (e.g., ROC curves)
$\frac{n_{xy}}{n_{.y}}$	(2)	Error rates w.r.t. predicted class size (e.g., Precision)
$\frac{\sum_x n_{xx}}{n_{..}}$	(3)	Accuracy, e.g., for binary data: $\frac{\mathrm{TP+TN}}{TP+TN+FP+FN}$
n_{xy}		Number of objects actually belonging to class x and classified as class y (i.e., errors if $x \neq y$)
$n_{x.}$		Total number of objects actually belonging to class x (i.e., actual class size)
$n_{.y}$		Total number of objects classified as class y (i.e., predicted class size)
$n_{..}$		Total number of objects to classify

3 RELATED WORK

Existing visualizations - Recent work developed visualizations to improve classification models [12, 21, 23], e.g., using barcharts [1, 28]. 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, i.e., using the same representations for all algorithms, and limiting complex and unnecessary information on the underlying algorithms (Requirement R5, Section 2).

ROC curves (Figure 1), Precision-Recall curves and confusion matrices are well-established algorithm-agnostic visualizations [14] 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). Furthermore, ROC and Precision-Recall curves omit the class sizes, a crucial information needed for understanding the errors to expect in classification end-results, and tuning classifiers (Table 2, Requirement R2).

Cost curves [11] are algorithm-agnostic and investigate specific end-usage conditions (e.g., class proportions, costs of errors) but they are also complex, intended for experts, omit class sizes (Requirement R2), and do not address multiclass data. The non-expert-oriented visualizations in [20, 25] use simpler trees, grids, Sankey or Euler diagrams, but are illegible with multiclass data due to multiple overlapping areas or branches.

Choice of error metrics - Different error metrics have been developed and their properties address different requirements [18, 29, 30]. 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 [29]. These values depend on the class sizes in the test set, which is not recommended by [29]. However, raw numbers convey the class sizes, omitted in rates, but needed to assess the class proportions and the statistical significance of error measurements (Requirement R2). These are crucial for estimating the errors to expect in end-usage applications [3].

PART 1 - Explanations

Classifiers are software used to classify items into categories called **Classes**. For instance, images of cells can be classified into *healthy* or *cancerous* classes. Classifiers learn to recognise the items from each class by using sets of examples that are manually classified.

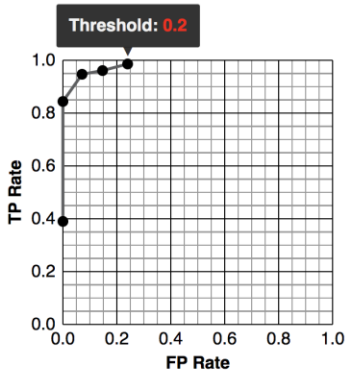
Classifiers can make errors, which are evaluated using another set of manually classified items. The true manual classifications are compared with the automatic classifications from the classifier. The evaluation reports the numbers of correct and incorrect classifications for each class.

For instance, let's consider a classifier used to select items. There are items to select (the **Positive** class), and items to discard (the **Negative** class). The classifier makes two kinds of correct classification:
- **True Positives (TP)** are items that are correctly selected (classified as *Positive*).
- **True Negatives (TN)** are items that are correctly discarded (classified as *Negative*).
Two kinds of error can occur:
- **False Positives (FP)** are items that should be discarded, but are selected (wrongly classified as *Positive*).
- **False Negatives (FN)** are items that should be selected, but are discarded (wrongly classified as *Negative*).

To limit the numbers of errors, software developpers often use a selection **Threshold**. They measure the classification errors for different thresholds, before choosing the best threshold. At each threshold, the number of correct and incorrect classifications are used to compute 2 metrics:
- **TP Rate**: the proportion of correctly selected items from the set of *Positive* items (True Positives TP and False Negatives FN)
- **FP Rate**: the proportion of wrongly selected items from the set of *Negative* items (True Negatives TN and False Positives FP)

$$\frac{TP}{TP + FN} \quad \text{TP Rate} \qquad \frac{FP}{FP + TN} \quad \text{FP Rate}$$

For instance, the visualization below reports the classification errors for 5 different thresholds. Each dot represents a threshold, and a rollover on the dots shows the threshold value (from 0.2 to 1). The X and Y axes represent the *TP Rate* and *FP Rate* measured at each threshold.



High thresholds (close to 1) discard more items that lower thresholds. For instance, high thresholds increase the False Negatives (FN). Since each False Negative (FN) represents a missing True Positive (TP), increasing the False Negatives (FN) also decreases the True Positives (TP). Similarly, high thresholds increase the True Negatives (TN) and decrease the False Positives (FP). Therefore, high thresholds decrease both *FP Rate* and *TP Rate*.

Fig. 1. Explanation of classification errors and ROC curves for binary classification, as provided to the participants of the study. The visualization shows threshold values on rollover (e.g., this screenshot shows a rollover on a data point corresponding to threshold 0.2).

Using raw numbers of errors, we focus on conveying basic error rates in equations (1)-(2), Table 3. Accuracy is a widely-used metric summarizing errors over all classes, shown in equation (3), Table 3. We also consider conveying accuracy, and focus on overcoming its bias towards large classes [18] and missing information on class sizes (Requirement R2) and error directionality, e.g., high accuracy can conceal significant errors for specific classes (Requirement R3).

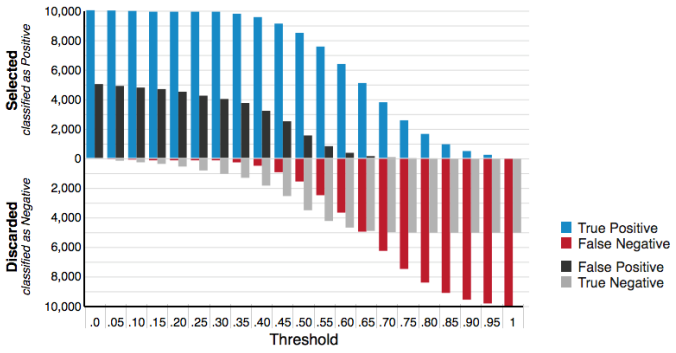


Fig. 2. Classee visualization of classification errors for binary data.

4 CLASSEE VISUALIZATION

The Classee project simplified the visualization of classification errors by using ordinary barcharts and raw numbers of errors (Figures 2 and 6). 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 bars' positions reinforces the perception of the actual class, as bars representing objects from the same actual class are staked on each other into a continuous bar, e.g., TP above FN (Figures 3 and 5, left) The zero line distinguishes the *predicted* class: TP and FP are above the zero line, FN and TN are below (Figure 3, right).

For binary data (Figure 2), objects from the same actual class are stacked in distinct bars: TP above FN for the positive class, and FP above TN for the negative class (Figure 3, left). Basic error rates can easily be interpreted visually (Figure 4). 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, i.e., equation (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 [18] 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 [14] as blue, red, black and grey areas can be perceived.

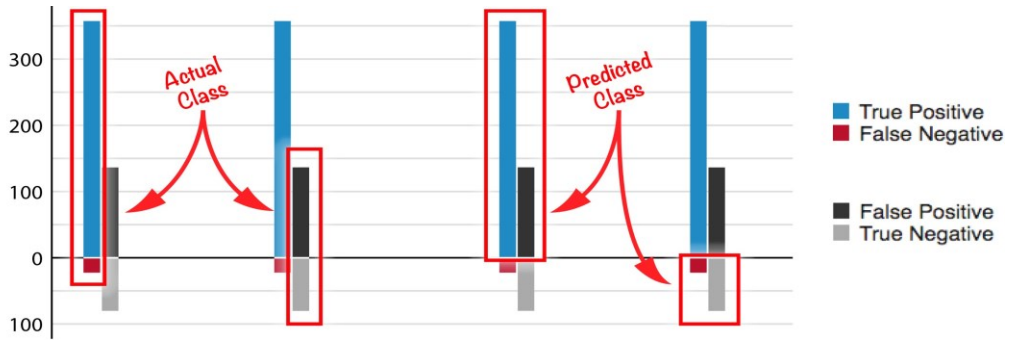


Fig. 3. Bars representing the actual and predicted classes.

