TELEGAM: Combining Visualization and Verbalization for Interpretable Machine Learning

Fred Hohman[†],[*] Georgia Institute of Technology
Arjun Srinivasan[‡],[*] Georgia Institute of Technology
Steven M. Drucker[§] Microsoft Research

ABSTRACT

While machine learning (ML) continues to find success in solving previously-thought hard problems, interpreting and exploring ML models remains challenging. Recent work has shown that visualizations are a powerful tool to aid debugging, analyzing, and interpreting ML models. However, depending on the complexity of the model (e.g., number of features), interpreting these visualizations can be difficult and may require additional expertise. Alternatively, textual descriptions, or verbalizations, can be a simple, yet effective way to communicate or summarize key aspects about a model, such as the overall trend in a model's predictions or comparisons between pairs of data instances. With the potential benefits of visualizations and verbalizations in mind, we explore how the two can be combined to aid ML interpretability. Specifically, we present a prototype system, TELEGAM, that demonstrates how visualizations and verbalizations can collectively support interactive exploration of ML models, for example, generalized additive models (GAMs). We describe TELEGAM's interface and underlying heuristics to generate the verbalizations. We conclude by discussing how TELEGAM can serve as a platform to conduct future studies for understanding user expectations and designing novel interfaces for interpretable ML.

Index Terms: Human-centered computing—Visual Analytics

1 INTRODUCTION

While machine learning (ML) continues to find success in solving previously-thought hard problems with data, its pitfalls, such as encoding and perpetuating cultural and historical data bias inside complex models [6–8], have been the subjects of critical discussion surrounding its appropriate and ethical use [2,5]. In fact, governmental policy has been put in place, giving people a "right to explanation" for any model prediction that could impact their financial or legal status [30]. To understand how models learn and behave, interpretable, or explainable, artificial intelligence (AI) research has seen intense focus and progress [14]. Within this field, interactive data visualization has been used as a medium for communicating explanations for both models and predictions, allowing data scientists to better understand and debug their models [3, 17, 26, 27].

Previous work has shown that data scientists explain model results continuously to other groups of people: management, technical peers, and other stakeholders with invested interest in an ML model or product [16]. However, often at the core of explainable ML sits an inherent trade-off between the completeness and simplicity of an explanation. To explain a single data instance's prediction from a complex model with hundreds of features often requires significant effort that could consist of creating and interpreting many visualizations. These explanatory visualizations can also require high graphicacy, i.e., visualization literacy, from the people that create and use them for model iteration and decision making; this results in significant time and effort needed to understand what a visualization is showing and what is most important.

Alternatively, text or natural language has also been used as a medium to communicate model results and predictions [33,37]. Text is useful for providing short, approximate explanations that provide most of the necessary information to understand a prediction without the cognitive burden of digesting a visualization. Natural language explanations could complement explanatory visualizations by helping people identify or verify inferences derived from a chart, identify prediction contributions they might have missed, or emphasize differences between predictions for multiple instances. However, systems combing both visual and natural language explanations for ML models remain largely underexplored. In this work, we investigate how system generated natural language explanations, or "verbalizations," can complement explanatory visualizations. Such interfaces that combine visualizations and verbalizations could help data scientists better understand and debug their ML models, and aid them when communicating modeling results to other stakeholders. For example, systems could present verbalizations that are related to but not immediately observable in a visualization to help data scientists pivot between visualizations exploring different aspects of their models (e.g., global model-level explanations vs. local instance-level predictions) or drill-down into a model's performance for specific instances (e.g., comparing predictions for two data instances).

To explore such possibilities, we extend recent work by Hohman et al. [16] on operationalizing model interpretability and contribute a prototype system demonstrating the potential of combining visualization and verbalization for explaining ML results. The system, TELEGAM, automatically generates natural language statements, or verbalizations, to complement explanatory visualizations for generalized additive models (GAMs). Incorporating the increasingly popular notion of interactively linking text and visualizations [19, 21, 22, 38], TELEGAM also lets users interact with verbalizations to visually manifest them in the explanatory visualization through simple annotations. By doing so, TELEGAM demonstrates how interfaces could better help data scientists fluidly understand and explain models along a completeness-simplicity spectrum and serve as a starting point for model analysis. TELEGAM can be accessed at: `https://poloclub.github.io/telegam/`.

