PREPRINT: Accepted to EuroVIS 2019

A User-based Visual Analytics Workflow for Exploratory Model Analysis

Dylan Cashman[1,*], Shah Rukh Humayoun[1,*], Florian Heimerl[2], Kendall Park[2], Subhajit Das[3], John Thompson[3], Bahador Saket[3], Abigail Mosca[1], John Stasko[3], Alex Endert[3], Michael Gleicher[2], and Remco Chang[1]

[1]Tufts University, USA [2]Georgia Tech, USA [3]University of Wisconsin – Madison, USA *These two authors contributed equally

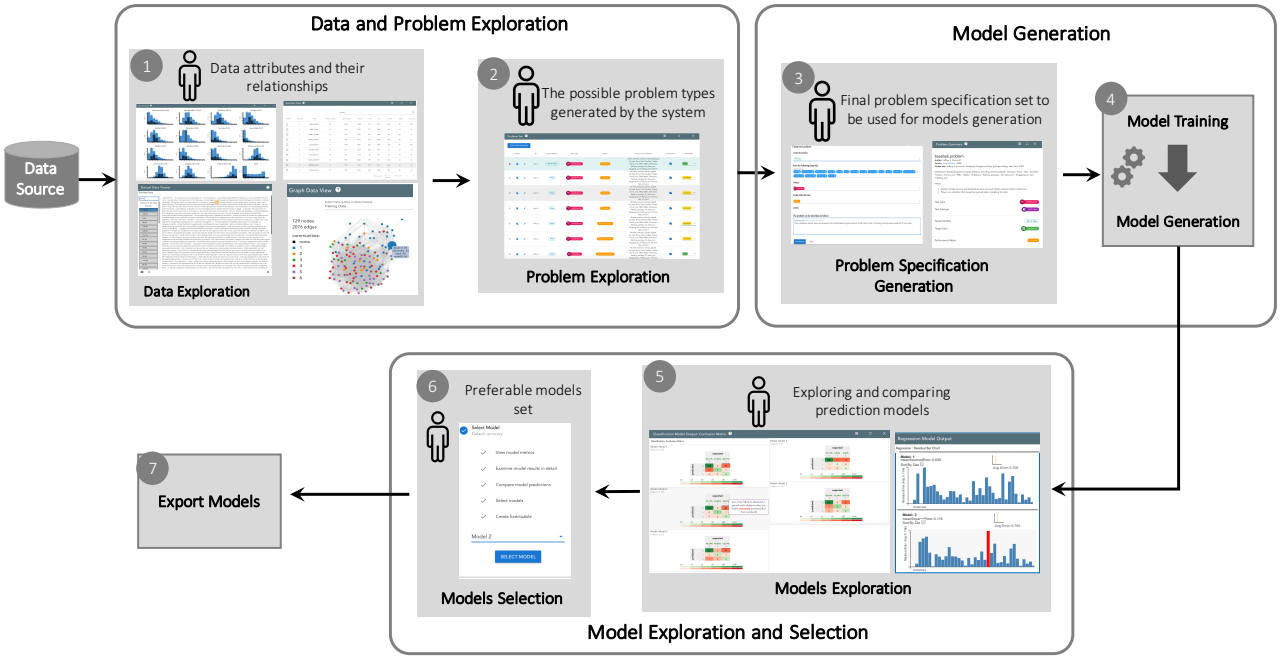

Figure 1: *The proposed EMA visual analytics workflow for discovery and generation of machine learning models. In **step 1**, the system uses interactive visualizations (such as histograms or graphs) to provide an initial data overview. The system then generates a number of possible modeling problems based on analyzing the data set (**step 2**) from which the user analyzes and selects one to try (**step 3**). Next, (**step 4**) an automated ML system trains and generates candidate models based on the data set and given problem. In **step 5**, the system shows comparisons of the generated prediction models through interactive visualizations of their predictions on a holdout set. Lastly, in **step 6**, users can select a number of preferable models, which are then exported by the system during **step 7** for predictions on unseen test data. At any time, users can return to **step 3** and try different modeling problems on the same dataset.*

Abstract

Many visual analytics systems allow users to interact with machine learning models towards the goals of data exploration and insight generation on a given dataset. However, in some situations, insights may be less important than the production of an accurate predictive model for future use. In that case, users are more interested in generating of diverse and robust predictive models, verifying their performance on holdout data, and selecting the most suitable model for their usage scenario. In this paper, we consider the concept of Exploratory Model Analysis (EMA), which is defined as the process of discovering and selecting relevant models that can be used to make predictions on a data source. We delineate the differences between EMA and the well-known term exploratory data analysis in terms of the desired outcome of the analytic process: insights into the data or a set of deployable models. The contributions of this work are a visual analytics system workflow for EMA, a user study, and two use cases validating the effectiveness of the workflow. We found that our system workflow enabled users to generate complex models, to assess them for various qualities, and to select the most relevant model for their task.

submitted to COMPUTER GRAPHICS *Forum* (6/2019).

1. Introduction

Exploratory data analysis (EDA) has long been recognized as one of the main components of visual analytics [CT05]. EDA is an analysis process through which a user "*searches and analyzes databases to find implicit but potentially useful information*" [KMSZ06], with the use of an interactive visual interface. As described by Tukey, the process of data exploration helps users to escape narrowly assumed properties about their data and allows them to discover patterns and characteristics that were not previously known [Tuk77]. In this sense, the goal of EDA and the use of traditional visual analytics systems is to help the user gain early insight into their data [Nor06, CZGR09].

However, in the modern era of big data, machine learning, and AI, visual analytics systems have begun to take on a new role: to help the user in refining *machine learning models*. Systems such as TreePOD [MLMP18], BEAMES [DCCE18], and Seq2SeqVis [SGB*18] propose new visualization and interaction techniques not for a user to better understand their data, but to understand the characteristics of the machine learning models trained on their data and the effects of modifying their parameters and hyperparameters. The goal of these visual analytics systems is to produce a predictive model which will then be used on unseen data.

These systems help analyze and refine a particular type of model with a predefined modeling goal. This limits their ability to support an exploratory analysis process since the user cannot try multiple modeling problems in the same system, and instead are confined to decision trees, regressions, and sequence-to-sequence models, respectively. In this work, we consider a previously unsupported scenario in which the *type of model and the modeling task is not known* at the beginning of the analysis. We introduce the term *Exploratory Model Analysis* (EMA), and define it as the process of exploring the set of potential models that can be trained on a given set of data. EMA shares characteristics with EDA in that both describe an analysis process that is open-ended and whose results are not clearly defined a priori, and may change and adapt during the process.

The goal of EMA is twofold: discover variables in the dataset on which reliable predictions can be made, and find the most suitable and robust types of models to predict these variables. There may be multiple models discovered at the end of the process - an analyst may end up discovering regression models between variables a, b, and c, classification models where variables d and e predict the label of variable f, and neural networks that use all independent variables to predict the value of variable g.

Despite the parallels between the two, the analysis processes that EDA and EMA describe are applicable to different sets of analysis scenarios. To illustrate the difference, consider two users of visual analytics systems in a financial services company: a broker, who must be able to explain the current state of the market, in the context of its near present and past, and the quantitative analyst, who must be able to model the future behavior of the market. The broker may use machine learning models to support their exploration of the data, but their ultimate goal is to understand current patterns in the data, so that they can make decisions in the current market landscape. In contrast, the quantitative analyst might be interested in what types of predictions are possible given the data being collected, and beyond that, which types of predictions are robust. Exploratory data analysis might expose some information that is predictive, such as the correlation between features, but for large and complex datasets, complex modeling is needed to make sufficiently robust predictions. The use of our visual analytics workflow can help the quantitative analyst to try different types of models and explore the model space.

In this example, there are two distinctions between these two users: (1) their intended goals, and (2) how data is used in the process. For the broker, the intended outcome of using visual analytics is a decision, a data item (e.g. in an anomaly detection task), or an interesting pattern within the data. The data is therefore the focus of the investigation. On the other hand, for an analyst, the intended outcome is a model (or set of models), its hyperparameters, and properties about its predictions on held out data. The data is used to train and validate the model. It is not in itself the focus of attention.

While there is a plethora of tools and techniques in the visual analytics literature that support using machine learning models, most existing workflows (such as the visual data-exploration workflow by Keim et al. [KAF*08], the knowledge generation model by Sacha et al. [SSS*14], the economic model of visualization by van Wijk [VW05], and four out of the six workflows described by Chen and Golan [CG16]) focus on the exploration and analysis of data, rather than the discovery of the model itself. These workflows presuppose that the user knows what their modeling goal was (e.g. using a regression model to predict the number of hours a patient will use a hospital bed). Although these workflows (and the many visual analytics systems built following these workflows) are effective in helping a user in data exploration tasks, we note that there is often an earlier step of modeling where users do not yet know what types of models can be built from a data source. Model exploration is an important aspect of data analysis that is underrepresented in visual analytics workflows. We do note that the two *Model-developmental Visualization* workflows from Chen and Golan do consider the goal of exporting a model rather than analyzing data [CG16]. However, they are not described in detail and only provided as abstractions. In contrast, this work delves deeply into each step of its workflow, and provides an example of its implementation.