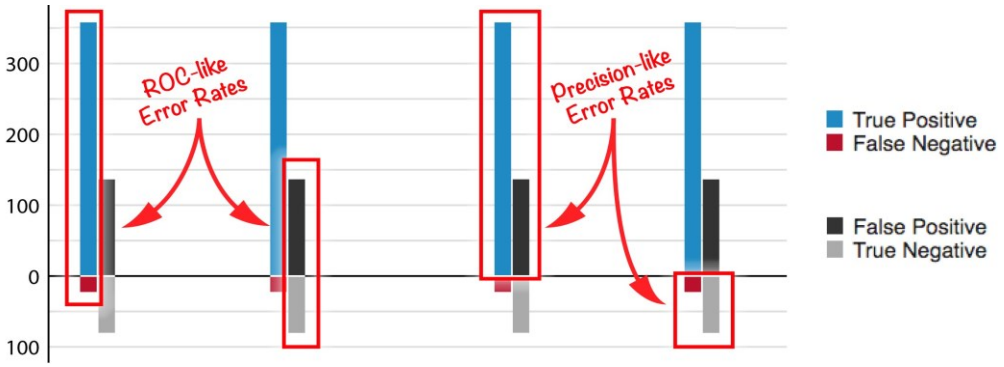


Fig. 4. Bars showing basic error rates in equations (1)-(2).

Perceiving ROC-like error rates (1) requires comparing *divided* and *adjacent* blocks. Human visual perceptions may be more accurate with *unadjacent* blocks [31], e.g., as used in [1, 28]. However, Classee shows *part-to-whole* ratios while [31] 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. Perceiving such positions is more accurate than perceiving relative bar lengths [9], which is the sole visual perception enabled in [1, 28]. Finally, precision-like error rates (2) are perceived using *aligned* and *adjacent* blocks. It supports more accurate perceptions compared to the *divided unadjacent* blocks [9, 31], e.g., as used in [1, 28].

For multiclass data (Figure 6), 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.

Basic error rates can easily be interpreted visually (Figure 4), using the same principles as for binary classification. ROC curve's error rates in equation (1) are visualized by comparing the blue and red blocks (representing the actual class, Figure 5, left). Precision-like rates in equation (2) are visualized by comparing the blue/black blocks (representing the predicted class, Figure 5, middle).

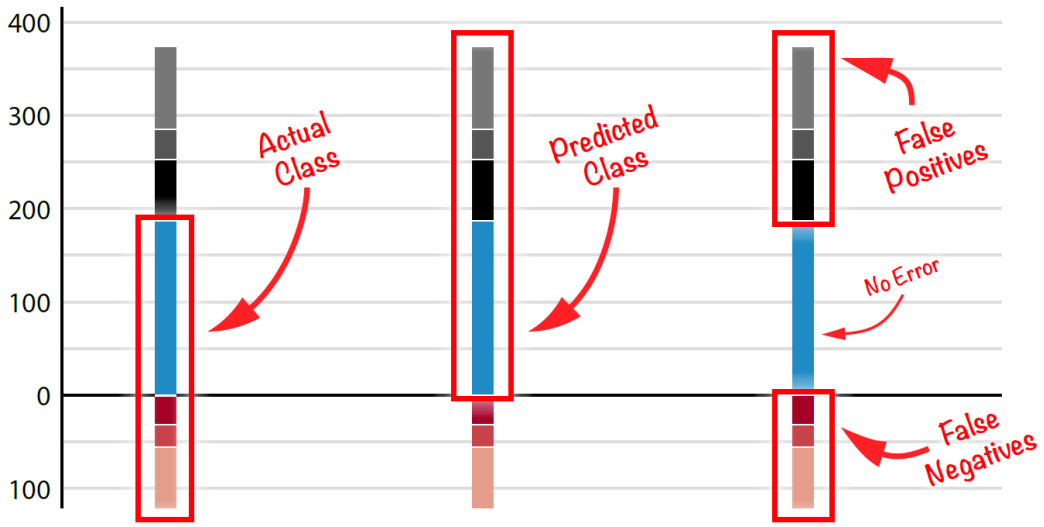


Fig. 5. Bars representing the actual and predicted classes.

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, although the exact equation of accuracy (3) is harder to interpret. However, Classee details the errors between each class, which are omitted in accuracy.

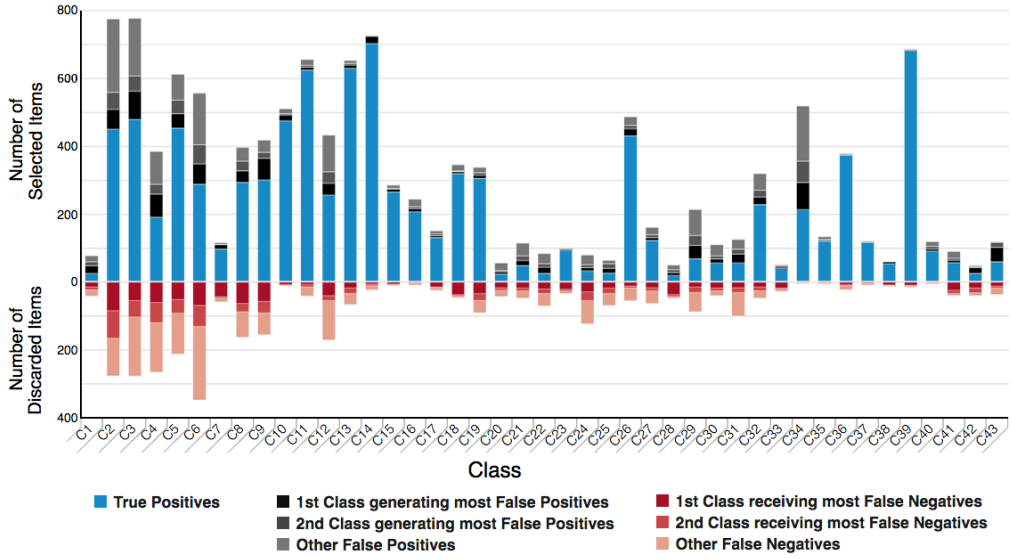


Fig. 6. Classee visualization of classification errors for multiclass data.

Compared to [28] stacking TP-FP-FN in this order, Classee stacking facilitates the interpretation of TP rates (1) and actual class sizes by showing continuous blocks for TP and FN (Figure 5, left). Compared to chord diagrams in [1] encoding error magnitudes with surface sizes, Classee uses bar length to support more accurate perceptions of error magnitudes [9].

Inspecting the error directionality, i.e., the magnitude of errors between specific classes, is crucial for understanding the impact of errors in end-results (Requirement R3, Section 2). 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. 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 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 a 3rd sub-block (Figure 7). 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 [28] may improve their rendering.

Users can select a class to inspect its errors (Figure 8). 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 Figure 8, 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 (Figure 9). 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 [9, 31]. 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 estimate the numbers of errors to expect in classification results [3].

Color choices - Classee uses blue rather than green as in [1] to address colorblindness [32] while maintaining a high contrast opposing warm and cold colors. Compared to class-specific colors in [28] which can clutters the visualization to illegibility, e.g., with more than 7 classes [26], Classee colors can handle large numbers of classes.

Following the *Few Hues, Many Values* design pattern [32], 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).

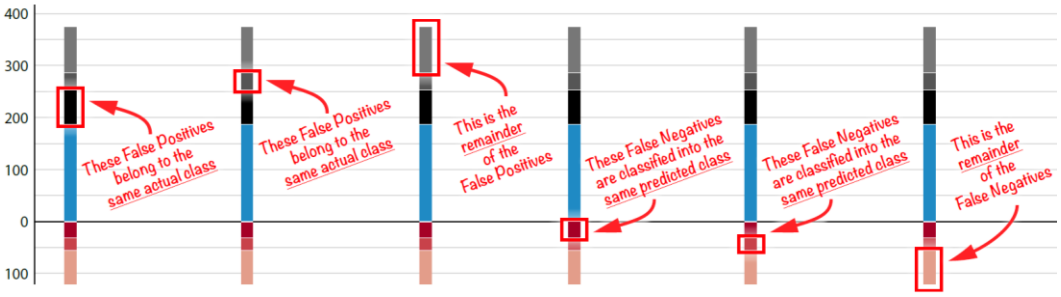


Fig. 7. Barchart blocks representing the main sources of errors.