2 RELATED WORK

Although a formal definition for interpretability has not yet been attained nor agreed upon [11,23], we can distinguish *interpretability* (synonymous with explainability) from an *explanation*. An explanation is a collection of features from an interpretable domain that relate a data instance to a model's outcome [28, 32]. Explanations can be truthful or deceptive, accurate or inaccurate; therefore, using multiple explanations can better guide people to gain an ultimate interpretation of a model. Recent research postulates that explanation resolution, i.e., the sophistication or completeness of an explanation, depends on the audience [13, 16, 32]. Model builders may prefer global, aggregate model explanations to address the generalizability of a model; model users may prefer local, specific instance predictions to assist decision-making. Both explanation paradigms will

[†] e-mail: fredhohman@gatech.edu

[*] Authors contributed equally

[‡] e-mail: arjun010@gatech.edu
[§] e-mail: sdrucker@microsoft.com

impact the interpretability of a system; in this work we support both global and local paradigms, as well as offer an interactive affordance that allows users to dynamically update the resolution of a verbalization to tailor the level of detail desired in explanation.

To better support collaborative intelligence sharing between people and ML systems [39], visual analytics has succeeded in helping diverse populations of people interact with ML [17,26,27,35,40]. Example visualizations and tasks include leveraging unit visualizations for interactive model debugging and performance analysis [4, 31], using diverging bar charts to show feature importance [32], and interacting with partial dependence line charts to allow data scientists to understand counterfactual outcomes for specific instances [20, 29].

While visualizations are powerful tools to help people better understand ML models, they may not be sufficient, and depending on a user's background, they can also be challenging to interpret. Recent work has begun to conjecture whether complementing visualizations with verbalizations can enhance model explanations. For instance, Sevastjanova et al. [37] present a design space discussing strategies for model explanation generation and presentation at the intersection of visualizations and verbalizations. In their design space, we focus on supporting *post-hoc interpretability* where an explanation uses the relationship between the input and output of a model instead of the model's inner mechanisms [28]. Specifically, following a strategy similar to recent visualization tools that systematically extract "data facts" to highlight potentially interesting observations in visualizations [10, 12, 38], we heuristically analyze the data associated with model-level and instance-level visualizations, and present them as textual statements alongside visualizations. In other words, we adopt an *overview and detail strategy* [37] for generating explanations where visualizations are used to give an overview while the verbalizations highlight specific features or trends. Furthermore, we also interactively link visualizations and verbalizations, supporting *details-on-demand* when presenting explanations [37].

3 TELEGAM: VISUALIZATION & VERBALIZATION

3.1 Design Goals

Through a literature survey and formative studies with ML researchers and practitioners, Hohman et al. [16] synthesize six model-agnostic capabilities that an explainable ML interface should support. In this work, we focus on four of these and use them as design goals (**DG**) for our system. The design goals below each contain an example interpretability question, which all reference a real-estate model that predicts the price of homes given the features of a house.

DG1. Local instance explanations. Given a single data instance, quantify each feature's contribution to the prediction.
Example: Given a house and its predicted price of $250,000, what features contributed to its price?

DG2. Instance explanation comparisons. Given a collection of data instances, compare what factors lead to their predictions.
Example: Given five houses in a neighborhood, what distinguishes them and their prices?

DG3. Feature importance. Given a model, rank the features of the data that are most influential to the overall predictions.
Example: Given a house price prediction model, does it make sense that the top three most influential features should be the square footage, year built, and location?

DG4. Counterfactuals. Given a single data instance, ask "what-if" questions to observe the effect that modified features have on its prediction.
Example: Given a house and its predicted price of $250,000, how would the price change if it had an extra bedroom?

