Diagnostic Visualization for Non-expert Machine Learning Practitioners: A Design Study

Dong Chen
Pennsylvania State University,
University Park, PA, USA
dxc360@ist.psu.edu

Rachel K. E. Bellamy*, Peter K. Malkin†, Thomas Erickson‡
IBM T. J. Watson Research Center
Yorktown Heights, NY, USA
{*rachel, †malkin, ‡snowfall}@us.ibm.com

***Abstract*—As machine learning (ML) becomes increasingly popular, developers without deep experience in ML – who we will refer to as ML practitioners – are facing the need to diagnose problems with ML models. Yet successful diagnosis requires high-level expertise that practitioners lack. As in many complex data-oriented domains, visualization could help. This two-phase study explored the design of visualizations to aid ML diagnosis. In phase 1, twelve ML practitioners were asked to diagnose a model using ten state-of-the-art visualizations; seven design themes were identified. In phase 2, several design themes were embodied in an interactive visualization. The visualization was used to engage practitioners in a participatory design exercise that explored how they would carry out multi-step diagnosis using the visualization. Our findings provide design implications for tools that better support ML diagnosis by non-expert practitioners.**

I. Introduction

Machine learning (ML) is a powerful tool for analyzing and predicting data. An ML model is created by running an ML algorithm on example inputs in order to make predictions about the value of a variable (or set of variables). Data consists of a set of instances with features, for example a dataset where each data instance is a student and the model is predicting overall grade for subsequent years might have features such as age, years of study, grade for previous years, etc. The performance of the model is tested on a data set for which the correct predictions are known. Model builders can examine content of data or parameters of a model to diagnose why a model is not doing a good job of making correct predictions, i.e. why the model is not performing well. Diagnosis can result in changes to the model input such as deciding on particular data features to leave out, or cleansing the data to remove bad values. The aim is to select features that given the ML algorithm being used lead to the best predictions overall.

Machine learning is rapidly spreading across research and application domains [1], [2]. We envision a day when ML models are a common component of software of all kinds, and, consequently, a day when ordinary developers will need to incorporate ML into their applications. We will refer to these developers – who have expertise in software development but *not* expertise in ML – as machine learning practitioners. Examples include software engineers who develop applications that need machine learning support, and researchers who use ML models to analyze their data. Our chief concern in this paper is to explore ways of supporting these ML practitioners as they incorporate ML models in their applications.

978-1-5090-0252-8/16/$31.00 © 2016 IEEE

Too often, ML practitioners report difficulty in using ML [2] to develop models with acceptable levels of performance. What the model is doing is often opaque to them [3], and the model's performance and error sources may seem disconnected [4]. As a result, non-expert ML practitioners may flounder without support for helping them identify, prioritize and remediate errors and sources of errors. For example, when an ML model performs poorly, the ML practitioner needs to diagnose the problems and take remedial actions, such as cleansing the training data to remove bad values, or deciding to discard particular data features.

This problem has captured the attention of both the HCI community and the ML community. Several papers highlight the need to lower the barrier to non-expert use of ML, e.g. [5], [6], others call for tools to be developed to aid diagnosis, e.g. [4], [6]. Aodha et al. [1] suggested that a more intuitive interface and visualization be provided to enable users to tune a model without understanding details of model parameters. Amershi et al. [7] call for increased collaboration across the fields of human computer interaction and ML as this will benefit both communities.

Limited work has been done to understand and support ML diagnosis. Patel et al. [2] conducted an observation study and identified the difficulties software developers encountered, but they were limited in providing design guidelines for supporting tools. In the report of several tools that have been developed, such as [4], [8], the authors focused more on demonstrating *what* design features were developed than on explicating *why* those design features were important and *how* they enabled diagnosis.

In this paper, we present our design process and results of a two-phase design study. We started with a "static" probe study with twelve ML practitioners, in which we probed for users' reactions to ten visualizations published in previous literature. We then summarized our findings into seven design themes. Based on the findings, we prototyped an interactive visualization embodying several of the design themes and used this to engage users in further participatory design. This paper explicates our design process and how we embody the resulting design rationale in an early prototype. Our novel contributions include: 1) an investigation of user needs in

ML diagnosis through both static and interactive design probe studies and 2) a set of design themes to help inform the design of future supporting tools for ML diagnosis.

II. Related Work

A. Empirical studies of machine learning diagnosis

Empirical study of ML practices provides first-hand information that could inspire and ground tool design. Although we can find "best practices" or general principles of ML in textbooks and relevant literature, real world practices may differ due to, for example, data corruption and time pressure. Additionally, non-experts could have developed their own way to approach ML, which may not match exactly with professional practice. Understanding their real world practice is essential for appropriate design.

Patel et al. [2] investigated the situation of software engineers using ML in their development and identified three difficulties they encountered: 1) difficulty understanding ML as an iterative and exploratory process; 2) difficulty understanding the relationship between the result (predictions) and the algorithm; and 3) difficulty evaluating model performance. Their work provides a starting point to understand the *problem space*. However, more work remains to be done to explore the *solution space*. Our work contributes to addressing the third obstacle and part of the second obstacle; that is, evaluating model performance and analyzing the relationship between errors and error sources.