The primary contribution of this work is a workflow for EMA that supports model exploration and selection. We first identified a set of functionality and design requirements needed for EMA through a pilot user study. These requirements are then synthesized into a step-by-step workflow (see Figure 1) that can be used to implement a system supporting EMA. To validate our proposed workflow for exploratory model analysis, we developed a prototype visual analytics system for EMA and ran a user study with nine data modelers. We report the outcomes of this study and also present two use cases of EMA to demonstrate its applicability and utility.

To summarize, in this paper we make contributions to the visual analytics community in the following ways:

- **Definition of exploratory model analysis**: We introduce the notion of exploratory model analysis and propose an initial definition.
- **Workflow for exploratory model analysis**: Based on a pilot study with users, we developed a workflow that supports exploratory model analysis.

- **User studies that validate the efficacy and feasibility of the workflow**: We developed a prototype visual analytics system based on our proposed workflow and evaluated its efficacy with domain expert users. We also present two use cases to illustrate the use of the system.

2. Related Work

2.1. Exploratory Data Analysis

The statistician Tukey developed the term *exploratory data analysis* (EDA) in his work from 1971 through 1977 [Tuk93] and his 1977 book of the same name [Tuk77]. EDA focuses on exploring the underlying data to isolate features and patterns within [HME00]. EDA was considered a departure from standard statistics in that it de-emphasized the methodical approach of posing hypotheses, testing them, and establishing confidence intervals [Chu79]. Tukey's approach tended to favor simple, interpretable conclusions that were frequently presented through visualizations.

A flourishing body of research grew out of the notion that visualization was a critical aspect of making and communicating discoveries during EDA [PS08]. This includes (static) statistical visualization libraries (such as ggplot [WC*08], plotly [SPH*16], and matplotlib [Hun07]), visualization libraries (such as D3 [BOH11], Voyager [SMWH17], InfoVis toolkit [Fek04]), commercial visualization systems (such as Tableau [tab], spotfire [spo], Power BI [pow]), and other visualization software designed for specific types of data or domain applications (for some examples, see surveys such as [DOL03, HBO*10]).

2.2. Visual Analytics Workflows

Visual analytics workflows[†], including the use of models, grew out of research into Information Visualization (Infovis). Chi and Riedl [CR98] proposed the *InfoVis reference model* (later refined by Card, Mackinlay and Shneiderman [CMS99]) that emphasizes the mapping of data elements to visual forms. The framework by van Wijk [VW05] extends this with interaction – a user can change the specification of the visualization to focus on a different aspect of the data.

The notion of effective design in Infovis has largely been summarized by Shneiderman's mantra; *Overview, zoom & filter, details-on-demand* [Shn96]. Keim et. al. noted that as data increases in size and complexity, it becomes difficult to follow such a mantra; an overview of a large dataset necessitates some sort of reduction of the data by sampling or, alternatively, an analytical model. The authors provide a framework of Visual Analytics that incorporates analytical models in the visualization pipeline [KKE10]. Wang et al. [WZM*16] extended the *models* phase in the framework by Keim et al. to include a model-building process with: *feature selection and generation*, *model building and selection*, and *model validation*. Chen and Golan [CG16] discuss prototypical workflows that include model building to aid data exploration with various degrees of model integration into the analysis workflows. Sacha et al. formalized the notion of user knowledge generation in visual analytics system, accounting for modeling in the feedback loop of a mixed-initiative system [SSZ*16]. While these frameworks have proven invaluable in guiding the design of countless visual analytics systems, they muddle the delineation between the different goals of including modeling in the visualization process, conflating model building with insight discovery.

The ontology for visual analytics assisted machine learning proposed by Sacha et al. [SKKC19] offers the clearest background on which to describe our workflow's application to EMA. In that work, the authors present a fairly complete knowledge encoding of common concepts in visual analytics systems that use machine learning, and offer suggestions of how popular systems in the literature map onto that encoding. While each step of our workflow can be mapped into the ontology, a key distinction in our workflow is in the *Prepare-Learning* process. The authors note that "in practice, quite often, the ML *Framework* was determined before the step *Prepare-Data* or even before the raw data was captured", and, in fact, none of the four example systems in that work explicitly use visual analytics to support the step of choosing a model. In our workflow, this is not the case - the framework, or machine learning modelling problem and its corresponding algorithms are not chosen a priori. We also note that the term EMA itself could comprise the entirety of the VIS4ML ontology, as each step could be useful in exploratory modeling. In that case, our workflow does not completely support EMA; such a system would need to support every single step of the ontology. However, in our definition, the choice of modeling problem is an necessary condition for EMA, and thus, our workflow is the first that is sufficient for supporting EMA with visual analytics.

Most similar to our proposed EMA workflow is the one recently introduced by Andrienko et al. [ALA*] that posits that the outcome of the VA process can either be an "answer" (to a user's analysis question) or an "externalized model". Externalized models can be deployed for a multitude of reasons, including automating an analysis process at scale or for usage in recommender systems. While similar in concept, we propose that the spirit of the workflow by Andrienko et al. is still focused on data exploration (via model generation) which does not adequately distinguish between a data- from a model-focused use case such as the aforementioned financial broker and the quantitative analyst. In a way, our EMA workflow can be considered as the process that results in an *initial model*, which can then be used as input to the model by Andrienko et al. (i.e. box (7) in Figure 1 as the input to the first box in the Andrienko model shown in Figure 2).

2.3. Modeling in Visual Analytics

We summarize several types of support for externalizing models using visual analytics with a similar categorization to that given by Liu et. al. [LWLZ17]. We summarize these efforts into four groups: visual analytics for model **explanation**, **debugging**, **construction**

[†] Frameworks, pipelines, models, and workflows are often used interchangeably in the visualization community to describe abstractions of sequences of task. In this paper, we use the word *workflow* to avoid confusion. Further, we use the word *model* to specifically refer to machine learning models and not visualization workflows.

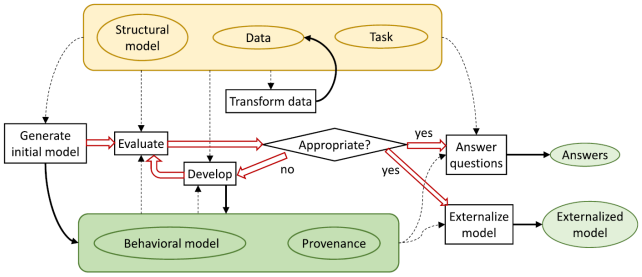

Figure 2: *The model generation framework of visual analytics by Andrienko et al. [ALA*].*

and steering, and **comparison**, noting how they differ from our definition of EMA.

Model Construction and Steering. A modeling expert frequently tries many different settings when building a model, modifying various hyperparameters in order to maximize some utility function, whether explicitly or implicitly defined. Visual analytics systems can assist domain experts to control the model fitting process by allowing the user to directly manipulate the model's hyperparameters or by inferring the model's hyperparameters through observing and analyzing the user's interactions with the visualization.

Sedlmair et al. [SHB*14] provide a comprehensive survey of visual analytics tools for analyzing the parameter space of models. Example types of models used by these visual analytics tools include regression [**?**], clustering [NHM*07, CD19, KEV*18, SKB*18], classification [VDEvW11, CLKP10], dimension reduction [CLL*13, JZF*09, NM13, AWD12, LWT*15], and domain-specific modeling approaches including climate models [WLSL17]. In these examples, the user directly constructs or modifies the parameters of the model through the interaction of sliders or interactive visual elements within the visualization.

In contrast, other systems support model steering by inferring a user's interactions. Sometimes referred to as semantic interaction [EFN12], these systems allow the user to perform simple, semantically relevant interactions such as clicking and dragging and dynamically adjusts the parameters of the model accordingly. For example, ManiMatrix is an interactive system that allows users to express their preference for where to allot error in a classification task [KLTH10]. By specifying which parts of the confusion matrix they don't want error to appear in, they tell the system to search for a classification model that fits their preferences. Disfunction [BLBC12] allows the user to quickly define a distance metric on a dataset by clicking and dragging data points together or apart to represent similarity. Wekinator enables a user to implicitly specify and steer models for music performance [FTC09]. BEAMES [DCCE18] allows a user to steer multiple models simultaneously by expressing priorities on individual data instances or data features. Heimerl et. al. [HKBE12] support the task of refining binary classifiers for document retrieval by letting users interactively modify the classifier's decision on any document.

Model Explanation. The explainability of a model is not only important to the model builder themselves, but to anyone else affected by that model, as required by ethical and legal guidelines such as the European Union's General Data Protection Regulation (GDPR) [Cou18]. Systems have been built to provide insight into how a machine learning model makes predictions by highlighting individual data instances that the model predicts poorly. With Squares [RAL*17], analysts can view classification models based on an in-depth analysis of label distribution on a test data set. Krause et. al. allowed for instance-level explanations of triage models based on a patient's medication listed during intake in a hospital emergency room [KDS*17]. Gleicher noted that a simplified class of models could be used in a VA application to trade off some performance in exchange for a more explainable analysis [Gle13]. Many other systems and techniques purport to render various types of models interpretable, including deep learning models [LSC*18, LSL*17, SGPR18, YCN*15, BJY*18], topic models [WLS*10], word embeddings [HG18], regression models [**?**], classification models [PBD*10, RSG16, ACD*15], and composite models for classification [LXL*18]. While model explanation can be very useful in EMA, it does not help a user discover models, it only helps interpret them. It is, however, a key tool in the exploration and selection of models (steps 5 and 6 of our workflow in Figure 1.