3.2 Model Class and Background

In this work we consider a particular model class, the *generalized additive model* (GAM) [15], which has recently attracted attention in the ML community [1, 34], and satisfies our four **DG**s. Modern ML techniques have enabled GAMs to compete favorably with more complex, state-of-the-art models on tabular data prediction tasks; however, GAMs remain intelligible and more expressive than simple linear models [9, 24, 25]. A GAM provides both local instance explanations similar to linear regression, but also global feature explanations which other models lack.

GAMs are a generalization of linear models; GAMs replace linear model's slope coefficients with smooth, shape functions. In both models, the relationship between the target variable and the features is still additive; however, each feature in a GAM is described by one shape function that can be nonlinear and complex (e.g., concave, convex, or "bendy") [18]. Therefore, GAMs are considered intelligible [9] since each feature's contribution to the final prediction can be understood by inspecting the shape functions. In this paper, we omit the technical details and mathematical formulations of GAMs and their training, which are covered in the literature [24, 25, 36, 41].

3.3 Realizing Design Goals in TELEGAM's Interface

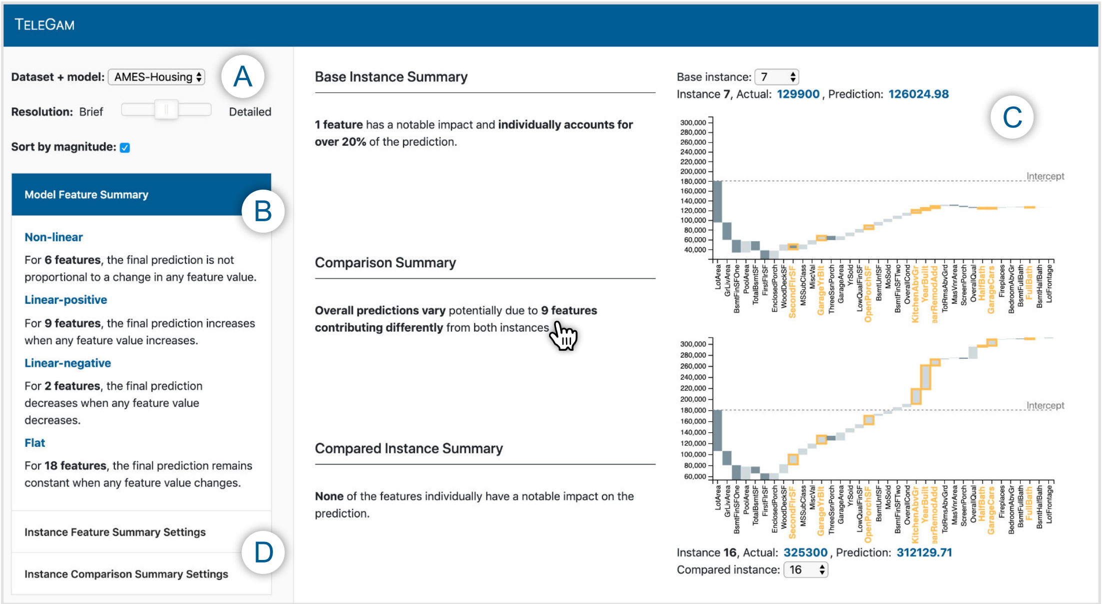
We first give an overview of TELEGAM's interface (Figure 1), deferring the details of the verbalizations to the next section. When a model is loaded (Figure 1A), the Global Model View (Figure 1B) displays sentences highlighting the features that may be interesting for the user to consider (**DG3**). Brushing over sentences displays a tooltip (Figure 2) showing the GAM shape function line charts that present an overview of the feature values (on the x-axis) and model predictions (on the y-axis) corresponding to the features listed in the sentence. These visualizations also enable a user to ask counterfacutals, i.e., "what if" questions, by quantifying the increase or decrease of predictions based on a change in any feature value (**DG4**).