Fiebrink et al. [9] conducted an observational study investigating how end users evaluate and improve a model. These findings provide information that can guide the design of tools to support ML. For example, they suggest that more detailed metrics than overall accuracy, such as error severity, could enable developers to make cost-sensitive assessment. They noted that while users were training a model, they were simultaneously "trained" by the model and learned to provide more appropriate training data. However, these particular findings are constrained to situations in which the user can efficiently supply training data. It is not clear whether these findings can be generalized to other ML diagnosis situations.

B. Visualization for machine learning

ML practitioners have been relying on summary statistics to assess the quality of a model. Metrics such as accuracy or error rate, precision and recall, and AUC (Area Under Curve) give people a brief overview of model performance. The confusion matrix visualization as shown in Figure 1a, offers a more detailed view by laying out data in a table with columns as the predicted value and rows as the actual value such that a cell contains the count of data items where the predicted value matches that actual value. The modeler's goal is to have high counts for cells where the predicted and actual value are the same. Sometimes the matrix cells are color coded to highlight the problematic predictions [10]. The problem with these conventional techniques, however, is that they only convey the performance of a model and do not inform users of error severity or causes of the errors. Users have to rely on separate tools to target data instances that were incorrectly predicted. They often have no clue what causes these errors and how to resolve them.

In recent years the research community has been calling for empowering the role of user in improving ML models [7], [1], [6]. Fails and Olson [11] proposed *interactive ML*, aimed at lowering the expertise barrier in ML and enabling users to iteratively evaluate and improve a model. A number of advanced visualization tools have been developed in recent years. SmartStripes [12] helps users interactively select features. EnsembleMatrix [10] converts the obscure activity, model ensembling, to an engaging visual interaction, and enables people to experiment with various combinations of models to optimize the result. ManiMatrix [8] empowers users to indicate their error tolerance, and lets user interactively steer a model to their preference. ModelTracker [4] provides an intuitive interface for performance analysis and debugging. Like us the designers of ModelTracker seek to facilitate ML diagnosis. The tool itself is limited to binary classification problems. In our study we include these visualizations as probe materials. By talking to participants about how they would use these visualizations, and having them act out a diagnosis scenario, we can better understand what works and what does not. They also help us seek inspirations for new visualizations to support ML diagnosis.

III. Static Design Probe Study

To understand how developers diagnose an ML model and to explore ideas for visualizations, we conducted a design probe study [13]. We presented participants with state-of-the-art visualizations from the literature[1], and asked how they would use them to understand and diagnose a model. We also asked what information was important during ML diagnosis, and what was missing from the visualizations. Using specific instances of ML visualizations provided a common ground for discussion, and exposing participants to multiple visualizations gave them the opportunity to indicate which were useful, which could be improved, and which were not useful.

A. Participants

We recruited 12 ML practitioners (10 male, 2 female; age ranges from 24 to 49 years old, M=33.5) from a large international company in the US. When asked to self-rate their ML knowledge on a scale of 1-10, participants' ratings ranged from 3-8, with an average of 5.8. All interviews but one were conducted face-to-face. The one not face-to-face was with a participant who was not in the US, and so was interviewed using a video conference system with a shared screen for material sharing. Participation was voluntary, with no reward.

B. Procedure and Probe materials

In our search for existing visualizations, we reviewed papers in several major publications in the area of design, visualization, and machine learning, such as ACM CHI, ACM IUI, and

[1]The original authors of these visualizations gave us permission to use, and customize their visualization in our study.