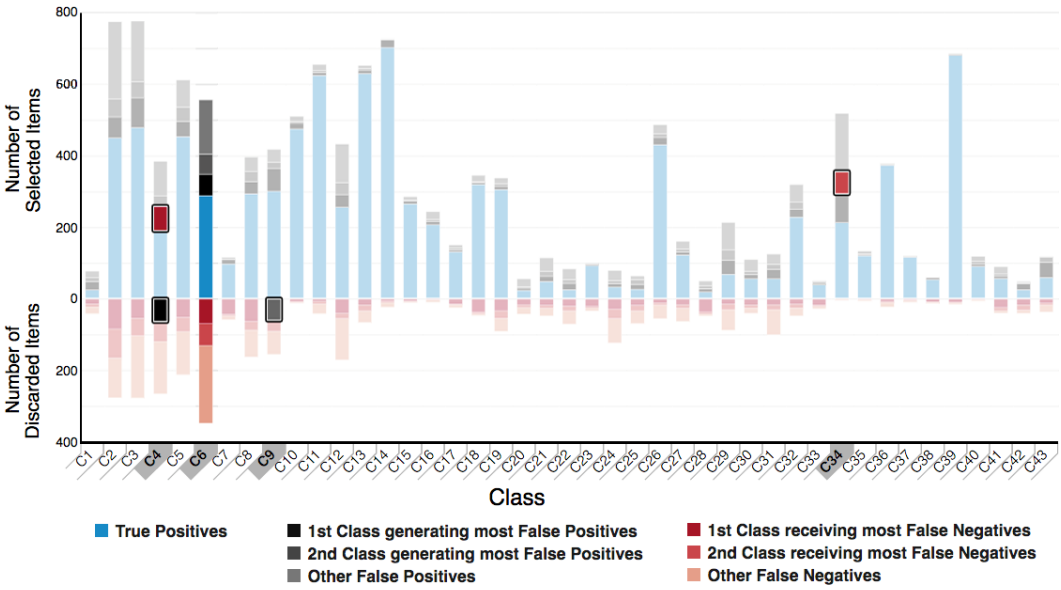


Fig. 8. Rollover detailing the errors for a specific class.

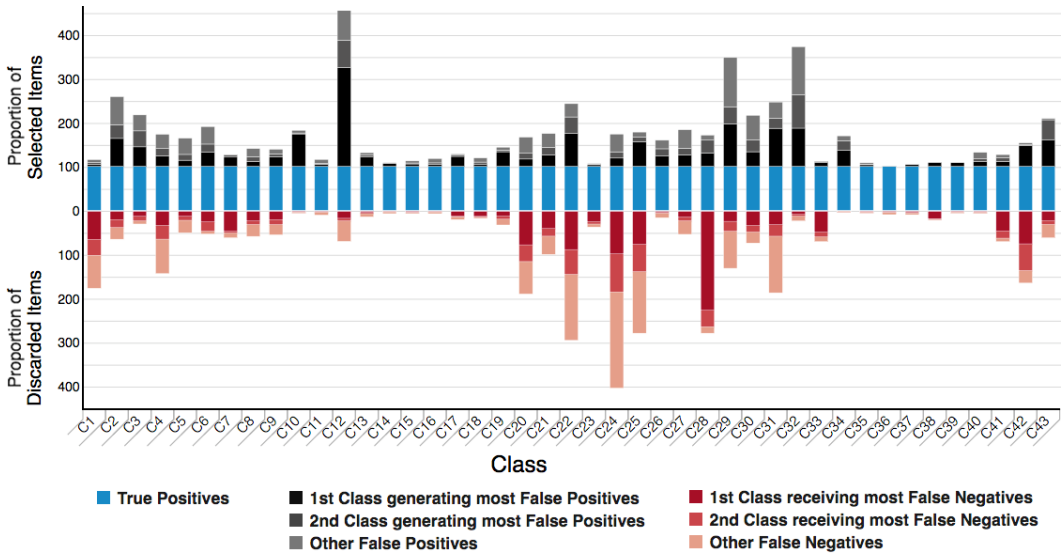


Fig. 9. Normalized view with errors proportional to True Positives.

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"* [22]. We focus on qualitatively evaluating the *Visual Data Analysis and Reasoning* [22], 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. Quantitative measurements of *User Performance* complement this qualitative study. 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 [19].

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 with 2 users per condition (3 groups x 3 visualizations x 2 users) is relatively small but was sufficient to collect important insights in our qualitative study, as we repeatedly identified key factors impacting user understanding.

The 3 experimental visualizations compared the simplified barcharts to two well-established alternatives: ROC curve and confusion matrix (Figures 10-12). ROC curves are preferred to Precision-Recall curves which exclude TN and do not convey the same information as the barcharts. All visualizations used the same data and users interacted only with one kind of visualization. This between-subject study accounts for the learning curve. After interacting with a first visualization, non-experts gain expertise that would bias the results with a second visualization.

For binary data, classification errors were shown for 5 values of a tuning parameter called a selection *threshold*. Confusion matrices for each threshold were shown as a table (Figure 11) with rows representing the thresholds, and columns representing TP, FN, TN, FP. The table included heatmaps reusing the color coding of the barcharts. The color gradients form the default heatmap

template from D3 library[1] were mapped on the entire table cells' values, which is not optimal. Each column's values have ranges that largely differ. Thus the color gradients may not render the variations of values within each column, as the variations are much smaller than the variations within the entire table. Hence color gradient should be mapped within each column separately.

For multiclass data, the confusion matrix also included a heatmap with the same color coding. The diagonal showed TP in blue scale. A rollover on a class showed the FP in dark grey scale and the FN in red scale (Figure 12, right). If no class was selected, red was the default color for errors (Figure 12, left). The ROC curves to multiclass data displayed a single dot per class, rather than complex multiclass curves. The option to normalize barchart (Figure 9) was not included, to focus on evaluating the basic barchart using raw numbers of errors.

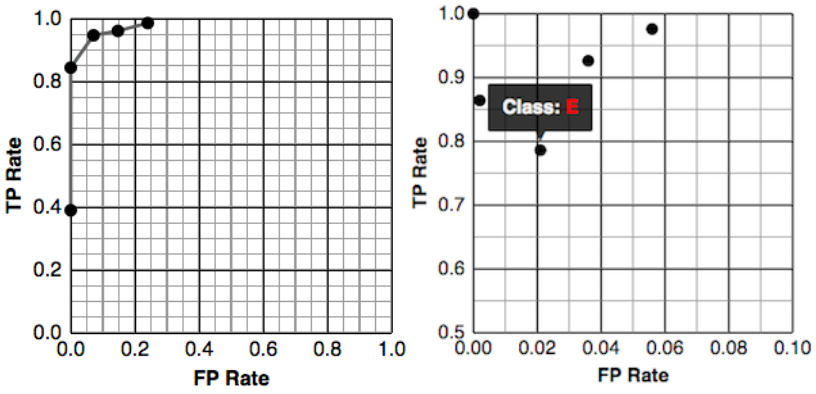


Fig. 10. ROC curves used for binary and multiclass data.

Threshold	True Positive	True Negative	False Positive	False Negative
0.2	1647	1160	367	22
0.4	1605	1302	225	64
0.6	1581	1419	108	88
0.8	1408	1527	0	261
1	651	1527	0	1018

Fig. 11. Confusion table for binary data..

[1] https://d3js.org/

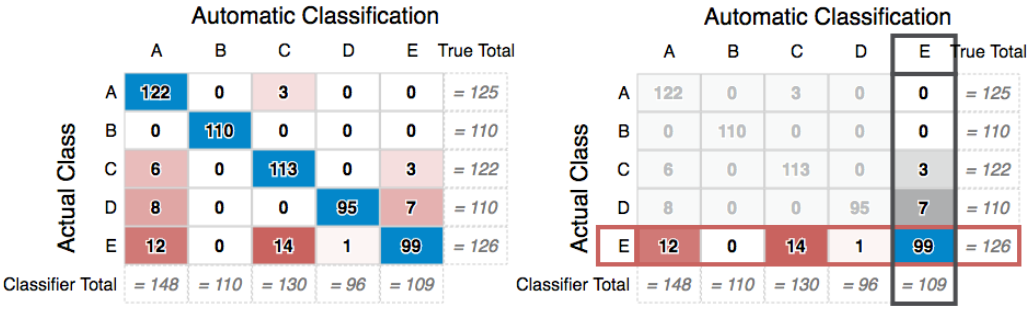


Fig. 12. Confusion matrices used for tasks T2-7 to T2-9.