Model Debugging. While the calculations used in machine learning models can be excessively complicated, endemic properties of models that cause poor predictions can sometimes be diagnosed visually relatively easily. RNNBow is a tool that uses intermediate training data to visually reveal issues with gradient flow during the training process of a recurrent neural network [CPMC17]. Seq2Seq-Vis visualizes the five different modules used in sequence-to-sequence neural networks, and provides examples of how errors in all five modules can be diagnosed [SGB*18]. Alsallakh et al. provide several visual analysis tools for debugging classification errors by visualizing the class probability distributions [AHH*14]. Kumpf et al. [KTB*18] provide an interactive analysis method to debug and analyze weather forecast models based on their confidence estimates. These tools allow a model builder to view *how* and *where* their model is breaking, on specified data instances. Similar to model explanation, it incrementally improves a single model rather than discovers new models.

Model Comparison. The choice of which model to use from a set of candidate models is highly dependent on the needs of the user and the deployment scenario of a model. Gleicher provides strategies for accommodating comparison with visualization, many of which could be used to compare model outputs [Gle18]. Interactivity can be helpful in comparing multiple models and their predictions on a holdout set of data. Zhang et. al. recently developed Manifold, a framework for interpreting machine learning models that allowed for pairwise comparisons of various models on the same validation data [ZWM*18]. Mühlbacher and Piringer [**?**] support analyzing and comparing regression models based on visualization of feature dependencies and model residuals. TreePOD [MLMP18] helps users balance potentially conflicting objectives such as accuracy and interpretability of decision tree models by facilitating comparison of candidate tree models. Model comparison tools sup-

port model selection, but they assume that the problem specification that is solved by those models is already determined, and thus they do not allow exploration of the model space.

3. A Workflow for Exploratory Model Analysis

The four types of modeling described above all presuppose that the user's modeling task is well-defined: the user of the system already knows what their goal is in using a model. We contend that our workflow solves a problem that is underserved by previous research - Exploratory Model Analysis (EMA). In EMA, the user seeks to discover what modeling can be done on a data source, and hopes to export models that excel at the discovered modeling tasks. Some of the cited works do have some exploratory aspects, including allowing the user to specify which feature in the dataset is the target feature for the resulting predictive model. However, to the best of our knowledge, no existing system allows for multiple modeling types, such as regression and classification, within the same tool.

Beyond the types of modeling outlined above, there are two new requirements that must be accounted for. First, EMA requires an interface for modeling problem specification - the user must be able to explore data and come up with relevant and valid modeling problems. Second, since the type of modeling is not known *a priori*, a common workflow must be distilled from all supported modeling tasks. All of the works cited above are specifically designed towards a certain kind of model, and take advantage of qualities about that model type (i.e. visualizing pruning for decision trees). To support EMA, an application must support model discovery and selection in a general way.

In this section, we describe our method for developing a workflow for EMA. We adopt a user-centric approach that first gathers task requirements for EMA following similar design methodologies by Lloyd and Dykes [LD11] and Brehmer et al. [BISM14]. Specifically, this design methodology calls for first developing a prototype system based on best practices. Feedback by expert users are then gathered and distilled into a set of design or task requirements. The expert users in this feedback study were identified by the National Institute of Standards and Technology (NIST) and were trained in data analysis. Due to confidentiality reasons, we do not report the identities of these individuals.

3.1. Prototype System

Our goal in this initial feedback study was to distill a common workflow between two different kinds of modeling tasks. Our initial prototype system for supporting exploratory model analysis allowed for only two types of models – classification and regression. The design of this web-based system consisted of two pages using tabs, where on the first page, a user sees the data overview summary through an interactive histogram view. Each histogram in this view represented an attribute/field in the data set, where the x-axis represented the range of values while the y-axis represented the number of items in each range of values. On the second tab of the application, the system showed a number of resulting predicted models based on the underlying data set. A screenshot of the second tab of this prototype system is shown in Figure 3.

Classification models were shown using scatter plots, where each scatter plot showed the model's performance on held out data, projected down to two dimensions. Regression models were visualized using bar charts, where each vertical bar represented the amount of residual and the shape of all the bars represents the model's distribution of error over the held out data.

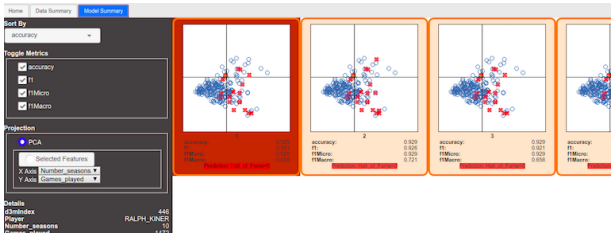

Figure 3: *A prototype EMA visual analytics system used to determine task requirements. Classification is shown in this figure. During a feedback study with expert users, participants were asked to complete model selection tasks using this view. This process is repeated for regression models (not shown). Feedback from this prototype was used to distill common steps in model exploration and selection across different model types.*

3.2. Task Requirements

We conducted a feedback study with four participants to gather information on how users discover and select models. The goal of the study was to distill down commonalities between two problem types, classification and regression, in which the task was to export the best predictive model. Each of the four participants used the prototype system to examine two datasets, one for a classification task and the other for regression. Participants were tasked with exporting the best possible predictive model in each case. The participants were instructed to ask questions during the pilot study. Questions as well as think-aloud was recorded for further analysis. After each participant completed their task, they were asked a set of seven open-ended questions relating to the system's workflow, including what system features they might use for more exploratory modeling. The participants' responses were analyzed after the study and distilled into a set of six requirements for exploratory model analysis:

- *G1: Use the data summary to generate prediction models:* Exploration of the dataset was useful for the participants to understand the underlying dataset. This understanding can then be transformed into a well-defined problem specification that can be used to generate the resulting prediction models. Visualization can be useful in providing easy exploration of the data, and cross-linking between different views into the dataset can help facillitate understanding and generate hypotheses about the data.
- *G2: Change and adjust the problem specification to get better prediction models:* Participants were interested in modifying the problem specifications to change the options (e.g., performance metrics such as accuracy, f1-macro, etc. or the target fields) so that they would get more relevant models. The insights generated by visual data exploration can drive the user's refinements of the problem specification.

- *G3: Initially rank the resulting prediction models:* Participants were interested to see the ranking of prediction models based on some criteria, e.g., a performance metric. The ranking should be meaningful to the user, and visualizations of models should help explain the rankings.
- *G4: Determine the most preferable model, beyond the initial rankings:* In many cases, ranking is not enough to make judgment of the superior model. For example, in a classification problem of cancer related data, two models may have the same ranking based on the given accuracy criteria. However, the model with fewer false negative predictions might be preferable. Visualizations can provide an efficient way to communicate the capabilities of different models; even simple visualizations like colored confusion matrices offer much more information than a static metric score.
- *G5: Compare model predictions on individual data points in the context of the input dataset:* Information about the model's predictions, such as their accuracies or their error, were difficult to extrapolate on without the context of individual data instances they predicted upon. Users suggested that having the data overview and the model results on separate tabs of the system made this difficult. Users want to judge model predictions in coordination with exploratory data analysis views. Model explanation techniques such as those linking confusion matrix cells to individual data instances offer a good example of tight linking between the data space and the model space [ZWM*18, ACD*15, RAL*17].
- *G6: Transition seamlessly from one step to another in the overall workflow:* Providing a seamless workflow in the resulting interface helps the user to perform the different tasks required in generating and selecting the relevant models. The system should guide the user in the current task as well as to transition it to the next task without any extra effort. Furthermore, useful default values (such as highly relevant problem specifications or the top ranked predictive model) should be provided for non-expert users so that they can finish at least the default steps in the EMA workflow. Accompanying visualizations that dynamically appear based on the current workflow step can provide easy-to-interpret snapshots of what the system is doing at each step.

It should be noted that our distilled set of requirements does not include participants' comments relating to data cleaning, data augmentation, or post-hoc manual parameters tuning of the selected models. While they are important to the users and relevant to their data analysis needs, these topics are familiar problems in visual analytics systems and are therefore omitted from consideration.

3.3. Workflow Design

Based on the six identified task requirements, we propose a workflow as shown in Figure 1. The workflow consists of seven steps that are then grouped into three high-level tasks: data and problem exploration, model generation, and model exploration and selection. Below we detail each step of the workflow.

Step 1 – Data Exploration: In response to **G1**, we identify data exploration as a required first step. Before a user can explore the model space, they must understand the characteristics of the data. Sufficient information needs to be presented so that the user can make an informed decision as to which types of predictions are suitable for the data. Furthermore, the user needs to be able to identify relevant attributes or subsets of data that should be included (or avoided) in the subsequent modeling process.

Step 2 – Problem Exploration: In response to **G1** and **G2**, we also identify the need of generating automatically a valid set of problem specifications. These problem specifications give the user an idea of the space of potential models, and they can use their understanding of the data from Step 1 to choose which are most relevant to them.