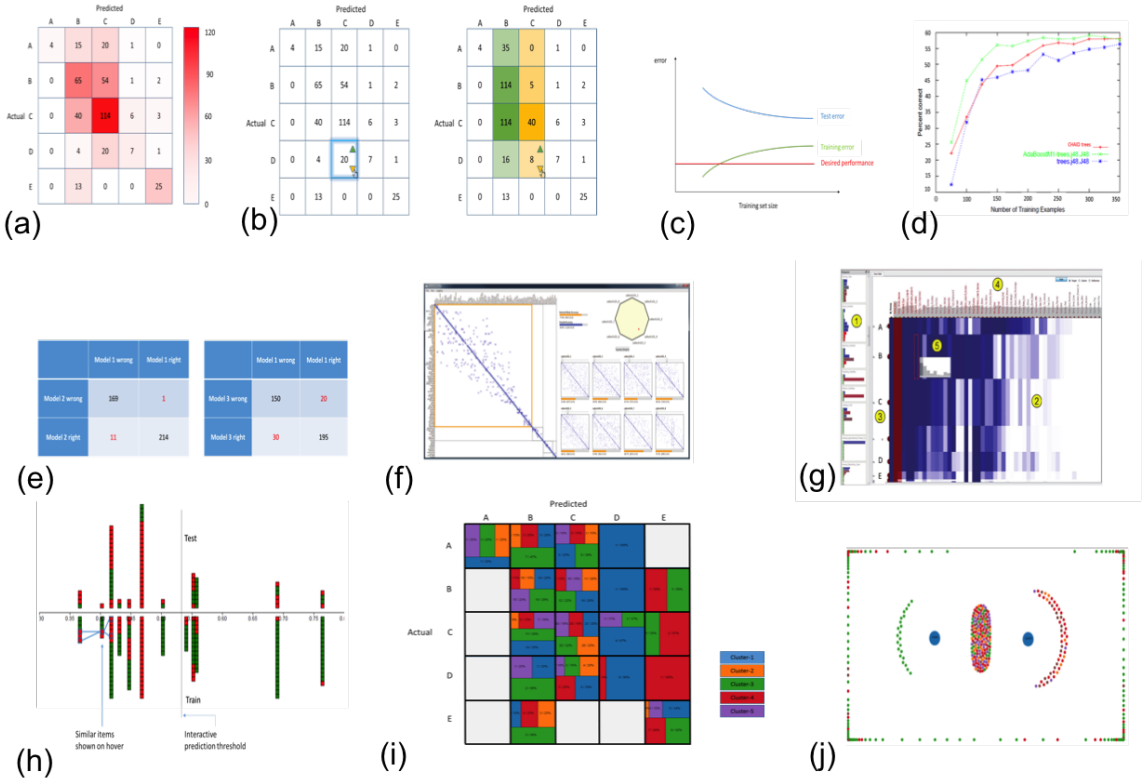


Fig. 1. Visualizations as probe materials. (a) shaded confusion matrix (b) ManiMatrix (c) learning curve (d) learning curve of multiple models (e) McNemar Test matrix (f) EnsembleMatrix (g) Customized SmartStripes (h) Customized ModelTracker (i) confusion matrix with sub-categories (j) force-directed graph originating from McNemar Test matrix

IEEE InfoVis. When searching for visualizations we did not limit our collection to those for model diagnosis only; instead, we included tools that were used in various phases in a ML pipeline. For example, SmartStripes [12] is used in feature selection; Confusion Matrix and the Learning Curve convey model performance. To compare two models, we found the matrix for McNeMar Test [14]. To compare ensemble multiple models, EnsembleMatrix [10] provides an interactive interface. And ModelTracker [4] is designed to diagnose model errors. ManiMatrix [8] enables users to interactively improve model performance by indicating error tolerance. Figure 1 a-h shows the eight existing visualizations.

We also mocked up two visualizations based on suggestions from two pilot studies with researchers in our internal group. The first one was a variant of a traditional confusion matrix. It differs in that it decomposes each cell to sub-cells, each representing a cluster of data (see Figure 1i). The idea originated from participants' desire to know which data points the number in each cell referred to. We wondered if users could gain extra information about model performance by viewing the performance of different clusters of data rather than the entire data set. Here we used K-means for clustering, but that choice was arbitrary. What we were really interested in was how participants perceived the idea. Another visualization mock up was based on the McNemar Test matrix. The idea was similar; instead of displaying an abstract number in each cell, the graph visualizes data instances (see Figure 1j). In the center of the graph lies two model nodes. Other smaller nodes represent data instances. Their positions are driven by force: attraction force if the instance is correctly predicted and repelling force if incorrectly predicted. Like the McNemar Test matrix, the instance nodes end up in four clusters: correctly predicted by both models, incorrectly by both models, correctly by one model but incorrectly by the other. The difference is this graph enables the user to drill down into instances. With these two additional visualizations, all together we had ten static visualizations that were discussed with participants.

There are three reasons to include such a wide variety of visualizations. First, while ML diagnosis is one stage of the whole pipeline, it connects closely to other stages: it examines the errors that occur in any stage. For example, an error in feature selection may result in missing features; an error in data cleansing may cause data corruption; and an error in model ensembling may cause under-optimized model combination. Therefore understanding how users address problems in other stages can benefit the design of visualizations for model diagnosis as well. Second, since different tools are designed for different purposes, they often emphasize some features while lacking others. This is not a criticism of the state-of-the-art visualizations—it is almost impossible to support such a complex task with a single tool—rather, differences in the features of each visualization allowed us to discuss with our

participants the relative importance of different features with respect to the diagnostic support they offered. Third, this probe study is exploratory, we sought inspiration for new designs rather than confirmation of the benefits provided the specific visualizations used as probes. The variety of visualizations we discussed with our participants provided an opportunity for us to gather more and more diverse ideas from them.

Interviews can flounder not only because of lack of a shared understanding between the interviewer and interviewee, but also because they move into generalities rather than specifics. To mitigate this tendency, and to further ground the conversation in specifics, we created a concrete ML scenario. The scenario describes a specific ML problem and the associated dataset. We used the dataset from UCI open dataset [15]. The task was a multiclassification problem: to predict students' course performance using the available data. We built a model but did not diagnose and improve it. We customized most visualizations to the data (except for Figure 1 f and g because they are data bound and algorithm driven).

In the interview, we presented the scenario and asked participants whether they had previously encountered similar situations. We then presented them with the visualizations one by one, in a random order to counterbalance the order effect. Each visualization was accompanied with a text explanation. If a participant had difficulty understanding the visualization, the interviewer would provide additional explanation. Note that knowing what additional information was needed to understand a visualization itself also provided insights into areas that new designs might address.

The interview was semi-structured. For each visualization we asked: (1) "what information did you learn about the model from the graph?" and (2) "what information about the model would you like to see but was not presented here?" New topics emerged as participants focused on certain features of a graph. Interviews were audio recorded and each lasted from 50-90 min. Note, although we followed many aspects of standard interview practice, our goal is design exploration and inspiration rather than confirmation or assessment.