The 15 user tasks were separated in two parts, for binary and multiclass data (Table 4). Each part started with a tutorial explaining the visualization and the technical concepts (Figure 1). This could be displayed anytime during the tasks. For binary problems, it explained TP, FN, FP, TN and the threshold parameter to balance FN and FP. For multiclass problems, it explained class-specific TP, FN, FP, TN in one-vs-all reductions, and that FN for one class (the actual class) are FP for another (the predicted class). The explanations of the technical concepts were the same for all users and visualizations. Only the explanations of the visualization differed.

The tasks used synthetic data that predefined the right answers. To assess user awareness of uncertainty, users had to indicate their confidence in their answers. User confidence should match the answer correctness (e.g., low confidence in wrong answers). The response time was measured, but without informing users to avoid *Time Complexity* and stress impacting user cognitive load [19]. The task complexity targeted 3 levels of data interpretation, drawn from Situation Awareness [13]. Level 1 concerned the understanding of individual data (e.g., a number of FP). Level 2 concerned the integration of several data elements (e.g., comparing FP and FN). Level 3 concerned the projection of current data to predict future situations (e.g., the potential errors in end-usage applications). To facilitate users' learning process, the tasks were performed from Level 1 to 3.

Compared to the 3 levels of *Task Complexity* in [19], our level 1 introduces a lower level of complexity. Our level 2 has less granularity and encompasses all 3 levels in [19]. Our level 3 introduces a higher level of complexity related to extrapolating unknown information (e.g., the errors to expect when applying classifiers to end-usage datasets). Our level 3 also introduces *Domain Complexity*, e.g., it concerns different application domains in tasks T1-4 to -6. The domain at hand can influence user answers. To channel this influence, tasks T2-5 to -9 are kept domain-agnostic, and T1-4 to-6 involve instructions that entail unambiguously right answers, and the same data and reasoning as previous tasks T1-1 to-3.

User feedback was collected with a questionnaire (Table 5) adapted from SUS method to evaluate interface usability [6]. Users indicated their agreement to positive or negative statements about the visualizations, e.g., disagreeing with negative statements is a positive feedback. At the very end of the experiment, we introduced the alternative visualizations to collect additional feedback with unstructured questions.

Table 4: Tasks of the experiment.

ID	*Level*	*Question*	*Right Answer*
Step 1 – Binary Classification			
T1-1	L1	Which threshold produces the most False Positives (FP)?	0.2
T1-2	L1	Which threshold produces the most False Negatives (FN)?	1
T1-3	L2	Which threshold produces the smallest sum of False Positives (FP) and False Negatives (FN)?	0.6
T1-4	L3	Choose the most appropriate threshold for person authentication? (*Task presentation tells users to limit FP*)	0.8 or 1
T1-5	L3	Choose the most appropriate threshold for detecting cancer cells? (*Task presentation tells users to limit FN*)	0.2
T1-6	L3	Choose the most appropriate threshold for detecting paintings and photographs? (*Task presentation tells users to limit both FP and FN*)	0.6
Step 2 – Multiclass Classification			
T2-1	L1	Which class has lost the most False Negatives (FN)?	Class E
T2-2	L1	Which class has the most False Positives (FP)?	Class A
T2-3	L2	Which class has the fewest False Positives (FP) and False Negatives (FN)?	Class B
T2-4	L3	Which statement is true? 1) Objects from Class A are likely to be classified as Class E. 2) Objects from Class E are likely to be classified as Class A. 3) Both statements are true. 4) No statement is true.	Statement 2
T2-5	L3	Which statement is true? 1) The number of objects in Class A is likely to be under-estimated (lower than the truth). 2) The number of objects in Class A is likely to be over-estimated (higher than the truth). 3) The number of objects in Class A is likely to be correctly estimated (close to the truth).	Statement 2
T2-6	L3	Which statement is true? 1) The number of objects in Class D is likely to be under-estimated (lower than the truth). 2) The number of objects in Class D is likely to be over-estimated (higher than the truth). 3) The number of objects in Class D is likely to be correctly estimated (close to the truth).	Statement 1
T2-7	L3	Imagine that you are particularly interested in Class D. Choose the classifier that will make the fewest errors for Class D.	Classifier 1
T2-8	L3	Imagine that you are particularly interested in Class A. Choose the classifier that will make the fewest errors for Class A.	Classifier 2
T2-9	L3	Imagine that you are interested in all the classes. Choose the classifier that will make the fewest errors for all Classes A to E	Classifier 2

Table 5: Feedback questionnaire.

F1-1,	F2-1	I would like to **use** the visualization **frequently**.
F1-2,	F2-2	The visualization is unnecessarily **complex**.
F1-3,	F2-3	The visualization was **easy to use**.
F1-4,	F2-4	I would **need the support** of an expert to be able to use the visualization.
F1-5,	F2-5	Most people would **learn** to use the visualization **quickly**.
F1-6,	F2-6	I felt very **confident** using the visualization.
F1-7,	F2-7	I would need to **learn a lot more** before being able to use the visualization.

6 QUANTITATIVE RESULTS

We discuss user prior knowledge (Figure 13), user performance between visualizations (Figure 14) and user groups (Figure 15). User performance is considered improved if i) wrong answers are limited; ii) confidence is lower for wrong answers and higher for right answers; and iii) user response time is reduced. Finally, we review the quantitative feedback (Figure 16). The detailed participants' answers are given in Figure 21 (p. 27).

Our qualitative results are not generalizable due to our small user sample. However, we briefly report them for completeness and future reference. In particular, impacts of task complexity are identified and inform the design of such quantitative studies.

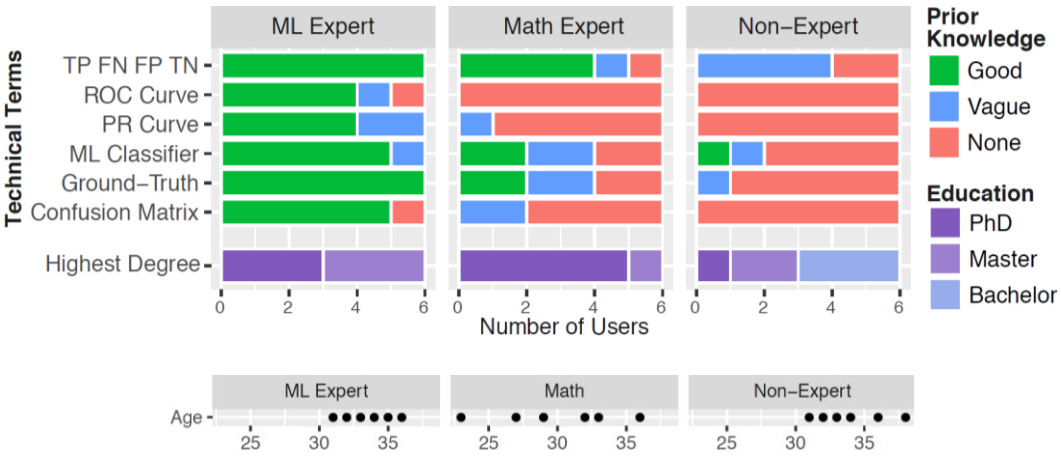


Fig. 13. Profiles of study participants.

The prior knowledge of math experts often included TP, FN, FP, TN as these are involved in statistical hypothesis testing (Figure 13). Machine learning experts knew the technical concepts well, except a self-taught practitioner who was only familiar to terms related to his daily tasks, e.g., *Accuracy* but not *ROC Curve* or *Confusion Matrix*. This participant, who was in charge of implementing, integrating and testing classifiers, 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, and related formulas, because their daily tasks involved only one of these.

Machine learning practitioners use different approaches for assessing classification errors, using specific metrics or visualizations. They may not recall the meaning and formulae of unused metrics, or even metrics used regularly. Some metrics are not part of their routine, 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, i.e., equations (1)-(2); and iii) Accuracy can also be interpreted, i.e., by comparing the relative proportions of errors (FP and FN in red and black bars) and correct classifications (TP in blue bars, TN in grey bars for binary data). Classee also shows the error composition (i.e., which specific classes are often confused) and class sizes. It supports machine learning experts tasks of tuning and improving classifiers (Table 2).