Step 3 – Problem Specification Generation: In response to **G2** and **G3**, we identify the need of generating a valid, machine-readable final set of problem specifications after the user explores the dataset and the automated generated problem specifications set. A EMA visual analytic system needs to provide the option to user to refine and select a problem specification from the system generated set or to add a new problem specification. Furthermore, the user should also be able to provide or edit performance metrics (such as accuracy, F1-score, mean squared root, etc.) for each problem specification.

Step 4 – Model Training and Generation: The generated problem specifications will be used to generate a set of trained models. Ideally, the resulting set of models should be diverse. For example, for a classification problem, models should be generated using a variety of classification techniques (e.g. SVM, random forest, k-means, etc.). Since these techniques have different properties and characteristics, casting a wide net will allow the user to better explore the space of possible predictive models in the subsequent EMA process.

Step 5 – Model Exploration: In response to **G3**, we identify the need of presenting the resulting predictive models in some ranked form (e.g., based on either used performance metric or the time required in generating the model). An EMA visual analytics system needs to present the resulting models through some visualizations, e.g., a confusion matrix for a classification problem type or a residual bar chart for regression problem type (see Fig. 1(5)), so that the user can explore predictions of the models and facillitate comparisons between them. We also identify from **G5** that cross-linking between individual data points in a model and data exploration visualization would be useful for the user to better understand the model. It should be noted that a prerequisite for model exploration is to present models in an interpretable encoding, and the available encoding depends on the types of models being explored. Lipton posited that there are two types of model interpretability: transparency, in which a model can be structurally interpreted, and post-hoc interpretability, in which a model is interpreted via its predictions on held out data [Lip16]. In our workflow, because we aim to allow for any type of model, it is difficult to compare wildly different parts of the model space (a kNN model vs. a deep learning model) based on their structure. Instead, we favor a post-hoc approach, where the models are explored via their predictions.

Step 6 – Model Selection: In response to **G4** and **G5**, we identify the need for selecting the user's preferred models based on the model and data exploration. An EMA visual analytics system needs to provide the option to the user to select one or more preferable models in order to export for later usage.

Step 7 – Export Models: In response to **G4**, we also identify that the user also requires to export the selected preferable models so that they can use them for future predictions.

Finally, we identify from the response of **G6** that an EMA visual analytic system needs to make sure that the transition from one workflow step to another one should be seamless. We assume that any implementation of our proposed EMA workflow in Figure 1 should supply such smooth transitions so that a non-expert user would also be able to finish the workflow from start to end.

4. Iterative System Design and Evaluation

To validate our visual analytics workflow for EMA, we performed two rounds of iterative design and evaluation of the initial prototype system. First, we describe the updated system used in the evaluation. Due to confidentiality concerns, the screenshots shown in this paper use a set of publicly available datasets[‡] that are different from the data examined by the subject matter experts during the two rounds of evaluations.

4.1. Redesigned EMA System

Our redesigned system used for the study significantly differs from the original prototype in three important ways. First, it fully supports the workflow as described in the previous section, including Problem Exploration and Problem Specification Generation. Second, the new system supports 10 types of models (compared to the prototype that only supported two). Lastly, in order to accommodate the diverse subject matter experts' needs, our system was expanded to support a range of input data types. Table 1 lists all of the supported model and data types of our redesigned system.

From a visual interface design standpoint, the new system also appears differently from the prototype. The key reason for the interface redesign is to provide better guidance to users during the EMA process, and to support larger number of data and model types. We realized during the redesign process that the UI of the original prototype (which used tabs for different steps of the analysis) would not scale to meet the requirements of addressing the seven steps of the EMA workflow.

Figure 4 shows screenshots of components of the system that highlight the system's support for guiding the user through the steps of the EMA workflow. The visual interface consists of two parts (see Fig. 4, where we provide the overall system layout in the center). The *workflow-panel* (see Fig. 4(a)), positioned on the left side of the system layout, shows the current level and status of workflow execution. On the right side of the system layout, the *card-panel* consists of multiple *cards* where each card targets a particular step described in the EMA workflow.

Visualization Support for Data Exploration:

For step one of the workflow, **data exploration**, the system renders several cards providing an overview of the dataset. This includes both a dataset summary card containing any metadata available, such as dataset description and source, as well as cards with interactive visualizations for each data type in the dataset (a few examples are provided in Fig. 1(a) and in Fig. 4(b)). Currently, the system supports eight input data types: tabular, graph, time-series, text, image, video, audio, and speech. Datasets are taken in as CSVs containing tabular data that can point to audio or image files, and rows can contain references to other rows, signifying graph linkages. Data types (e.g., numeric, categorical, temporal, or external references) are explicitly provided - the system does no inference of data types. If a dataset contains multiple types of data, the system a specifically designed card for each type of data. In all cases, the user is also provided a searchable, sortable table showing the raw tabular data. All data views are cross-linked to facilitate insight generation. To limit the scope of the experimental system, our system is not responsible for data cleaning or wrangling, and it assumes that these steps have already been done before the system gets the data.

For example, in the case of only tabular data a set of cross-linked histograms are provided (see Fig. 4(b)), empowering the user to explore relationships between features and determine which features are most predictive. Furthermore, a searchable table with raw data fields is also provided. For graph data, node-link diagrams are provided (see Fig. 4(b)). Temporal data is displayed through one or more time-series line charts (see Fig. 4(b)), according to the number of input time-series. For textual data, the system shows a simple searchable collection of documents to allow the user to search for key terms. Image data is displayed in lists sorted by their labels. Audio and speech files are displayed in a grid-list format with amplitude plots, and each file can also be played in the browser through the interface. Video files are also displayed in a grid-list format, and can be played in the browser through the interface as well. In the case of an input dataset with multiple types of data, such as a social media networks where each row in the table references a single node in a graph, visualizations are provided for both types of data (e.g., histograms for table and node-link diagrams for graphs) and are cross-linked via the interaction mechanisms (i.e., brushing and linking). The exact choices for visual encodings for input dataset types are not contributions of this paper, and so mostly standard visualizations and encodings were used.

Problem Specification Generation and Exploration:

After data exploration, the user is presented with a list of possible problem specifications depending on the input dataset (step 2, **problem exploration** in the EMA workflow). This set is autogenerated by first choosing each variable in the dataset as the target variable to be predicted, and then generating a problem specification for each machine learning model type that is valid for that target variable. For example, for a categorical target variable, a classification problem specification is generated. For a numeric target variable, specifications are generated for both regression and collaborative filtering. Table 1 shows the relationships between an input dataset and the possible corresponding model types supported in our system. The system also generates different problem specifications for each metric valid for the problem type and the type of predicted variable (e.g., accuracy, f1 score, precision). Together, the target prediction variable, the corresponding model type, metrics,

[‡] https://gitlab.com/datadrivendiscovery/tests-data

Data Types	Model Types: Classification	Regression	Clustering	Link Prediction	Vertex Nomination	Community Detection	Graph Clustering	Graph Matching	Time Series Forecasting	Collaborative Filtering
Tabular	✔	✔	✔	X	X	X	X	X	X	✔
Graph	✔	✔	✔	✔	✔	✔	✔	✔	X	✔
Time Series	✔	✔	✔	X	X	X	X	X	✔	✔
Texts	✔	✔	✔	X	X	X	X	X	X	✔
Image	✔	✔	✔	X	X	X	X	X	X	✔
Video	✔	✔	✔	X	X	X	X	X	X	✔
Audio	✔	✔	✔	X	X	X	X	X	X	✔
Speech	✔	✔	✔	X	X	X	X	X	X	✔

Table 1: *List of all model types and data types supported by our experimental system. A check mark indicates if a model type can be applied to a particular data type, while a cross mark is used to denote incompatible matching between data and model types.*