C. Emergent Themes

All interviews were transcribed. Lacking *a priori* themes, we used an open coding approach [16] to identify emergent themes and design ideas. This resulted in seven design themes to consider when designing aids for ML diagnosis.

a) Overview + Detail: 'Overview' refers to the summary of the overall performance of the model, including metrics like accuracy, precision and recall. The overview helped participants decide if it's worthwhile to continue debugging a model: if the accuracy is too low, it may be wiser to try another ML model. However, our participants told us that summary statistics could be misleading. For example, in a multiclassification problem, a single accuracy metric is not enough. Costs may vary across misclassifications. A high level of accuracy does not mean that the model is performing well because it still may have made errors in the most important classes; conversely, low accuracy doesn't necessarily mean a model is performing poorly: it may have confused classes that are indeed similar, but done well in distinguishing among important classes. Furthermore, we were told that while summary statistics convey the presence of errors, they say little about the possible cause of these errors or severity of the errors. Our participants reported that they employed multiple tools to target the errors and determine their priority.

Almost all participants said it was useful to drill down into data instances. When they saw errors, they wanted to go to the raw data to look for patterns. For example, the Confusion Matrix listed the number of instances correctly and incorrectly predicted in each class, but it *"would be good to be able to trace down what happened in the misclassification ones"*.

b) Group and Compare: Another strategy participants mentioned is to group instances and compare the groups. We observed participants engaging in two ways of grouping. One is to group instances of the same prediction and see if any *common* patterns are evident. For example, one participant said he'd like an aggregated view of all students incorrectly predicted to receive a particular grade, as these students might share characteristics that were misleading the model. Participants also mentioned this strategy when using visualizations that didn't support it. For instance, Figure 1h (customized from ModelTracker [4]) displays data instances according to prediction confidence, and thus errors are also distributed. While users could easily examine individual error instances, it was difficult to analyze sets of errors for patterns. As one commented, *"One way to go through the errors is just go one by one...but sometimes you can see systematic problems if you see all these wrongs together. So I wonder if I could have exploded views somehow that I can see all of the guys."*

Another way is to form two groups of data and compare *different* patterns. For instance, in the Confusion Matrix, one participant stated that he wanted to compare students who actually got E but were predicted to get B (E-B) with students who got E and were predicted to E (E-E). These two groups of students actually had the same class value (E) but were treated differently by the model. Comparing the two groups might reveal what was misleading the model. Another participant suggested comparing students in E-B with students in B-B (students who were correctly predicted to get a B by the model), as these two groups were seen "alike" by the model, although they actually got different grades. The comparison might reveal features that the model missed. And a participant also suggested comparing students in B-B and E-E with students in B-E and E-B; that is, comparing students that were predicted correctly with those predicted incorrectly. In these cases, they were looking for differing patterns between the two groups that might account for the model's behavior.

c) Data space, feature space, and prediction space: Information that participants used, or that they wished they could use but was missing, can be thought of as belonging to one of the following information spaces: 1) data space, the raw data instances; 2) feature space, features that were used by the model; and 3) prediction space, predicted output from the model. Combining information from any two spaces helps to

TABLE I
DATA SPACE, FEATURE SPACE, AND PREDICTION SPACE

Information space	Question	Example
Data	What are data examples?	Spreadsheet
Feature	What are features used by models?	List
Prediction	What are the performance of the model?	Learning Curve, Confusion Matrix
Data and Feature	How are data distributed along feature values?	Tool by SmartStripes [12]
Feature and Prediction	How are features influencing predictions?	EnsembleMatrix [10]
Data and Prediction	How are subsets of data predicted by the model?	ModelTracker [4], ManiMatrix [8]
Data, Feature, and Prediction	What is causing prediction errors?	

answer different kinds of questions about the data and model (Table I). The visualizations we presented to participants were all presenting information from one space, or links between two of the spaces. For example, the Confusion Matrix and Learning Curves show the performance space. SmartStripes [12] combines the data space and feature space, and also enables investigation of the distribution of data along feature values. It allowed our participants to see if data was sparse in a feature space, or if data was over concentrated within a certain value range; both were used as indications that the feature might be inappropriate. Visualizations that combined the feature space and prediction space allowed our participants to ask what features contribute most to the prediction. This question becomes particularly important when trying to identify features that led to incorrect predictions. Many of the visualizations we used display information from the data space and the prediction space, such as ModelTracker and ManiMatrix, which show how data are predicted. However, none of the existing visualizations displays information from all three spaces. Indeed, when asked what information was missing but could be useful for model diagnosis, participants would often refer to the third space of information that was not presented in the visualization currently being discussed.

d) Diagnostic assistance: Participants wanted assistance. For example, they would like the visualization to highlight places that were problematic and indicate errors requring immediate attention. A typical strategy used to address the limited time they had to training the model was to prioritize errors needing resolution. They spent most of their time debugging high cost errors, or those that could lead to the most significant improvements in model performance. ML has some rules to judge if there is a problem and what the severity of that problem might be. A classical example is in cancer diagnosis: a false negative has more severe consequence than a false positive. So a visualization might render false negative errors in a more prominent way so that practitioners can first look into why the model did not identify the cancer. Another strategy was to focus on high-confidence misclassifications. Such misclassifications are problematic when the cost of errors is high. A visualization could map confidence to a visual feature to allow this information to be readily perceived.