Selecting an instance from the dropdown in the Local Instance View (Figure 1C) displays the actual and predicted values for the instance. TELEGAM presents a waterfall chart similar to that in GAMUT [16] where the features are listed on the x-axis and the contribution to the prediction from each feature are represented by the height of the bars. The color of the bar indicates whether the contribution is positive (**light gray**) or negative (**dark gray**). By default, the features are sorted by the absolute magnitude of their contributions, i.e., the feature with the highest absolute contribution is shown on the left. A toggle is present (Figure 1A) to sort features by their actual contributions instead of their absolute contributions if desired. To summarize the contributions of features towards an instance's prediction (**DG1**), along with displaying a waterfall chart, TELEGAM also presents a textual summary alongside the chart. Brushing over this sentence visually highlights the notable features and their corresponding bars in the waterfall chart in **orange**, as seen in Figure 1C. This also is useful for asking counterfactual questions by identifying which features could be changed to increase or decrease an instance's final prediction (**DG4**).

With a base instance selected (Figure 1C, top), users can select a second instance for comparison (Figure 1C, bottom). To enable visual comparison, TELEGAM ensures the scale of the y-axis as well as the ordering of the features on the x-axis in both waterfall charts are normalized and consistent. In addition to showing a textual summary for each instance (Figure 1C: top-left, bottom-left), TELEGAM also generates a comparative summary highlighting the differences between the predictions, displaying possible features that may cause the difference (**DG2**). Similar to the individual instance summaries, brushing the comparative summary visually highlights the described features in the visualization **orange** (Figure 1C).

3.4 Generating Verbalizations

Following the design goals, TELEGAM presents three types of verbalizations to accompany feature and instance-level visualizations. We converged to these types of verbalizations based on interactions with participants during GAMUT's user study [16] as well as other



Dataset + model: AMES-Housing

Resolution: Brief — Detailed

Sort by magnitude: ☑

Model Feature Summary

Non-linear

For **6 features**, the final prediction is not proportional to a change in any feature value.

Linear-positive

For **9 features**, the final prediction increases when any feature value increases.

Linear-negative

For **2 features**, the final prediction decreases when any feature value decreases.

Flat

For **18 features**, the final prediction remains constant when any feature value changes.

Instance Feature Summary Settings

Instance Comparison Summary Settings

Base Instance Summary

1 feature has a notable impact and **individually accounts for over 20%** of the prediction.

Comparison Summary

Overall predictions vary potentially due to **9 features contributing differently** from both instances.

Compared Instance Summary

None of the features individually have a notable impact on the prediction.

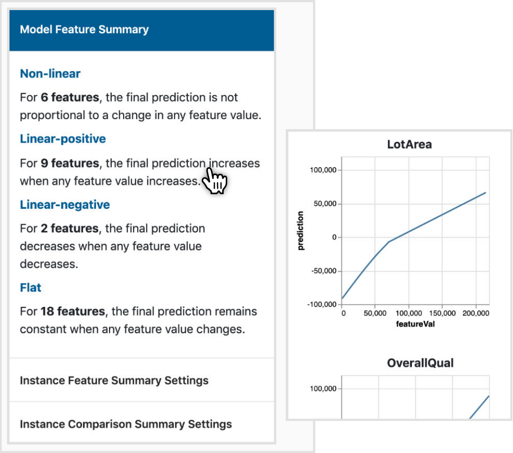
Base instance: 7

Instance **7**, Actual: **129900** , Prediction: **126024.98**

Instance **16**, Actual: **325300** , Prediction: **312129.71**

Compared instance: 16

Figure 1: The TeleGam user interface contains (**A**) a model selector and parameters for the visualizations and verbalizations. (**B**) **The Global Model View** displays model feature-level verbalizations of GAM shape function charts that describe a feature's overall impact on model predictions. (**C**) **The Local Instance View** displays two data instance's waterfall charts, an explanatory visualization that shows the cumulative sum of the contribution each feature has on the final prediction. Alongside are instance-level verbalizations that, when brushed, highlight in **orange**) the corresponding marks of the visualization that the verbalization refers to. (**D**) Settings to interactively tune verbalization generation thresholds.



Model Feature Summary

Non-linear

For **6 features**, the final prediction is not proportional to a change in any feature value.

Linear-positive

For **9 features**, the final prediction increases when any feature value increases.

Linear-negative

For **2 features**, the final prediction decreases when any feature value decreases.