With binary data, the number of **wrong answers** differed between tasks T1-1 to -3 and T1-4 to -6 while both sets of tasks entail the same answers and use the same dataset (Figure 14, top). Tasks T1-4 to -6 involved extrapolations for end-usage applications. These tasks introduced *Domain Complexity* [19] and the tasks' description had increased *task discretion* (less detailed instructions provided to users) thus increasing the cognitive load [16]. The task discretion had an important impact as users spent considerable efforts relating the terms TP, FN, FP, TN to the real objects they represent (e.g., intruders are FP). With barcharts, user **confidence** better matched answer correctness (lower for wrong answers, higher for right answers) and so for all user profiles (Figure 15, top). Machine learning and math experts gave almost no wrong answers regardless of the visualization, but were more confident with barcharts than ROC curves (and than tables for machine learning experts). Non-experts gave more wrong answers and were over-confident with tables, but with barcharts and ROC curves their lower confidence indicates a better awareness of their uncertainty.

User **response time** was lower with barcharts (Figure 15, bottom) except for machine learning experts. Their response time was equivalent for all visualizations but were most homogeneous with ROC curves, possibly because this graph was most familiar.

With multiclass data, wrong answers were limited until task T2-4 (Figure 14, top). Answers were mostly wrong from task T2-4 onwards, as task complexity increased to concern extrapolations of errors in end-results. With barcharts, wrong answers were scarce after T2-4, e.g., after users have familiarized with the graph, but remained high with other graphs. Machine learning and math experts were more **confident** with barchart (Figure 15, middle) but non-experts were under-confident. Yet their **response time** decreased with barcharts, and was as fast as machine learning and math experts (Figure 15, bottom).

User feedback was collected twice, after the tasks for binary and multiclass data, with the same questionnaire (Table 5). **At the user profile level** (Figure 16, top), for binary data, non-experts and machine learning experts had the most negative feedback for ROC curves. Math experts had equivalent feedback for all visualizations. For multiclass data, confusion matrices had the most negative feedback from non-experts and math experts. ROC-like visualizations had the most positive feedback from all profiles. **At the question level** (Figure 16, middle), for binary data, barcharts had the most positive feedback on the design *complexity* (F1-2). ROC curves had the most negative feedback for *frequent use* and *need for support* (F1-4). For multiclass data, confusion matrices received negative feedback at all questions, especially for *confidence* and *need for training* (F2-6, -7).

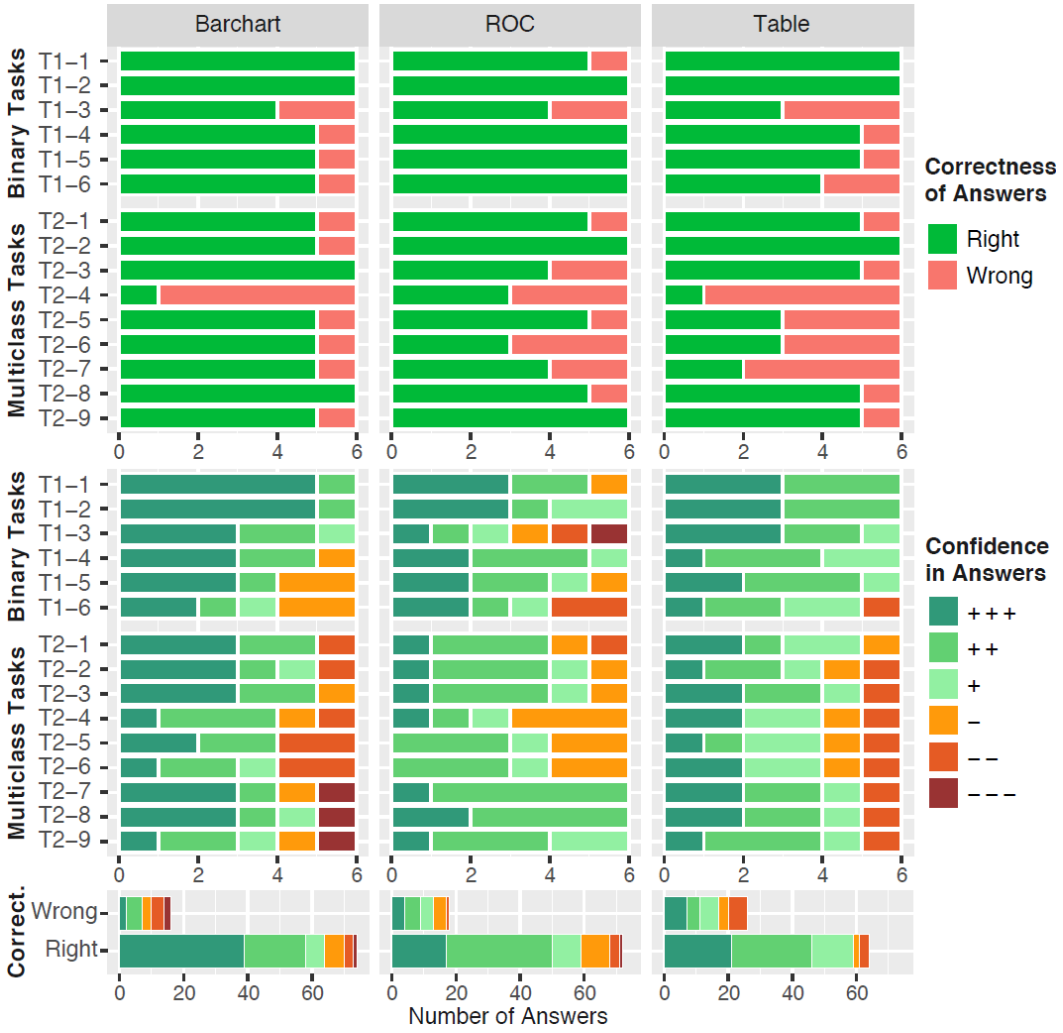


Fig. 14. Task performance per visualization (from top to bottom: right and wrong answers, confidence levels).

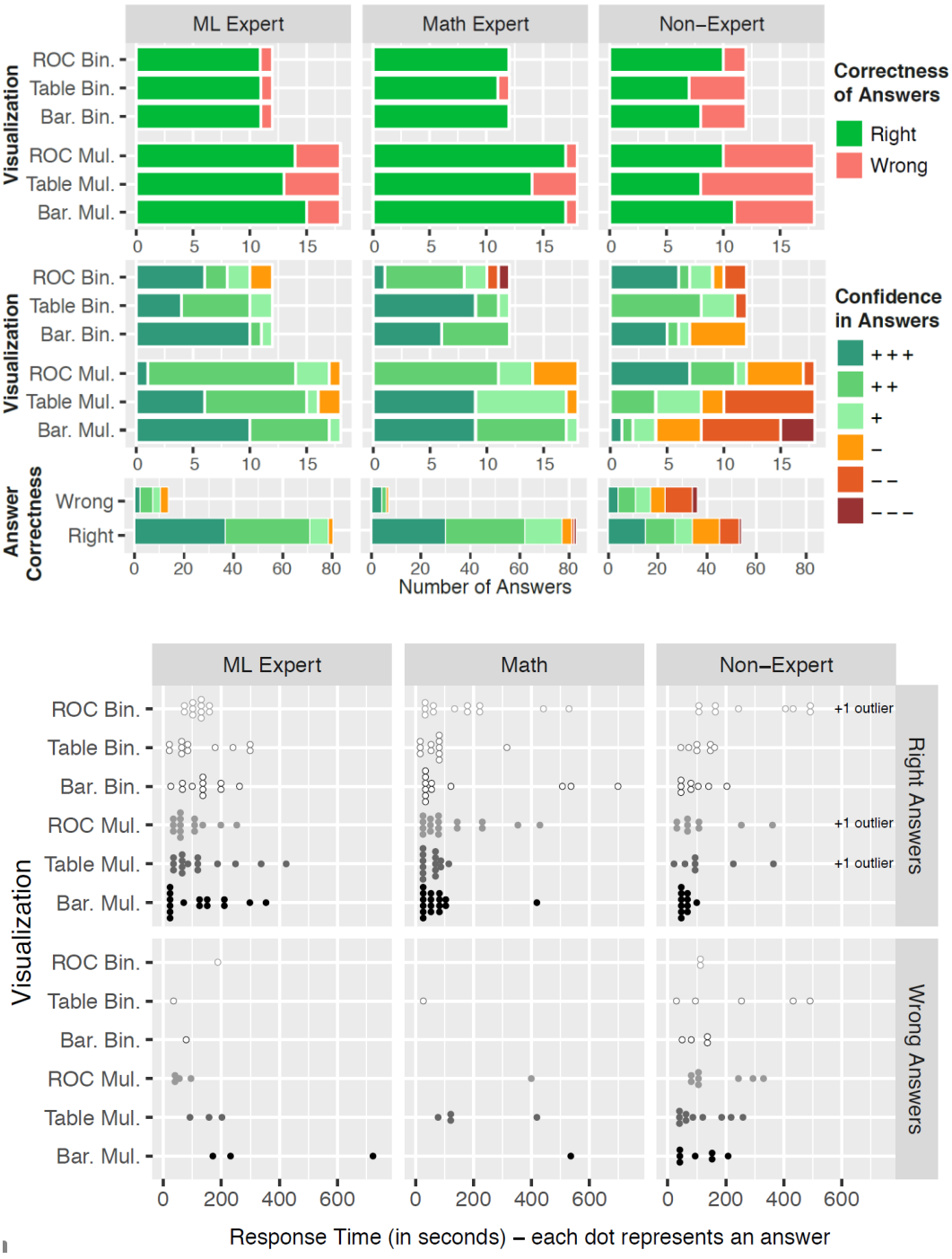


Fig. 15. Task performance per user group (from top to bottom: right and wrong answers, confidence levels, response time).