Our conversations with participants indicated that sometimes the visualization would highlight the wrong thing, which made it difficult for our participants to read the visualization, and sometimes led to confusion and misinterpretation. Since visualization is largely data-driven, it is natural to encode visual cues by quantity. However, designers need to be careful about the meaning of quantity in ML. For example, in the confusion matrix, larger numbers in diagonal cells means more instances were correctly predicted, which is 'good', whereas larger numbers in off-diagonal cells means lower accuracy, which is 'bad'. Most often, however, both types of cells are encoded in the same color shade. If a model has a relatively good performance, diagonal cells will be darker and thus more prominent than off-diagonal cells and capture user's immediate attention, but in fact people are more interested in misclassifications when they are trying to improve a model. So while the color shading gives a good indication of overall performance, our participants did not find it helpful in directing their attention appropriately for diagnosis. Similar problems happened when comparing performance of two models. For instance, ManiMatrix [8] uses green and red to encode increase and decrease in cells of two confusion matrices. Two of our participants pointed out the inconsistency problem. As one of them commented, *"it's confusing that you have the two increase is good or increase is bad depending on whether you are here or there. For me instead of coding whether it's increase or decrease, I would just code whether it's better or worse ... I'm having a hard time getting my head around. so ultimately I just ignore the color and look at the number again."*

e) Interactive model diagnosis: Participants suggested that the diagnosis process is largely exploratory: they observe an error, come up with a hypothesis, make modifications, and re-run the model to test the hypothesis. In this way, they iteratively validate their hypotheses and improve the model. In line with this they wanted the visualization to let them to modify the model as they found possible error sources. For instance, if they suspect a feature is misleading, they may just remove the feature and retrain the model to see if it improves performance. As one participant said, *"It will be good if I can define my own features. For example, if I can remove features that are problematic and add features back which were removed before... I would like to control."*

f) Scalability: ML deals with big data and therefore participants put a lot of value on the visualization's ability to scale up to a large amount of data. Specifically, they asked how well it could accommodate a large number of training and test data instances, and how well it could accommodate a large number of features. And in the case of multiclassification

problem, how it would accommodate a large number of classes.

g) Psychological issues: Apart from the information required for model diagnosis, we noticed that psychological issues also affected how participants perceived a visualization – something that has often been neglected in prior research. One repeated theme was trust. If the visualization did not provide sufficient transparency to reveal how it was related to the ML model, participants had trust issues. When presented with Figure 1b, three participants said they could not trust it because they did not understand how the model learned user preferences and retrained itself. *"I don't know how this miracle is done. The confusion matrix you have here is exactly the same number as what you ask for, not more not less. This sounds like miracle"*. One approach to this problem is to make the process more transparent. As in this case, the tool was actually modifying the cost matrix to get closer to the user's expectation. One participant even suggested a solution *"If it could visualize the cost matrix as well, it would help me better understand the mechanism behind it."*

IV. Interactive Design Probe Study

The first study offers rich insights into the information needed for ML diagnosis. But since it relied on paper artifacts, we were limited in exploring *how* practitioners used diagnostic information. Participants noted that ML diagnosis is largely a data-driven, multi-step process. As next-step interactions are often driven by emerging findings, participants in the first study had difficulty explaining what they would do next without seeing the real data. For example, when asked what kind of patterns they would look for when drilling down into data, they often couldnt say: *"it is hard to say without looking at the data. It could be anything, or it could be nothing"*.

Therefore, in the second phase, we developed an interactive visualization prototype based on insights from the first study and conducted an *interactive* design probe study. This study enabled us to observe how participants dynamically diagnosed a model, enriching our findings from the first study.

A. Prototype

We prototyped an interactive visualization tool, VizML (shown in Figure 2). The prototype consists of multiple coordinated views: 1) a confusion matrix, with color shading encoding the number of instances in a cell. Note that we use two color scales for diagonal cells (correctly predicted data) and off-diagonal cells (incorrectly predicted data) as a remedy for problematic highlighting as discussed previously. 2) A scatterplot, showing data instances distributed along prediction confidence (x axis) and misclassification cost (y axis). 3) Bar charts showing data distribution along feature values. And 4) a table displaying data instances and their attribute values. None of the views is completely new in itself, and require little in the way of learning efforts from ML practitioners. We use an example usage scenario, gleaned from observations in the second study, to demonstrate the use of the tool.