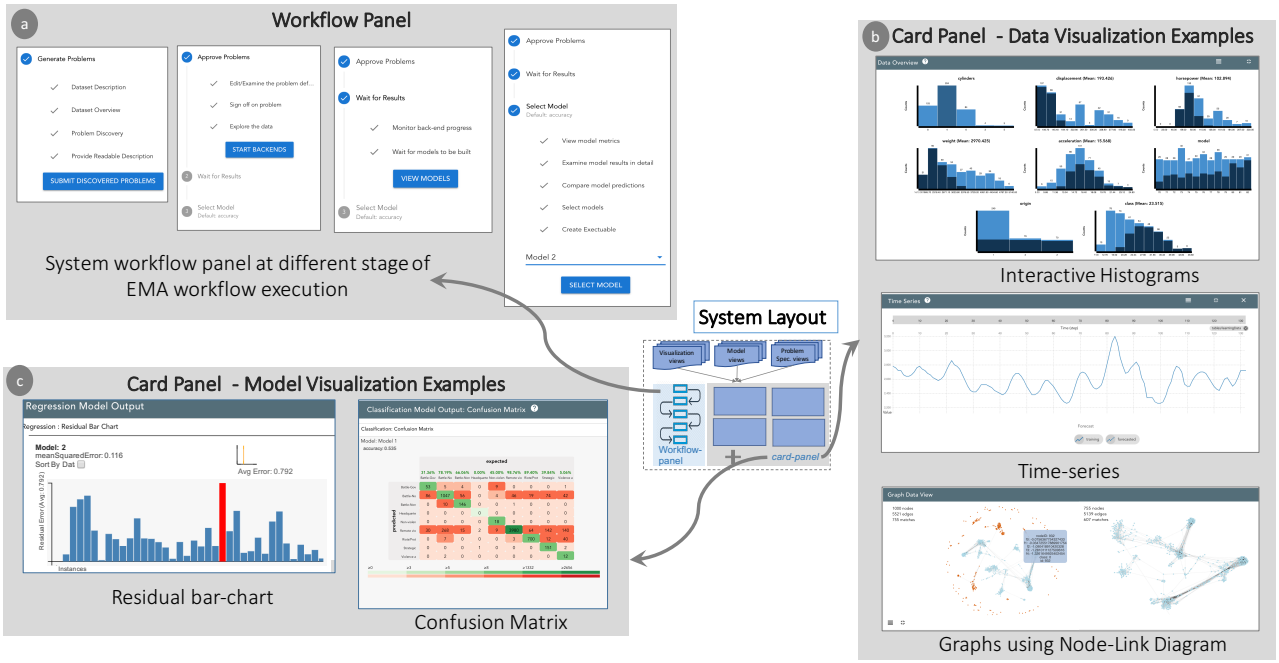

Figure 4: *Components of the experimental system. The box in the center shows the system layout, which consists of two parts, the left-side workflow panel and the right-side card panel. (a) shows EMA workflow at different stages in the experimental system, (b) shows three examples of data visualization cards, and (c) shows two examples of model visualization cards.*

and features to be used for predicting the target prediction variable make up a *problem specification*.

The user can select interesting problem specifications from the system-generated list of recommendations, and refine them by removing features as predictors. Users can also decide to generate their own problem descriptions from scratch, in case non of the system-generated suggestions fit the their goals. In either case, the next step of the EMA workflow is for the user to finalize the **problem specifications** (see Fig. 1(3)). The resulting set of problem specifications is then used by backend autoML systems to generate the corresponding machine learning models.

Visualization Support for Model Exploration and Selection:

Our system's support for **model generation** (step 4 of the EMA workflow) relies on the use of an automated machine learning (autoML) library, developed under the DARPA D3M program [She]. These autoML systems are accessible through an open source API[§] based on the gRPC protocol[¶]. An autoML system requires the user to provide a well-defined problem specification (i.e., the target prediction variable, the model type, the list of features to be used for training the model, the performance metrics) and a set of training data. It then automatically searches through a collection of ML algorithms and their respective hyperparameters, returning the "best" models that fit the user's given problem specification and data. Different autoML libraries such as AutoWeka [THHLB13, KTH*16], Hyperopt [BYC13, KBE14], and Google Cloud AutoML [LL] are in use either commercially or as open source tools. Our system is designed to be compatible with several autoML libraries under the D3M program, including [JSH18, SSW*18]. Note that the sampling of models is entirely driven by the connected autoML systems, and our system does not encode any instructions to the autoML systems beyond the problem specification chosen by the user. However, the backends we connect to generate diverse, complex models, and automatically construct machine learning pipelines including feature extraction and dimensionality reduction steps.

Given the set of problem specifications identified by the user in the previous step, the autoML library automatically generates a list of candidate models. The candidate models are then visualized in an appropriate interpretable representation of their predictions, corresponding to the modeling problem currently being explored by the user (step 5, **model exploration**). All types of classification models, including multiclass, binary, and variants on other types of data such as community detection, are displayed to the user as interactive confusion matrices (see Fig. 4(c)). Regression models and collaborative filtering models are displayed using sortable interactive bar charts displaying residuals (see Fig. 4(c)). Time-series forecasting models are displayed using line charts with dotted lines for predicted points. Cross-linking has been provided between individual data points on these model visualizations and the corresponding attributes in the data exploration visualizations of the input dataset. Furthermore, cross-linking between the models has also been provided to help the user in comparing between the generated models.

Our system initially shows only the highest ranked models produced by the autoML library, as the generated models could be in the hundreds in some cases. This ranking of models is based on the user selected metric in the problem specification.

After a set of suggested models had been generated and returned by the autoML engine, the system provides views to inspect the model's predictions on holdout data. Using this information, they select one or more specific models and request the auotML library to export the selected model(s) (Steps 6 and 7, **model selection** and **export models**).

§ https://gitlab.com/datadrivendiscovery/ta3ta2-api

¶ https://grpc.io/

4.2. Evaluation

To evaluate the validity of our proposed EMA workflow and the efficacy of the prototype system, we conducted two rounds of evaluations. Similar to the feedback study, the participants of these two rounds of evaluation were also recruited by NIST. Five subject matter experts participated in the first round of evaluation, and four participated in the second. One participant in the second round was unable to complete the task due to connectivity issues. None of the experts participated in both studies (and none of them participated in the previous feedback study). The two groups each used different datasets, in an aim to test out the workflow in differing scenarios.

Method: Several days prior to each experiment, participants were part of a teleconference in which the functioning of the system was demonstrated on a different dataset than would be used in their evaluation. They were also provided a short training video [snob] and a user manual [snoa] describing the workflow and individual components of the system they used.

For the evaluation, participants were provided with a link to a web interface through which they would do their EMA. They were asked to complete their tasks without asking for help, but were able to consult the training materials at any point in the process. The modeling specifications discovered by users were recorded, as well as any exported models. After completing their tasks, participants were given an open-ended questionnaire about their experience. After the first round of evaluation, some user feedback was incorporated into the experimental system, and the same experiment was held with different users and a different dataset. All changes made to the experimental system were to solve usability issues, in order to more cleanly enable users to follow the workflow presented in this work.

Tasks: In both evaluation studies, participants were provided with a dataset on which they were a subject matter expert. They were given two successive tasks to accomplish within a 24-hour period. The first task was to explore the given dataset and come up with a set of modeling specifications that interested them. The second task supplied them with a problem specification, and asked them to produce a set of candidate models using our system, explore the candidate models and their predictions, and finally choose one or more models to export with their preference ranking. Their ranking was based on which models they believed would perform the best on held out test data. The two tasks taken together encompass the workflow proposed in this work. The problem specifications discovered by participants were recorded, as well as the resulting models with rankings exported by the participants.

4.3. Findings

All 8 of the participants were able to develop valid modeling problems and export valid predictive models. Participants provided answers to a survey asking for their top takeaways from using the system to complete their tasks. They were also asked if there were additional features that were missing from the workflow of the system. We report common comments on the workflow, eliding comments pertaining to the specific visual encodings used in the system.

Efficacy of workflow: Participants felt that the workflow was successful in allowing them to generate models. One participant noted that the workflow "*...can create multiple models quickly if all (or most data set features are included... [the] overall process of generating to selecting model is generally easy*". Another participant agreed, stating that "*The default workflow containing comparisons of multiple models felt like a good conceptual structure to work in*."

The value of individual stages of the workflow were seen as well: "*The problem discovery phase is well laid out. I can see all the datasets and can quickly scroll through the data*". During this phase, participants appreciated the ability to use the various visualizations in concert with tabular exploration of the data, with one participant stating that "*crosslinking visualizations [between data and model predictions] was a good concept*", and another commenting that the crosslinked tables and visualizations made it "*very easy to remove features, and also simple to select the problem*."

Suggestions for implementations: Participants were asked what features they thought were most important for completing the task using our workflow. We highlight these so as to provide guidance on the most effective ways to implement our workflow, and also to highlight interesting research questions that grow out of tools supporting EMA.

Our experimental system allowed for participants to select which features to use as predictor features (or independent variables) when specifying modeling problems. This led several participants to desire more sophisticated capabilities for feature generation, to "*create new derivative fields to use as features*".

One participant noted that some of the choices in generating problem specifications were difficult to make without first seeing the resulting models, such as the loss function chosen to optimize the model. The participant suggested that, rather than asking the user to provide whether root mean square error or absolute error is used for a regression task, that the workflow "*have the system combinatorically build models for evaluation (for example, try out all combinations of "metric")*". This suggests that the workflow can be augmented by further automating some tasks. For example, some models could be trained before the user becomes involved, to give the user some idea of where to start in their modeling process.

The end goal of EMA is one or more exported models, and several participants noted that documentation of the exported models is often required. One participant suggested the system could "*export the data to include the visualizations in reports*". This suggests that an implementation of our workflow should consider which aspects of provenance it would be feasible to implement, such as those expounded on in work by Ragan et al. [RESC16], in order to meet the needs of data modelers. Another participant noted that further "*understanding of model flaws*" was a requirement, not only for the sake of provenance, but also to aid in the actual model selection. Model understandability is an open topic of research [Gle13], and instance-level tools such as those by Ribeiro et al. [RSG16] and Krause et al. [KDS*17] would seem to be steps in the right direction. Lastly, it was noted that information about how the data was split into training and testing is very relevant to the modeler. Exposing the training/testing split could be critical if there is some property in the data that makes the split important (i.e. if there are seasonal effects in the data).