One barchart user gave the lowest possible feedback to almost all questions. This user disliked math and any form of graph (*"Ah! I hate graphs!", "I hate looking at graphs, it's too abstract for me"*) and was particularly reluctant to *frequently using* the graphs (F1-1, F2-1). However, this user's performance was excellent with barcharts for binary data: only right answers with high confidence, and positive feedback especially on the *learnability* (F1-2, *"The graph is easy, even I can use it"*).

Besides this participant, barcharts had the most positive feedback for *frequent use*, *usability* and *need for training* (F2-1, -3, -7). ROC curves had the most positive feedback on *complexity* and *learnability* (F2-2, -5) but its apparent simplicity (only 5 dots on a grid) may conceal underlying data complexity, leading to wrong answers (Figure 14).

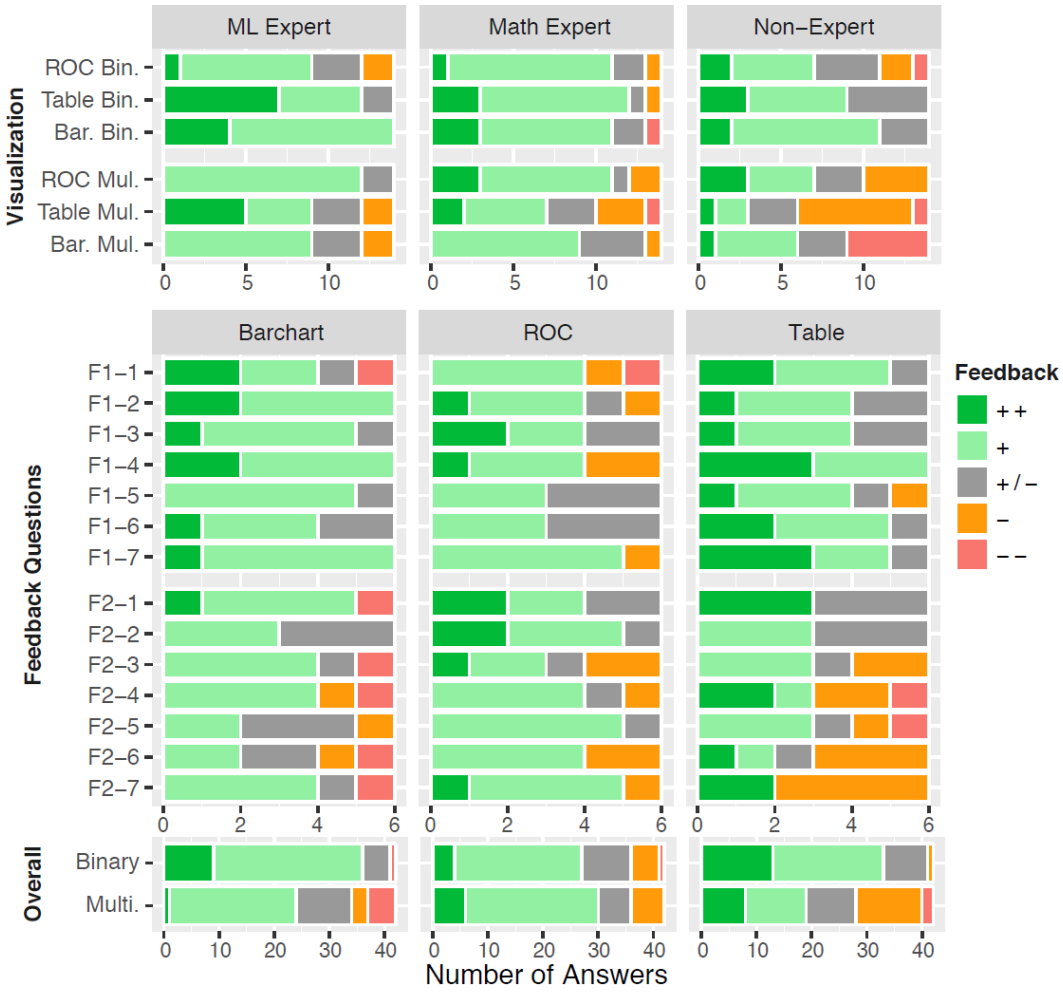


Fig. 16. User feedback.

Over all questions (Figure 16, bottom), for binary data the most negative feedback was observed for ROC curves. The feedback was equivalently positive for barcharts and tables. For multiclass data, the most negative feedback was observed for confusion matrices. The feedback was

equivalently positive for barcharts and ROC visualizations, excluding the barchart user especially averse to any data visualization.

Users wondered if their feedback should also concerned the explanations, hence the collected feedback may not concern only the visualization. Other limitations concern the small number of users, and user tendency to avoid either average or extreme feedback *("I'm not the kind of person having strong opinions"*). More detailed and generalizable insights on the usability are elicited from our qualitative analysis of user interviews.

7 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:

- The terminology (e.g., TP, FN, FP, TN are confusing terms);
- The error directionality (e.g., considering both FN and FP);
- 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.

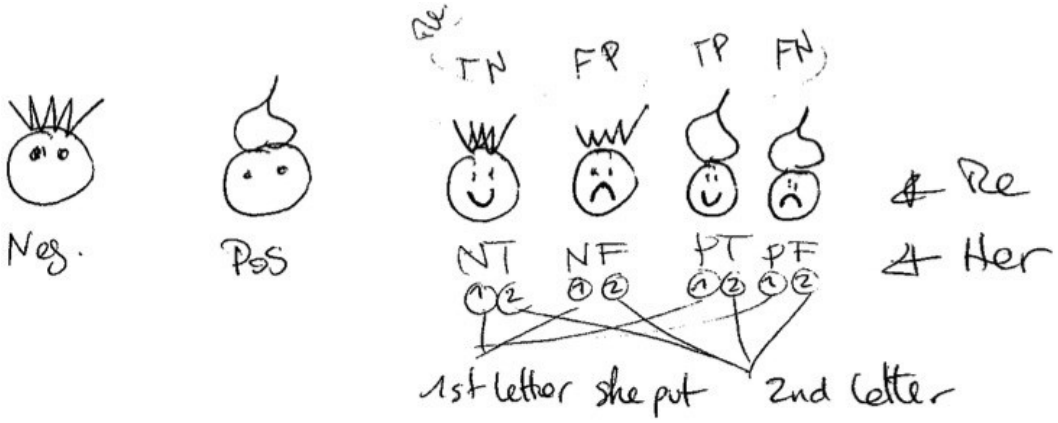


Fig. 17. User-suggested icons for TP, FN, FP, TN. Drawn by the interviewer following user's instructions in post-experiment discussions. User-suggested labels are below the icons. Usual labels were later added above.

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"*, Figure 17). This user preferred labels mentioning the actual class first (using *Negative/Positive*) then the errors (using *True/False*).

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.

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 [20] where misinterpretations were more frequent with denominators than numerators, and with [17] 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 [9]. 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 (Figure 8), 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"* Figure 18). Animations may better show the related FN and FP (e.g., FN blocks moving to the position of their corresponding FP blocks).

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"*).