An analyst was using VizML to diagnose a model trained for predicting student performance. He observed *that in the confusion matrix, of the 38 students who got an E, 25 were predicted correctly and 13 were predicted to get a B. The model mistakenly distinguished the two groups of data while they should be treated the same. He* hypothesized *that there might be some features that misled the model. To* test *the hypothesis, he selected the two cells in the matrix, which were then highlighted in different colors in all views (Figure 2). In the feature bar charts, he* observed *that the two groups varied significantly in the feature fail, which means the number of classes students failed in the past. All 13 students who were predicted B had no class failure in the past, whereas the other 25 students who were predicted E had at least one class failure before. He* hypothesized *that the feature fail was misleading the model, and that seemed to well explain the why the 13 students were predicted to get better grades.*

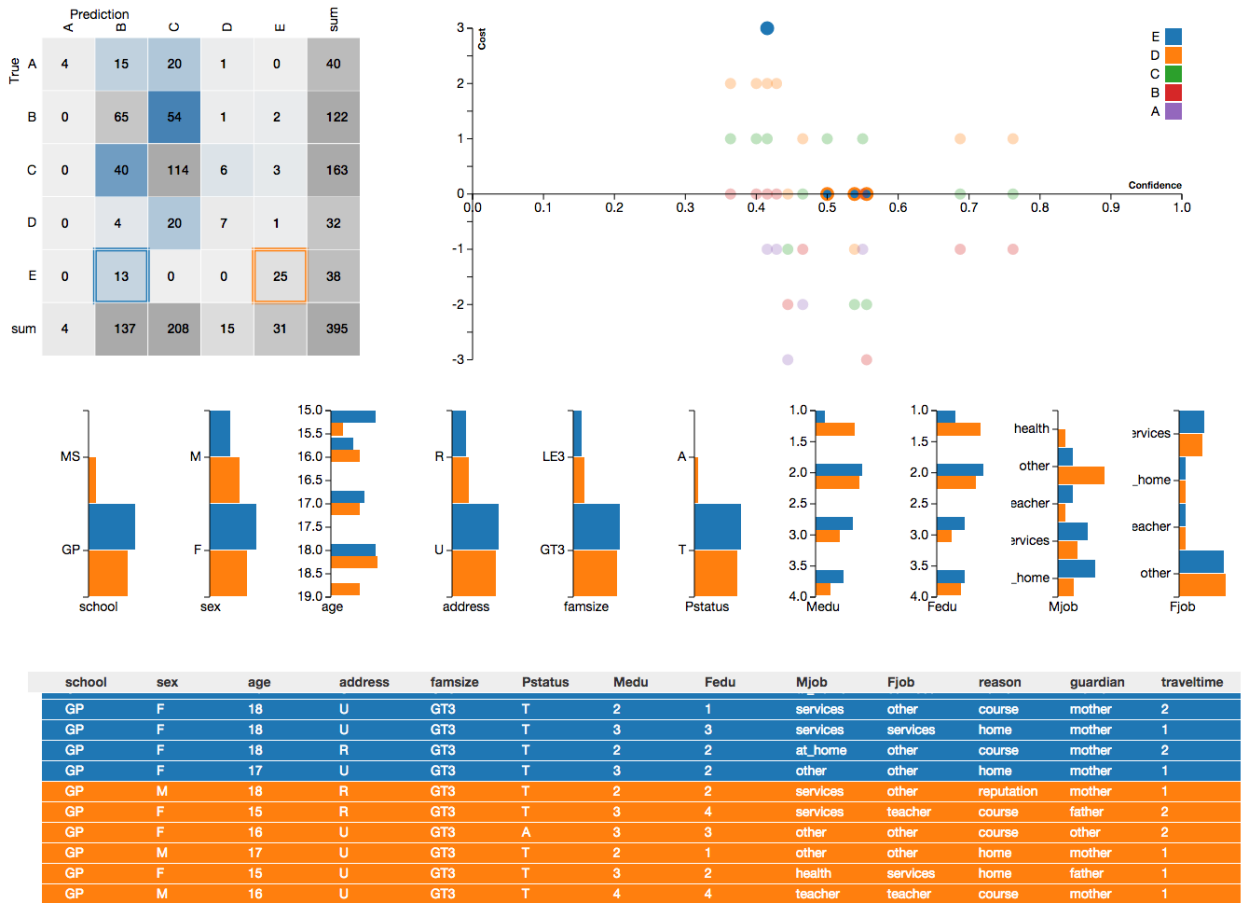
The design aligns with the findings from the first study in several ways: 1) The coordinated views allow users to zoom into details of interest as well as have an overview of the context; 2) Users can select and highlight multiple subsets of data for side-by-side comparison; 3) The views show the prediction space (confusion matrix and precision-cost scatter plot), feature space (feature bar chart) and data space (data table); and 4) the color coding in the confusion matrix and spatial layout in the scatter plot help users prioritize errors.

B. Method

We recruited four users, three from the first study, to provide feedback on our early prototype. The study used the same scenario, data, and procedure as before, except that participants were able to fluidly interact with the data. We observed how they used the prototype, noted the thoughts they expressed while doing so, and asked them for more details on why they did what they did, and how it helped them. We also listened for confusions, and asked them what was missing that might aid them in diagnosis. Each interview lasted for about one hour and was audio recorded.