Limitations of the Workflow: The participants noted that there were some dangers in developing a visual analytics system that enabled exploratory modeling, noting that "*simplified pipelines like those in the task could very easily lead to serious bias or error by an unwary user (e.g. putting together a causally nonsensical model)*". The ethics of building tools that can introduce an untrained audience to new technology is out of the scope of this work, but we do feel the topic is particularly salient in EMA, as the resulting models will likely get deployed in production. We also contend that visual tools, like those supported by our workflow, are preferable to non-visual tools in that the lay user can get a sense of the behavior of models and the training data visually. It could be that additional safeguards should be worked into the workflow to offer a sort of spell-check of the models, similar to how Kindlmann and Scheidegger recommend that visualizations are run through a suite of sanity checks before they are deployed [KS14].

The same participant also noted that streamlining can also limit the ability of the user if they are skilled: "*it doesn't provide sufficient control or introspection... I wanted to add features, customize model structure, etc., but I felt prisoner to a fixed menu of options, as if I was just a spectator*". While some of this can be ameliorated by building a system more angled at the expert user and including more customization options, ultimately the desired capabilities of a system by an expert user may be beyond the ceiling of the workflow we have presented.

5. Usage Scenarios

In this section, we offer two examples of how the system might be used for exploratory model analysis. Through these two scenarios we explain the role of the user during each step of the workflow. The first scenario involves the exploration of a sociological dataset of children's perceptions of popularity and the importance of various aspects of their lives. It is used to build predictive models which can then be incorporated into an e-learning tool.The second scenario requires building predictive models of automobile performance for use in prototyping and cost planning.

5.1. Analyzing the Popular Kids Dataset

The *Popular Kids*[‖] dataset consists of 478 questionnaires of students in grades 4, 5, and 6 about how they perceive importance of various aspects of school in the popularity of their peers. The original study found that among boy respondents, athletic prowess is perceived as most important for popularity, while among girl respondents, appearance is perceived as most important for popularity [CD92].

John works for a large public school district that is trying to determine what data to collect for students on an e-learning platform. Project stakeholders believe that they have some ability to gather data from students in order to personalize their learning plan, but that gathering too much data could lead to disengagement from students. Therefore, John must find what sorts of predictive models can be effective on data that is easy to gather.

‖ http://tunedit.org/repo/DASL

John downloads the Popular Kids dataset and loads it into the application. The system shows him linked histograms of the various features of the dataset, as well as a searchable table. He explores the data (**Step 1 in EMA workflow**), noting that prediction of a student's belief in the importance of grades would be a valuable prediction for the e-learning platform. He scans the list of generated problems (**Step 2**), selecting a regression problem predicting the belief in grades. He refines the problem (**Step 3**, removing variables in the training set of which school the student was from, since that data would not be relevant in his deployment scenario. He then sends this problem to the autoML backend, which returns a set of models (**Step 4**). The system returns to him a set of regression models (**Step 5**), displayed as bar charts showing residuals on held out data (see Figure 6. He notes that none of the regression models have particularly good performance, and in particular, by using cross linking between the regression models and the raw data visualizations, he notes that the resulting models have much more error on girls than on boys.

At this point, John determines that the dataset is not particularly predictive of belief in grades, and decides to search for another predictive modeling problem. He returns to **Step 3** and scans the set of possible problems. He notes that the dataset contains a categorical variable representing the student's goals, with each student marking either *Sports*, *Popular*, or *Grades* as their goal. He chooses a classification problem, predicting student goal, and removes the same variables as before. He submits this new problem and the backend returns a set of models (**Step 4**). The resulting classification models are visualized with a colored confusion matrix, seen in figure 5. John compares the different confusion matrices (**Step 5**), and notes that even though model 2 is the best performing model, it performs poorly on two out of the three classes. Instead, he chooses model 3, which performs farily well on all three classes (**Step 6**). He exports the model (**Step 7**), and is able to use it on data gathered by the e-learning platform.

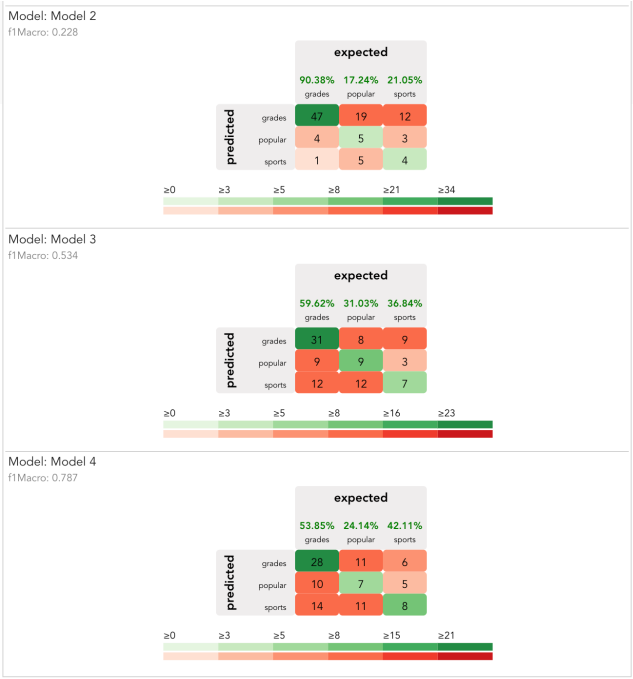

Figure 5: *A set of confusion matrices showing the different classification models predicting Goal of students in the Popular Kids dataset [CD92]. The user is able to see visually that, while the middle model has the highest overall accuracy, it performs much better on students who have high grades as their goal. Instead, the user chooses the bottom model, because it performs more equally on all classes.*

5.2. Modeling Automobile Fuel Efficiency

Erica is a data scientist at an automobile company and she would like to develop predictive models that might anticipate the performance or behavior of a car based on potential configurations of independent variables. In particular, she wants to be able to predict how various designs and prototypes of vehicles might affect properties of the car that affect its sales and cost. She hopes to discover a model that can be used to assess new designs and prototypes of vehicles, before they are built.

Erica has access to a dataset containing information about 398 cars (available from OpenML [VvRBT13]), and she would like to build a set of predictive models using different sets of prediction features to determine which features may be most effective at predicting fuel efficiency. She begins by loading the **Data Source** and explores the relationship between attributes in the histogram view (**Step 1**), shown in the top histogram visualization in Figure 4(b). By hovering over the bars corresponding to mpg, she determines that the number of cylinders and the class may be good predictors. She then explores the system generated set of problem specifications (**Step 2**). She looked on all the generated problem specifications with "class" as predicting feature. She decides on predicting miles per gallon, and selects a regression task. She selects the provided default values for the rest of the problem specification (**Step 3**).

The ML backend trains on the given dataset and generates six models (**Step 4**). Erica starts to explore the generated regression models, visualized through residual bar charts (**Step 5**). The model visualization in this case gives Erica a sense of how the different models apportion residuals by displaying a bar chart of residuals by instance, sorted by the magnitude of residual (see Fig. 6).

Erica notices that the two best models both have similar scores for mean squared error. She views the residual plots for the two best models, and notes that, while the mean squared error of model 4 is lowest, model 5 apportions residuals more evenly among its instances (see Fig. 6). Based on her requirements, it is more important to have a model that gives consistently close predictions, rather than a model that performs well for some examples and poorly for others. Therefore, she selects the model 5 (**Step 6**) to be exported by the system (**Step 7**). By following the proposed EMA workflow, Erica was able to get a better sense of her data, to define a problem, to generate a set of models, and to select the model that she believed would perform best for her task.

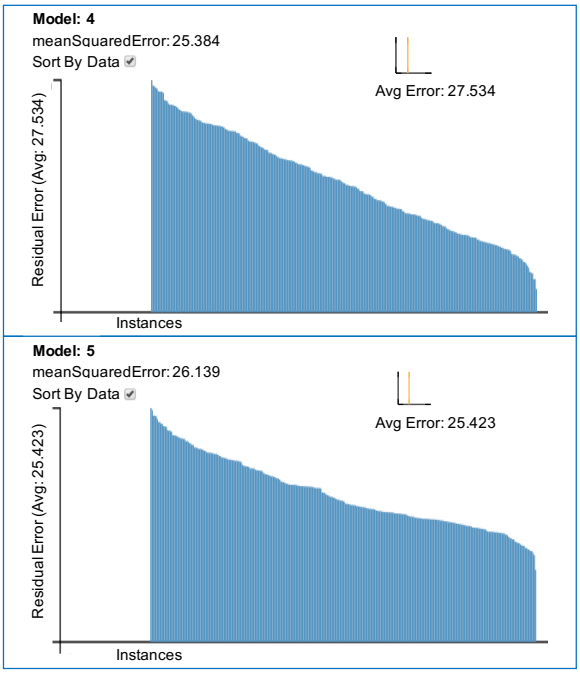

Figure 6: *Regression plots of the two best models returned by a machine learning backend.*

6. Conclusion

In this work, we define the process of exploratory model analysis (EMA), and contribute a visual analytics workflow that supports EMA. We define EMA as the process of discovering and selecting relevant models that can be used to make predictions on a data source. In contrast to many visual analytics tools in the literature, a tool supporting EMA must support problem exploration, problem specification, and model selection in sequence. Our workflow was derived from feedback from a pilot study where participants discovered models on both classification and regression tasks.