Flat

For **18 features**, the final prediction remains constant when any feature value changes.

Instance Feature Summary Settings

Instance Comparison Summary Settings

Figure 2: In TeleGam, brushing a model feature verbalization displays a tooltip with features' corresponding shape function charts, a common GAM visualization. For example, here, the contribution of the linear-positive feature **LotArea** on overall model predictions approximately constantly increases as the feature value increases.

data scientists who frequently interact with GAMs. Specifically, we considered and collated comments with respect to communicating a model's performance to different stakeholders. Then, using an iterative trial-and-error approach, we defined a set of heuristics to generate a set of verbalizations that were common across the observations. Note that the current set of verbalizations are only an initial step towards exploring how visualizations and verbalizations can be integrated in the context of GAMs, and are not exhaustive.

Instance Feature Summary. For an individual instance, TeleGam verbalizes the features that have a notable impact on the its final prediction. To generate this verbalization, we first compute the ratio of each feature's contribution with respect to the total prediction. If the ratio is greater than a predefined threshold τ_{contrib}, then that feature is included in the verbalization. In other words, a feature x_i is included in the verbalization if $f(x_i)/y > \tau_{\text{contrib}}$, where $f(x_i)$ is the feature's GAM prediction contribution and y is the final instance prediction. We empirically set τ_{contrib} to 0.15. For example, for the base instance in Figure 1C top, only one feature (**LotArea**) is included in the verbalization because it is the only feature that has a contribution of over 0.15 (or 15%) towards the instance's prediction.

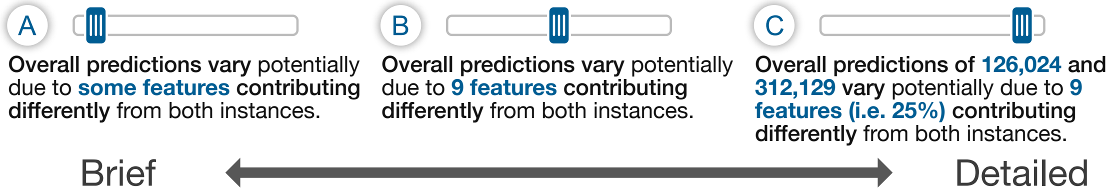
Instance Comparison Summary. When verbalizing comparisons between two instances, TeleGam identifies how similar, or different, the predictions for the instances are while highlighting which features may be contributing to the prediction difference. To do so, we first normalize both the total predictions and the individual feature contributions for all instances to $[0-1]$ so the comparison can be made in context of the entire dataset. Then, we check for the differences between the considered pair of normalized predictions and compare them to preset thresholds (τ_{minDiff}, τ_{minDiff}) to generate the verbalization. Specifically, given two data instances and their normalized predictions y_1 and y_2, the predictions are considered:

$$\begin{cases} \text{too similar} & \text{if } |y_1 - y_2| < \tau_{\text{minDiff}}, \\ \text{too different} & \text{else if } |y_1 - y_2| > \tau_{\text{maxDiff}}, \\ \text{moderately varying} & \text{else.} \end{cases}$$

where τ_{minDiff} and τ_{maxDiff} are empirically set to 0.25 and 0.75. For the second half of the verbalization, a feature is considered accountable for the difference between the final predictions of two instances if for any feature x_i their normalized feature contributions $f(x_{1,i})$ and $f(x_{2,i})$ satisfy

$$|f(x_{1,i}) - f(x_{2,i})| > \tau_{\text{featureContrib}}$$

where $\tau_{\text{featureContrib}}$ is empirically set to 0.25. For example, in Figure 1C, the verbalization states "*overall predictions vary*" because, in the context of the dataset, the two instances have moderately differing predictions which may be because of "*9 features contributing differently*," since the normalized differences between the predictions for those features was over 0.25.



Overall predictions vary potentially due to some features contributing differently from both instances.

Overall predictions vary potentially due to 9 features contributing differently from both instances.

Overall predictions of 126,024 and 312,129 vary potentially due to 9 features (i.e. 25%) contributing differently from both instances.