C. Result

While feedback from four participants should never be taken as more than indicative, in general their feedback was positive regarding utility of VizML for ML diagnosis and its ease of use. More specifically, all participants commented that VizML allowed them to explore the data in a way that would facilitate diagnosis, and indeed we observed this as they interacted with it. As one participant, who was also in the first phase study conducted three weeks earlier, commented, *"Right, this is what we talked about [in the static probe study], [we need to] use the tool to navigate through data."*. Another commented positively on the grouping and comparing support, *"This visualization can capture the differences in the examples; I can see that being valuable."*. Participants also liked the way that the visualization helped prioritize the errors because the confusion matrix and the scatter plot *"help quickly get to the*



school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian	traveltime
GP	F	18	U	GT3	T	2	1	services	other	course	mother	2
GP	F	18	U	GT3	T	3	3	services	services	home	mother	1
GP	F	18	R	GT3	T	2	2	at_home	other	course	mother	2
GP	F	17	U	GT3	T	3	2	other	other	home	mother	1
GP	M	18	R	GT3	T	2	2	services	other	reputation	mother	1
GP	F	15	R	GT3	T	3	4	services	teacher	course	father	2
GP	F	16	U	GT3	A	3	3	other	other	course	other	2
GP	M	17	U	GT3	T	2	1	other	other	home	mother	1
GP	F	15	U	GT3	T	3	2	health	services	home	father	1
GP	M	16	U	GT3	T	4	4	teacher	teacher	course	mother	1

Fig. 2. VizML consists of a confusion matrix, a scatter plot, bar charts, and a data table. Two groups of data are highlighted here: students who actually got an E and were predicted correctly, and students who actually got an E but were predicted to get a B.

errors". When asked if they would use the tool in their real work, all participants indicated that they would like to see what their real data would be like in the tool.

An area of concern raised by participants was scalability. For example, they noted that only a limited number of classes could be displayed in the confusion matrix, and that the list of bar charts would become to long to be useful if the number of features is large.

When participants were interacting with the tool in model diagnosis, we observed several primary activities:

a) Understanding overall model performance: Participants began with an overview of the model's performance before diving into details of errors. They found the confusion matrix and scatter plot useful in that it provided richer information than overall accuracy. For example, three participants noted that the model tended to over-predict students with low grades and under-predict students with high grades, something not visible through overall accuracy. The scatter plot also enabled users to assess how "bad" the situation was. For example, they could take different approaches when wrong predictions were of low confidence or high confidence.

b) Identifying and prioritizing errors: When identifying errors, participants did not expect to find and correct all errors. Instead, they made prioritized. They started with errors of highest "impact": errors that either caused most reduction in model accuracy, or that had the highest cost. They explained that a model can never be perfect, and that they needed to train a model "well enough" within a specific period of time. Starting with high-impact errors is more time-efficient. Participants used the confusion matrix to find errors of biggest size, and used the scatter plot to identify errors of highest cost, and combined them to make a decision. For instance, participants noted that a large number of confusions existed between students who got B and C, but decided that the problem was more severe in students who got E but were predicted to B because the cost was much higher.

c) Hypothesizing error sources: In this stage, participants generated possible reasons for errors. Being able to explore and navigate between data and model helped participants target error sources. For example, in the confusion matrix, participants selected the cells E-B and E-E, and compared the feature space of two subsets to understand what was making the model generate different predictions. Participants also selected cells of E-B and B-B, in an attempt to understand why the model made the same prediction for the two subsets of students.

d) Changing model parameters and re-training: To validate their hypotheses, participants said they need to make appropriate modifications and re-train the model. For example, they would modify the features they used, or adjust the training data set. Since our tool is not integrated with ML development environment, full iterative diagnosis and testing was not possible during the study. But our observations suggest that an integrated environment would be desirable, and a system that tracks and manages hypothesis testing is needed.

V. Discussion and Conclusion

Our starting point was to make it easier for ML practitioners to diagnose ML models for use in software applications. Such practitioners have expertise in software development rather than the in the development of ML models.

In designing tools to facilitate ML diagnosis for non-expert ML practitioners, a challenge lies in addressing the tension between exposing details of the model and being generalizable to various ML algorithms. For example, work such as [17] well presents the process of how data instances are predicted by a decision tree, but it is bound to a specific algorithm. Our goal is to design a generally applicable tool that facilitates understanding and diagnosis of an ML model.

From this starting point, we needed to decide what our design process should be. User experience design is a sufficiently mature field that we had many options to choose from. We could have conducted a field study observing how users routinely diagnose a model in their real work. Such a study would have provided us with information about the diagnostic tasks of ML practitioners using their chosen tools and visualizations, from which we could have derived design requirements. However, as is often the case in design, the pragmatics of our situation did not allow for such lengthy study—participants were not willing to be observed for extended periods of time either due to personal preference or due to work practices that did not allow them to be regularly co-located with an observer, or much of their work was highly confidential. We could have designed a visualization based on our best guesses and then conducted an evaluative study to see if it supported ML diagnosis. We could even have compared our design to an existing visualization. While this would have given us much freedom in our design choices, we wanted to be informed by our target users and by design work that had already been done by others researching ML visualization. We needed to find a method for engaging with our target users, engaging with the existing ML visualization research, and engaging together in a creative design process. For these reasons we choose to use existing visualizations as design probes with ML practitioners.

Because we were engaging with practitioners in a specialized domain, and risked falling into shared generalities or miscommunication due to lack of common ground, we used an ML scenario and dataset to make our conversations with practitioners very concrete. One downside to this approach is that our scenario may have been too simplistic, and the dataset insufficiently large to more deeply address issues of diagnosing ML on massive datasets. The specific visualizations chosen also influenced the content of our conversations with practitioners. The variety of visualizations covered mitigated the latter downside, and issues of scaling were discussed and did emerge as an important theme in both studies. Having static images results in loss of value of some visualizations which were designed to be interactive, but the static probe approach proves to be cost-effective and enables us to gain insights into user information needs rapidly and effectively. The loss of interaction space design is then complemented by the interactive design probe.