To validate our workflow, we built a prototype system and ran user studies where participants were tasked with exploring models on different datasets. Participants found that the steps of the workflow were clear and supported their ability to discover and export complex models on their dataset. Participants also noted distinct manners in which how visual analytics would be of value in implementations of the workflow. We also present two use cases across two disparate modeling scenarios to demonstrate the steps of the workflow. By presenting a workflow and validating its efficacy, this work lays the groundwork for visual analytics for exploratory model analysis through visual analytics.

7. Acknowledgements

We thank our collaborators in DARPA's Data Driven Discovery of Models (d3m) program. This work was supported by National Science Foundation grants IIS-1452977, 1162037, and 1841349, as well as DARPA grant FA8750-17-2-0107.

References

[ACD*15] AMERSHI S., CHICKERING M., DRUCKER S. M., LEE B., SIMARD P., SUH J.: Modeltracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (2015), ACM, pp. 337–346. 4, 6

[AHH*14] ALSALLAKH B., HANBURY A., HAUSER H., MIKSCH S., RAUBER A.: Visual methods for analyzing probabilistic classification data. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 1703–1712. 4

[ALA*] ANDRIENKO N., LAMMARSCH T., ANDRIENKO G., FUCHS G., KEIM D., MIKSCH S., RIND A.: Viewing visual analytics as model building. *Computer Graphics Forum 0*, 0. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13324, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13324, doi:10.1111/cgf.13324. 3, 4

[AWD12] ANAND A., WILKINSON L., DANG T. N.: Visual pattern discovery using random projections. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on* (2012), IEEE, pp. 43–52. 4

[BISM14] BREHMER M., INGRAM S., STRAY J., MUNZNER T.: Overview: The design, adoption, and analysis of a visual document mining tool for investigative journalists. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 2271–2280. 5

[BJY*18] BILAL A., JOURABLOO A., YE M., LIU X., REN L.: Do convolutional neural networks learn class hierarchy? *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 152–162. 4

[BLBC12] BROWN E. T., LIU J., BRODLEY C. E., CHANG R.: Dis-function: Learning distance functions interactively. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on* (2012), IEEE, pp. 83–92. 4

[BOH11] BOSTOCK M., OGIEVETSKY V., HEER J.: D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 12 (2011), 2301–2309. 3

[BYC13] BERGSTRA J., YAMINS D., COX D. D.: Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in Science Conference* (2013), pp. 13–20. 9

[CD92] CHASE M., DUMMER G.: The role of sports as a social determinant for children. *Research Quarterly for Exercise and Sport 63* (1992), 18–424. 10, 11

[CD19] CAVALLO M., DEMIRALP Ã.: Clustrophile 2: Guided visual clustering analysis. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2019), 267–276. 4

[CG16] CHEN M., GOLAN A.: What may visualization processes optimize? *IEEE Transactions on Visualization and Computer Graphics 22*, 12 (2016), 2619–2632. 2, 3

[Chu79] CHURCH R. M.: How to look at data: A review of john w. tukey's exploratory data analysis 1. *Journal of the experimental analysis of behavior 31*, 3 (1979), 433–440. 3

[CLKP10] CHOO J., LEE H., KIHM J., PARK H.: ivisclassifier: An interactive visual analytics system for classification based on supervised dimension reduction. In *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on* (2010), IEEE, pp. 27–34. 4

[CLL*13] CHOO J., LEE H., LIU Z., STASKO J., PARK H.: An interactive visual testbed system for dimension reduction and clustering of large-scale high-dimensional data. In *Visualization and Data Analysis 2013* (2013), vol. 8654, International Society for Optics and Photonics, p. 865402. 4

[CMS99] CARD S. K., MACKINLAY J. D., SHNEIDERMAN B.: *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999. 3

[Cou18] COUNCIL OF EUROPEAN UNION: 2018 reform of eu data protection rules, 2018. https://ec.europa.eu/commission/priorities/justice-and-fundamental-rights/data-protection/2018-reform-eu-data-protection-rules_en. 4

[CPMC17] CASHMAN D., PATTERSON G., MOSCA A., CHANG R.: Rnnbow: Visualizing learning via backpropagation gradients in recurrent neural networks. In *Workshop on Visual Analytics for Deep Learning (VADL)* (2017). 4

[CR98] CHI E. H.-H., RIEDL J. T.: An operator interaction framework for visualization systems. In *Information Visualization, 1998. Proceedings. IEEE Symposium on* (1998), IEEE, pp. 63–70. 3

[CT05] COOK K. A., THOMAS J. J.: Illuminating the path: The research and development agenda for visual analytics. 2

[CZGR09] CHANG R., ZIEMKIEWICZ C., GREEN T. M., RIBARSKY W.: Defining insight for visual analytics. *IEEE Computer Graphics and Applications 29*, 2 (2009), 14–17. 2

[DCCE18] DAS S., CASHMAN D., CHANG R., ENDERT A.: Beames: Interactive multi-model steering, selection, and inspection for regression tasks. *Symposium on Visualization in Data Science* (2018). 2, 4

[DOL03] DE OLIVEIRA M. F., LEVKOWITZ H.: From visual data exploration to visual data mining: a survey. *IEEE Transactions on Visualization and Computer Graphics 9*, 3 (2003), 378–394. 3

[EFN12] ENDERT A., FIAUX P., NORTH C.: Semantic interaction for sensemaking: inferring analytical reasoning for model steering. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (2012), 2879–2888. 4

[Fek04] FEKETE J.-D.: The infovis toolkit. In *Information Visualization, IEEE Symposium on* (2004), IEEE, pp. 167–174. 3

[FTC09] FIEBRINK R., TRUEMAN D., COOK P. R.: A meta-instrument for interactive, on-the-fly machine learning. In *New Interfaces for Musical Expression* (2009), pp. 280–285. 4

[Gle13] GLEICHER M.: Explainers: Expert explorations with crafted projections. *IEEE Transactions on Visualization and Computer Graphics*, 12 (2013), 2042–2051. 4, 10

[Gle18] GLEICHER M.: Considerations for visualizing comparison. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 413–423. 4

[HBO*10] HEER J., BOSTOCK M., OGIEVETSKY V., ET AL.: A tour through the visualization zoo. *Communications of ACM 53*, 6 (2010), 59–67. 3

[HG18] HEIMERL F., GLEICHER M.: Interactive analysis of word vector embeddings. In *Computer Graphics Forum* (2018), vol. 37, Wiley Online Library, pp. 253–265. 4

[HKBE12] HEIMERL F., KOCH S., BOSCH H., ERTL T.: Visual classifier training for text document retrieval. *IEEE Transactions on Visualization and Computer Graphics*, 12 (2012), 2839–2848. 4

[HME00] HOAGLIN D. C., MOSTELLER F., (EDITOR) J. W. T.: *Understanding Robust and Exploratory Data Analysis*, 1 ed. Wiley-Interscience, 2000. 3

[Hun07] HUNTER J. D.: Matplotlib: A 2d graphics environment. *Computing in science & engineering 9*, 3 (2007), 90–95. 3

[JSH18] JIN H., SONG Q., HU X.: Efficient neural architecture search with network morphism. *arXiv preprint arXiv:1806.10282* (2018). 9

[JZF*09] JEONG D. H., ZIEMKIEWICZ C., FISHER B., RIBARSKY W., CHANG R.: ipca: An interactive system for pca-based visual analytics. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 767–774. 4

[KAF*08] KEIM D., ANDRIENKO G., FEKETE J.-D., GÖRG C., KOHLHAMMER J., MELANÇON G.: Information visualization. Springer-Verlag, Berlin, Heidelberg, 2008, ch. Visual Analytics: Definition, Process, and Challenges, pp. 154–175. URL: http://dx.doi.org/10.1007/978-3-540-70956-5_7, doi:10.1007/978-3-540-70956-5_7. 2

[KBE14] KOMER B., BERGSTRA J., ELIASMITH C.: Hyperopt-sklearn: automatic hyperparameter configuration for scikit-learn. In *ICML workshop on AutoML* (2014). 9

[KDS*17] KRAUSE J., DASGUPTA A., SWARTZ J., APHINYANAPHONGS Y., BERTINI E.: A workflow for visual diagnostics of binary classifiers using instance-level explanations. *Visual Analytics Science and Technology (VAST), IEEE Conference on* (2017). 4, 10

[KEV*18] KWON B. C., EYSENBACH B., VERMA J., NG K., DE FILIPPI C., STEWART W. F., PERER A.: Clustervision: Visual supervision of unsupervised clustering. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 142–151. 4

[KKE10] KEIM E. D., KOHLHAMMER J., ELLIS G.: Mastering the information age: Solving problems with visual analytics, eurographics association, 2010. 3

[KLTH10] KAPOOR A., LEE B., TAN D., HORVITZ E.: Interactive optimization for steering machine classification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2010), ACM, pp. 1343–1352. 4

[KMSZ06] KEIM D. A., MANSMANN F., SCHNEIDEWIND J., ZIEGLER H.: Challenges in visual data analysis. In *Information Visualization, 2006. IV 2006. Tenth International Conference on* (2006), IEEE, pp. 9–16. 2

[KS14] KINDLMANN G., SCHEIDEGGER C.: An algebraic process for visualization design. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 2181–2190. 10