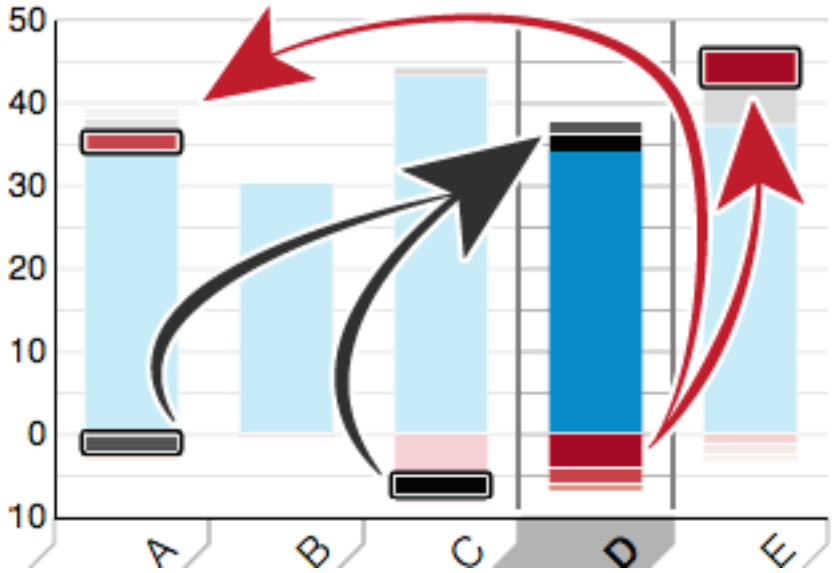


Fig. 18. User-suggested animation with arrows.

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"*). The two machine learning experts who used Classee during the experiment preferred this form of visualization. The other four 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).

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 (Requirement R1-3, Section 2). 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 the errors to expect in end-usage applications.

Multiclass problems remain particularly difficult to visualize. All three experimental visualizations involve unusual representations in otherwise common graphs. ROC curves are simple line charts, but have rates on both axes. Confusion matrices are simple tables, but must be read both column- and row-wise. Classee visualizations are simple histograms, but have 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).

Future work is required to fully address the issue of estimating the errors to expect in classification end-results. Additional information is required to assess the validity of the error measurements w.r.t. the end-usage conditions. End-users need to verify that error rates measured with specific test sets are representative of the end-usage datasets, as the class sizes, proportions, or object features may differ. Changes in object feature distributions particularly undermine error estimation methods, e.g., if end-usage datasets are of lower data quality the error rates may differ [3]. Future work is required to address this issue, i.e., to develop error estimation methods that account for feature distributions.

Future work is also required to design visualizations for exploring the relationships between classification errors and feature distributions. Visualizing classification errors as a function of varying feature distributions is complex, but Classee visualizations can provide basic templates to address this problem. For example, with binary data, Classee visualization can display the feature values as the x-axis, instead of the threshold parameter in Figure 2, and corresponding errors remain displayed as the y-axis. For multiclass data, the x-axis can also be used to represent the feature values, instead of the classes in Figure 6. However, in this case the graph can only display the errors

for a single class, hence assessing relative class sizes and error directionality is complex (Requirement R2-3).

End-users should also assess the statistical validity of error measurements, e.g., if a class is scarce in either the test set or the end-usage dataset, error estimations can be impacted by high variance. The variance of error rates can be estimated using the Sample-to-Sample method [3] and visualized with Classee as in Figures 19-20.

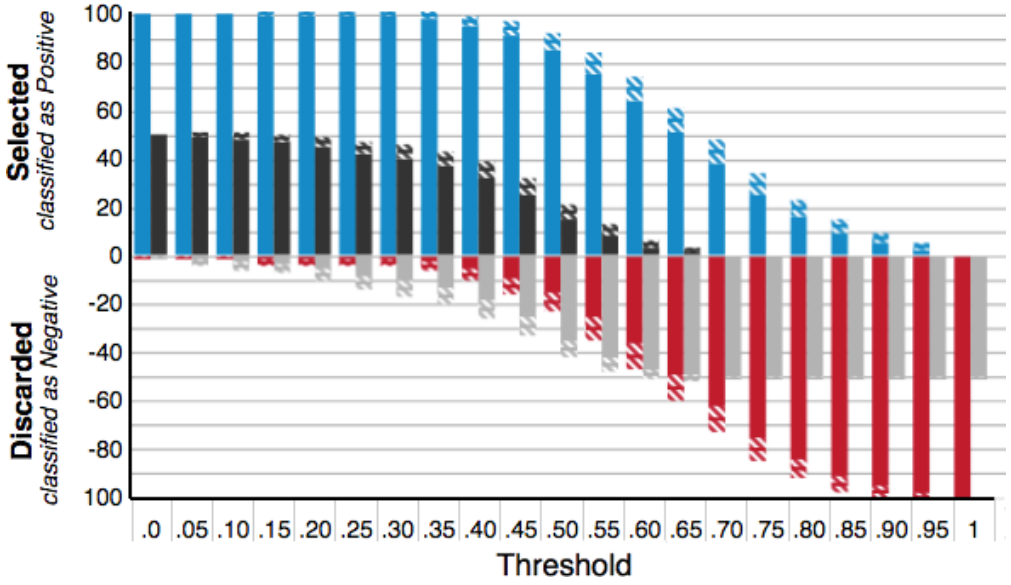


Fig. 19. Visualization of error variance, avoiding error bars as recommended by [10] .

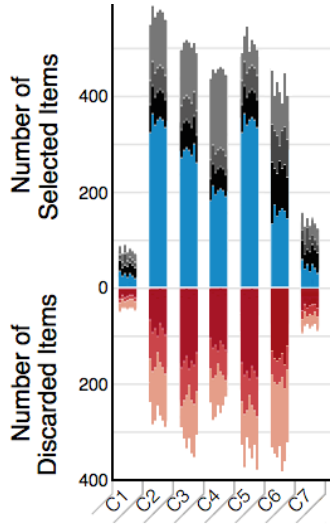


Fig. 20. Visualization of error variance for stacked barcharts, splitting the data in 10 subsamples and juxtaposing them (as stacking variance is mathematically incorrect).