In our interactive probe study, we explored several of these themes by designing and prototyping an interactive visualization called VizML. In particular we explored the usefulness of multiple coordinated views in displaying the three ML information spaces; with some views providing an overview of an ML information space and other coordinated views providing further detail. Participants reported they could easily navigate and target needed information with VizML. We found the need for a visualization that allows ML practitioners to engage in iterative cycles of exploratory observation, hypothesis formation and hypothesis testing. We identified primary activities as a means to help us develop a better conceptual model of tasks we want to support, which proves to be an effective way to ensure effectiveness of technology [18].

One concern with VizML raised by our participants is whether it can scale up. There are a few design moves we could make to mitigate issues of scaling to large data sets. We can incorporate zooming techniques and create a *zoomable confusion matrix* [19], or design ranking algorithms on the bar charts to rank the information gain by each feature (weight of feature in the model), or by similarity in the feature distribution when comparing groups of data, or strengthen visualization assistance by highlighting the most problematic errors.

We plan on using the insights gleaned from this study to update our design and conduct longer-term investigations of VizML. We are interested in how patterns of usage vary in real world practice and in different application domains. To this end, we have deployed the tool at a large US company as a web service. Company employees have access to it and can use it to examine their own data and models.

Acknowledgment

We would like to thank the IBM tooling team for providing valuable suggestions, especially John Vergo and Peri Tarr for their support. Many thanks to the machine learning practitioners who offered valuable insights in the design process.

References

[1] O. M. Aodha, M. Terry, and M. Girolami, "Putting the Scientist in the Loop - Accelerating Scientific Progress with Interactive Machine Learning," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, 2014, pp. 9–17.

[2] K. Patel, J. Fogarty, J. a. Landay, and B. Harrison, "Investigating Statistical Machine Learning as a Tool for Software Development," *Proceedings of the Conference on Human Factors in Computing Systems*, pp. 667–676, 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1357054.1357160

[3] P. Fox and J. Hendler, "The Science of Managing Data Science," *Communications of the ACM*, vol. 58, no. 6, pp. 44–47, 2015. [Online]. Available: http://online.liebertpub.com/doi/abs/10.1089/big.2014.0011

[4] S. Amershi, M. Chickering, S. M. Drucker, B. Lee, P. Simard, and J. Suh, "ModelTracker : Redesigning Performance Analysis Tools for Machine Learning," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015, pp. 337–346.

[5] K. Wagstaff, "Machine Learning that Matters," in *Proceedings of the 29th International Conference on Machine Learning*, 2012, pp. 529–536. [Online]. Available: http://icml.cc/discuss/2012/298.html

[6] K. Patel, "Lowering the barrier to applying machine learning," Ph.D. dissertation, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=1866222

[7] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the People : The Role of Humans in Interactive Machine Learning," *BE A PART OF AN*, p. 105, 2013.

[8] A. Kapoor, B. Lee, D. Tan, and E. Horvitz, "Interactive optimization for steering machine classification," in *Proceedings of the Conference on Human Factors in Computing Systems*, 2010, pp. 1343–1352. [Online]. Available: http://portal.acm.org/citation.cfm?id=1753529

[9] R. Fiebrink, P. R. Cook, and D. Trueman, "Human Model Evaluation in Interactive Supervised Learning," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 147–156, 2011. [Online]. Available: http://doi.acm.org/10.1145/1978942.1978965

[10] J. Talbot, B. Lee, A. Kapoor, and D. S. Tan, "EnsembleMatrix: Interactive Visualization to Support Machine Learning with Multiple Classifiers," in *Proceedings of the Conference on Human Factors in Computing Systems*, 2009, pp. 1283–1292.

[11] J. A. Fails and D. R. Olsen, "Interactive machine learning," *Proceedings of the 8th international conference on Intelligent user interfaces IUI 03*, pp. 39–45, 2003. [Online]. Available: http://portal.acm.org/citation.cfm?doid=604045.604056

[12] T. May, A. Bannach, J. Davey, T. Ruppert, and J. Kohlhammer, "Guiding feature subset selection with an interactive visualization," in *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, 2011, pp. 111–120. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6102448

[13] B. Gaver, T. Dunne, and E. Pacenti, "Design: Cultural probes," *interactions*, vol. 6, no. 1, pp. 21–29, Jan. 1999. [Online]. Available: http://doi.acm.org/10.1145/291224.291235

[14] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.

[15] UCI, "UCI Machine Learning Repository," 2015. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Student+Performance

[16] J. Corbin and A. Strauss, *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage publications, 2014.

[17] S. Yee and T. Chu, "A Visual Introduction to Machine Learning," 2015. [Online]. Available: http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

[18] D. A. Norman, *The design of everyday things: Revised and expanded edition*. Basic books, 2013.

[19] N. Elmqvist, T. N. Do, H. Goodell, N. Henry, and J. D. Fekete, "ZAME: Interactive large-scale graph visualization," *IEEE Pacific Visualisation Symposium 2008, PacificVis - Proceedings*, pp. 215–222, 2008.