[KTB*18] KUMPF A., TOST B., BAUMGART M., RIEMER M., WESTERMANN R., RAUTENHAUS M.: Visualizing confidence in cluster-based ensemble weather forecast analyses. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 109–119. 4

[KTH*16] KOTTHOFF L., THORNTON C., HOOS H. H., HUTTER F., LEYTON-BROWN K.: Auto-weka 2.0: Automatic model selection and hyperparameter optimization in weka. *Journal of Machine Learning Research 17* (2016), 1–5. 9

[LD11] LLOYD D., DYKES J.: Human-centered approaches in geovisualization design: Investigating multiple methods through a long-term case study. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2498–2507. 5

[Lip16] LIPTON Z. C.: The mythos of model interpretability. *arXiv preprint arXiv:1606.03490* (2016). 6

[LL] LI F.-F., LI J.: Cloud automl: Making ai accessible to every business. https://www.blog.google/topics/google-cloud/cloud-automl-making-ai-accessible-every-business/. Accessed: 2018-03-29. 9

[LSC*18] LIU M., SHI J., CAO K., ZHU J., LIU S.: Analyzing the training processes of deep generative models. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 77–87. 4

[LSL*17] LIU M., SHI J., LI Z., LI C., ZHU J., LIU S.: Towards better analysis of deep convolutional neural networks. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2017), 91–100. 4

[LWLZ17] LIU S., WANG X., LIU M., ZHU J.: Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics 1*, 1 (2017), 48–56. 3

[LWT*15] LIU S., WANG B., THIAGARAJAN J. J., BREMER P.-T., PASCUCCI V.: Visual exploration of high-dimensional data through subspace analysis and dynamic projections. In *Computer Graphics Forum* (2015), vol. 34, Wiley Online Library, pp. 271–280. 4

[LXL*18] LIU S., XIAO J., LIU J., WANG X., WU J., ZHU J.: Visual diagnosis of tree boosting methods. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 163–173. 4

[MLMP18] MÜHLBACHER T., LINHARDT L., MÖLLER T., PIRINGER H.: Treepod: Sensitivity-aware selection of pareto-optimal decision

trees. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 174–183. 2, 4

[NHM*07] NAM E. J., HAN Y., MUELLER K., ZELENYUK A., IMRE D.: Clustersculptor: A visual analytics tool for high-dimensional data. In *2007 IEEE Symposium on Visual Analytics Science and Technology* (2007), pp. 75–82. 4

[NM13] NAM J. E., MUELLER K.: Tripadvisor^{ND}: A tourism-inspired high-dimensional space exploration framework with overview and detail. *IEEE Transactions on Visualization and Computer Graphics 19*, 2 (2013), 291–305. 4

[Nor06] NORTH C.: Toward measuring visualization insight. *IEEE Computer Graphics and Applications 26*, 3 (2006), 6–9. 2

[PBD*10] PATEL K., BANCROFT N., DRUCKER S. M., FOGARTY J., KO A. J., LANDAY J.: Gestalt: integrated support for implementation and analysis in machine learning. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology* (2010), ACM, pp. 37–46. 4

[pow] Power BI. https://powerbi.microsoft.com/. Accessed: 2018-12-12. 3

[PS08] PERER A., SHNEIDERMAN B.: Integrating statistics and visualization: case studies of gaining clarity during exploratory data analysis. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (2008), ACM, pp. 265–274. 3

[RAL*17] REN D., AMERSHI S., LEE B., SUH J., WILLIAMS J. D.: Squares: Supporting interactive performance analysis for multiclass classifiers. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2017), 61–70. 4, 6

[RESC16] RAGAN E. D., ENDERT A., SANYAL J., CHEN J.: Characterizing provenance in visualization and data analysis: an organizational framework of provenance types and purposes. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2016), 31–40. 10

[RSG16] RIBEIRO M. T., SINGH S., GUESTRIN C.: Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016), ACM, pp. 1135–1144. 4, 10

[SGB*18] STROBELT H., GEHRMANN S., BEHRISCH M., PERER A., PFISTER H., RUSH A. M.: Seq2seq-vis: A visual debugging tool for sequence-to-sequence models. *IEEE Transactions on Visualization and Computer Graphics* (2018). 2, 4

[SGPR18] STROBELT H., GEHRMANN S., PFISTER H., RUSH A. M.: Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 667–676. 4

[SHB*14] SEDLMAIR M., HEINZL C., BRUCKNER S., PIRINGER H., MÖLLER T.: Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics*, 99 (2014). 4

[She] SHEN W.: Data-driven discovery of models (d3m). https://www.darpa.mil/program/data-driven-discovery-of-models. Accessed: 2018-03-24. 9

[Shn96] SHNEIDERMAN B.: The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on* (1996), IEEE, pp. 336–343. 3

[SKB*18] SACHA D., KRAUS M., BERNARD J., BEHRISCH M., SCHRECK T., ASANO Y., KEIM D. A.: Somflow: Guided exploratory cluster analysis with self-organizing maps and analytic provenance. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 120–130. 4

[SKKC19] SACHA D., KRAUS M., KEIM D. A., CHEN M.: Vis4ml: An ontology for visual analytics assisted machine learning. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2019), 385–395. 3

[SMWH17] SATYANARAYAN A., MORITZ D., WONGSUPHASAWAT K., HEER J.: Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2017), 341–350. 3

[snoa] d3m Snowcat Training Manual. http://www.eecs.tufts.edu/~dcashm01/public_img/d3m_manual.pdf. Accessed: 2019-03-30. 9

[snob] d3m Snowcat Training Video. https://youtu.be/_JC3XM8xcuE. Accessed: 2019-03-30. 9

[SPH*16] SIEVERT C., PARMER C., HOCKING T., CHAMBERLAIN S., RAM K., CORVELLEC M., DESPOUY P.: plotly: Create interactive web graphics via plotly.js. *R package version 3*, 0 (2016). 3

[spo] spotfire. https://www.tibco.com/products/tibco-spotfire. Accessed: 2018-12-12. 3

[SSS*14] SACHA D., STOFFEL A., STOFFEL F., KWON B. C., ELLIS G., KEIM D. A.: Knowledge generation model for visual analytics. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 1604–1613. 2

[SSW*18] SHENI G., SCHRECK B., WEDGE R., KANTER J. M., VEERAMACHANENI K.: Prediction factory: automated development and collaborative evaluation of predictive models. *arXiv preprint arXiv:1811.11960* (2018). 9

[SSZ*16] SACHA D., SEDLMAIR M., ZHANG L., LEE J. A., WEISKOPF D., NORTH S. C., KEIM D. A.: Human-Centered Machine Learning Through Interactive Visualization: Review and Open Challenges. In *Proceedings of the 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, Belgium* (Apr. 2016). 3

[tab] Tableau. https://www.tableau.com/. Accessed: 2018-12-12. 3

[THHLB13] THORNTON C., HUTTER F., HOOS H. H., LEYTON-BROWN K.: Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (2013), ACM, pp. 847–855. 9

[Tuk77] TUKEY J. W.: *Exploratory data analysis*, vol. 2. Reading, Mass., 1977. 2, 3

[Tuk93] TUKEY J. W.: *Exploratory data analysis: past, present and future*. Tech. rep., PRINCETON UNIV NJ DEPT OF STATISTICS, 1993. 3

[VDEvW11] VAN DEN ELZEN S., VAN WIJK J. J.: Baobabview: Interactive construction and analysis of decision trees. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on* (2011), IEEE, pp. 151–160. 4

[VvRBT13] VANSCHOREN J., VAN RIJN J. N., BISCHL B., TORGO L.: Openml: Networked science in machine learning. *SIGKDD Explorations 15*, 2 (2013), 49–60. URL: http://doi.acm.org/10.1145/2641190.2641198, doi:10.1145/2641190.2641198. 11

[VW05] VAN WIJK J. J.: The value of visualization. In *Visualization, 2005. VIS 05. IEEE* (2005), IEEE, pp. 79–86. 2, 3

[WC*08] WICKHAM H., CHANG W., ET AL.: ggplot2: An implementation of the grammar of graphics. *R package version 0.7, URL: http://CRAN.R-project.org/package=ggplot2* (2008). 3

[WLS*10] WEI F., LIU S., SONG Y., PAN S., ZHOU M. X., QIAN W., SHI L., TAN L., ZHANG Q.: Tiara: a visual exploratory text analytic system. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (2010), ACM, pp. 153–162. 4

[WLSL17] WANG J., LIU X., SHEN H.-W., LIN G.: Multi-resolution climate ensemble parameter analysis with nested parallel coordinates plots. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2017), 81–90. 4

[WZM*16] WANG X.-M., ZHANG T.-Y., MA Y.-X., XIA J., CHEN W.: A survey of visual analytic pipelines. *Journal of Computer Science and Technology 31* (2016), 787–804. 3

[YCN*15] YOSINSKI J., CLUNE J., NGUYEN A. M., FUCHS T. J., LIPSON H.: Understanding neural networks through deep visualization. *CoRR abs/1506.06579* (2015). URL: http://arxiv.org/abs/1506.06579. 4

[ZWM*18] ZHANG J., WANG Y., MOLINO P., LI L., EBERT D. S.: Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models. *IEEE Transactions on Visualization and Computer Graphics* (2018). 4, 6