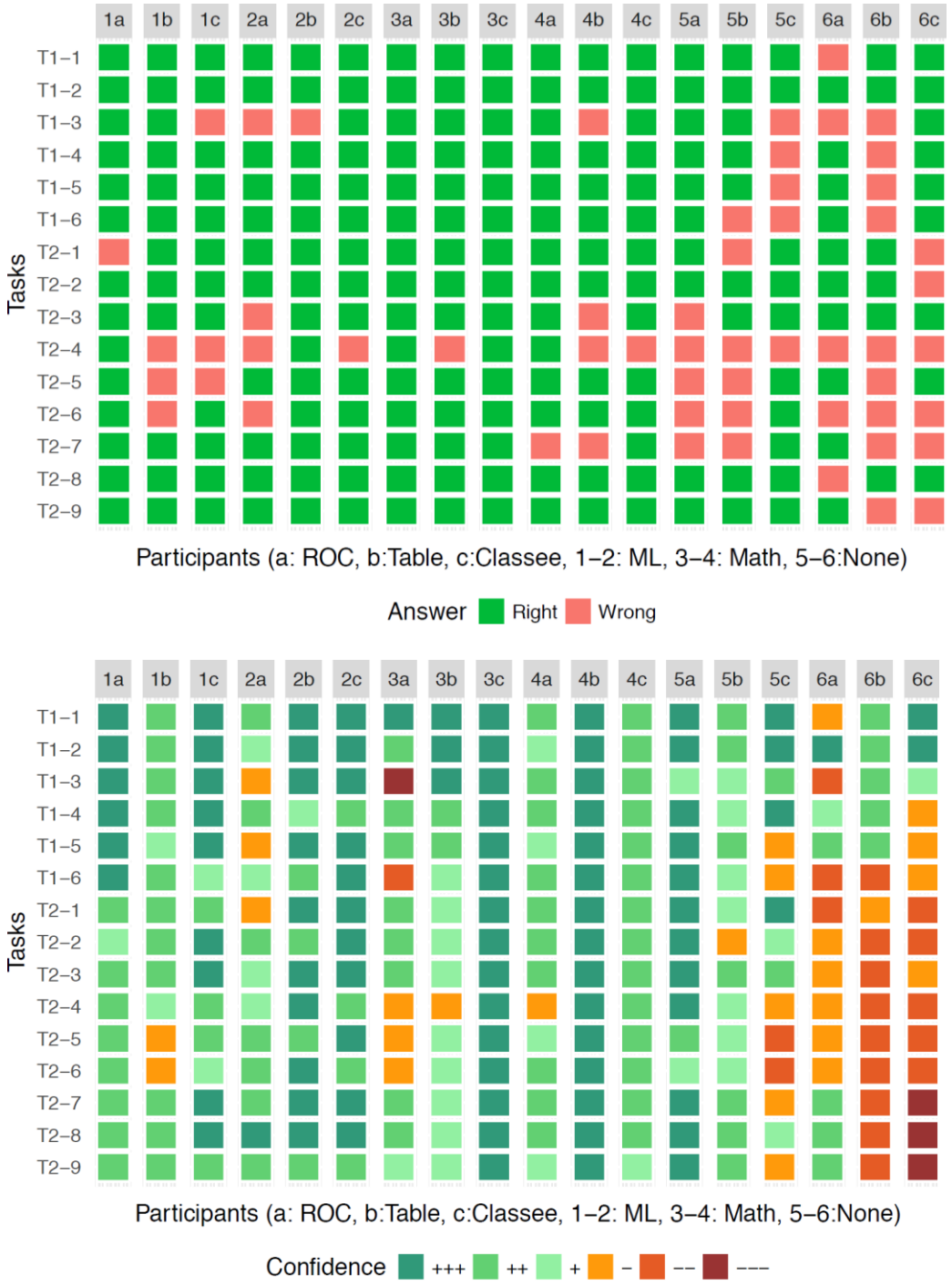


Fig. 21. Answers' correctness and confidence for each participant.

REFERENCES

[1] Alsallakh, B., Hanbury, A., Hauser, H., Miksch, S. and Rauber, A. 2014. Visual methods for analyzing probabilistic classification datasets. DOI: 10.1109/TVCG.2014.2346660. *IEEE Transactions on Visualization and Computer Graphics*. 20, 12 (2014).

[2] Beauxis-Aussalet, E. 2019. *Statistics and Visualizations for Assessing Class Size Uncertainty*. Utrecht University.

[3] Beauxis-Aussalet, E. and Hardman, L. 2017. Extended Methods to Handle Classification Biases. DOI: 10.1109/DSAA.2017.52. *IEEE International Conference on Data Science and Advanced Analytics (DSAA)* (2017).

[4] Beauxis-Aussalet, E. and Hardman, L. 2014. Simplifying the Visualization of Confusion Matrix. *26th Benelux Conference on Artificial Intelligence (BNAIC)* (2014).

[5] Boom, B.J., Beauxis-Aussalet, E., Hardman, L. and Fisher, R.B. 2016. Uncertainty-Aware Estimation of Population Abundance using Machine Learning. DOI: 10.1007/s00530-015-0479-0. *Multimedia System Journal*. 22, 6 (2016). DOI:https://doi.org/10.1007/s00530-015-0479-0.

[6] Brooke, J. 1996. SUS - A quick and dirty usability scale. *Usability evaluation in industry*. 189, 194 (1996).

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

[8] Classee tool and {D}3 components: http://classee.project.cwi.nl: 2016. *http://classee.project.cwi.nl*. Accessed: 2017-07-19.

[9] Cleveland, W.S. and McGill, R. 1984. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*. 79, 387 (1984).

[10] Correll, M. and Gleicher, M. 2014. Error bars considered harmful: Exploring alternate encodings for mean and error. DOI: 10.1109/TVCG.2014.2346298. *IEEE Transactions on Visualization and Computer Graphics*. 20, 12 (2014).

[11] Drummond, C. and Holte, R.C. 2006. Cost Curves: An improved method for visualizing classifier performance. DOI: 10.1007/s10994-006-8199-5. *Machine Learning*. 65, 1 (2006).

[12] Elzen, S. van den and Wijk, J.J. van 2011. Baobabview: Interactive construction and analysis of decision trees. DOI: 10.1007/s10994-006-8199-5. *IEEE Conference on Visual Analytics Science and Technology (VAST)* (2011).

[13] Endsley, M.R. 1995. Towards a theory of situation awareness in dynamic systems. DOI: 10.1518/001872095779049543. *Human factors*. 37, 1 (1995).

[14] Fawcett, T. 2006. An introduction to ROC analysis. DOI: 10.1016/j.patrec.2005.10.010. *Pattern Recognition Letters*. 27, 8 (2006).

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

[16] Gill, T.G. and Hicks, R.C. 2006. Task Complexity and Informing Science: A Synthesis. DOI: 10.28945/469. *Informing Science Journal*. 9, (2006).

[17] Hoffrage, U., Krauss, S., Martignon, L. and Gigerenzer, G. 2015. Natural frequencies improve Bayesian reasoning in simple and complex inference tasks. DOI: 10.3389/fpsyg.2015.01473. *Frontiers in Psychology*. 6, (2015).

[18] Hossin, M. and Sulaiman, M.N. 2015. A review on evaluation metrics for data classification evaluations. DOI: 10.5121/ijdkp.2015.5201. *International Journal of Data Mining and Knowledge Management Process (IJDKP)*. 5, 2 (2015).

[19] Huang, W., Eades, P. and Hong, S.H. 2009. Measuring effectiveness of graph visualizations: A cognitive load perspective. DOI: 10.1057/ivs.2009.10. *Information Visualization*. 8, 3 (2009).

[20] Khan, A., Breslav, S., Glueck, M. and ek, K.H. 2015. Benefits of visualization in the Mammography Problem. DOI: 10.1016/j.ijhcs.2015.07.001. *Int. J. Human-Computer Studies*. 83, (2015).

[21] Krause, J., Dasgupta, A., Swartz, J., Aphinyanaphongs, Y. and Bertini, E. 2017. A Workflow for Visual Diagnostics of Binary Classifiers using Instance-Level Explanations. DOI: 10.1109/VAST.2017.8585720. *IEEE Conference on Visual Analytics Science and Technology (VAST)* (2017).

[22] Lam, H., Bertini, E., Isenberg, P., Plaisant, C. and Carpendale, S. 2012. Empirical studies in information visualization: Seven scenarios. DOI: 10.1109/TVCG.2011.279. *IEEE Transactions on Visualization and Computer Graphics*. 18, 9 (2012).

[23] Liu, S., Wang, X., Liu, M. and Zhu, J. 2017. Towards better analysis of machine learning models: A visual analytics perspective. DOI: 10.1016/j.visinf.2017.01.006. *Visual Informatics*. 1, 1 (2017).

[24] MacEachren, A.M. 2015. Visual Analytics and Uncertainty: It's Not About the Data. DOI: 10.2312/eurova.20151104. *EuroVis Workshop on Visual Analytics (EuroVA)* (2015).

[25] Micallef, L., Dragicevic, P. and Fekete, J.D. 2012. Assessing the Effect of Visualizations on Bayesian Reasoning through Crowdsourcing. DOI: 10.1109/TVCG.2012.199. *IEEE Transactions on Visualization and Computer Graphics*. 18, 122 (2012).

[26] Murch, G.M. 1984. Physiological Principles for the Effective Use of Color. DOI: 10.1109/MCG.1984.6429356. *IEEE Computer Graphics and Applications*. 4, 11 (1984).

[27] Parasuraman, R. and Riley, V. 1997. Humans and automation: Use, misuse, disuse, abuse. DOI: 10.1518/001872097778543886. *Human factors*. 39, 2 (1997).

[28] Ren, D., Amershi, S., Lee, B., Suh, J. and Williams, J.D. 2017. Squares: Supporting interactive performance analysis for multiclass classifiers. DOI: 10.1109/TVCG.2016.2598828. *IEEE Transactions on Visualization and Computer Graphics*. 23, 1 (2017).

[29] Sebastiani, F. 2015. An axiomatically derived measure for the evaluation of classification algorithms. DOI: 10.1145/2808194.2809449. *International Conference on the Theory of Information Retrieval* (2015).

[30] Sokolova, M. and Lapalme, G. 2009. A systematic analysis of performance measures for classification tasks. DOI: 10.1016/j.ipm.2009.03.002. *Information Processing and Management*. 45, 4 (2009).

[31] Talbot, J., Setlur, V. and Anand, A. 2014. Four experiments on the perception of bar charts. DOI: 10.1109/TVCG.2014.2346320. *IEEE Transactions on Visualization and Computer Graphics*. 20, 12 (2014).

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