Figure 3: TeleGam supports an initial interactive affordance to realize the simplicity-completeness explanation spectrum in an interface. As a user drags the slider, the resolution of the natural language explanation updates from "brief" to "detailed." In the example above, the comparison summary for two instances is shown at three different levels of explanation resolution, including **(A)** brief, **(B)** default, and **(C)** detailed.

Model Feature Summary. TeleGam highlights four groups of feature-level verbalizations based on the overall geometry of the shape function line charts—namely, features that have positively linear, negatively linear, non-linear, or flat geometry. To identify these groups, we used an agglomerative hierarchical clustering approach: a bottom-up technique for clustering data; in this case, we represent the shape function line charts in Figure 2 as time series and cluster them based on their overall geometry. We then inspected and labeled the clusters as the four groups listed above. Since some features may have expected predictions that are typically linear (e.g., the predicted price of a house increases with its square footage), these high-level groups and their corresponding verbalizations help users focus on features that are potentially more interesting (e.g. those with with non-linear geometry) while still summarizing every feature.

3.5 User-specified Verbalization Resolution

Professional data scientists have different reasons to interpret models and tailor explanations for specific audiences, often balancing competing concerns of simplicity and completeness [13,16,32]. Previous work has also suggested interfaces where users could specify the resolution of presented explanations; this can help adapt to users with differing preferences or expertise levels [37]. TeleGam supports an initial interactive affordance to realize this simplicity-completeness tradeoff spectrum. Located in Figure 1A, a slider adjusts how detailed verbalizations should be. Currently, there are three positions ranging from "brief" to "detailed." As a user drags the slider from one end to the other, the verbalizations update to provide more (or less) detail about a data instance's prediction.

For example, Figure 3 shows three different slider positions for verbalization summarization for the comparison of two instance predictions. When set to "Brief" (Figure 3A), the verbalization is composed only of text to describe the difference between the two instance's predictions. Dragging the slider right, in the second position (Figure 3B), the sentence updates and displays the exact number of features that the two predictions differ in. Finally, in the "Detailed" position (Figure 3C), the sentence updates and lists the actual prediction values, the number of differing features, and what percent of the total features the instances differ in. This is only one realized example of how a system could provide explanations based on user-specified resolutions to better communicate results to differing stakeholders invested in an ML model.

3.6 Illustrative Usage Scenario

We now demonstrate how the different views of TeleGam could be used to interpret a GAM through a hypothetical usage scenario. June is a data scientist at a real-estate firm exploring the available properties to gain insight into the company's portfolio. As June loads a pre-trained model into TeleGam to understand the data and predictions, the system automatically displays textual statements summarizing the major feature trends (Figure 1B). By interacting with these statements, June explores how the different features are represented inside the model (Figure 2). Next, recollecting a property (data instance) they recently visited (*id=7*) but did not sell despite it being affordable, June inspects it as the base instance (Figure 1C, top). Based on their understanding of the individual feature trends from Global Model View and through the visualization in the Local Instance View, they infer that the *LotArea* feature is the primary factor determining the property's value. The text alongside the chart simultaneously confirms this inference (Figure 1C, top-left).

To understand potential factors that make a house more saleable, June compares the selected property to another (*id=16*) that recently sold although it was more expensive. Through a combination of the juxtaposed waterfall charts and the verbalizations comparing the two charts, June notes that the differences in price arise from multiple non-salient features (e.g., *OpenPorchSF*, *SecondFlrSF*) that they would have otherwise missed without a visual linking between the text and the charts (Figure 1B). Adjusting the comparison verbalization resolution (Figure 3), TeleGam further reveals the specifics of the contributing feature quantity and distribution differences. Finally, to prepare a report to share with their colleagues, June sets the detail level of the verbalizations to "Brief" and captures a screenshot, moving onto other instances and continuing their analysis.

4 Discussion and Future Work

Through the design of TeleGam, we show how combining visualizations and verbalizations can support interactive exploration and interpretation of ML models, demonstrated using GAMs. TeleGam is only an initial step; our broader goal is to understand if verbalizations can enhance interpretability by augmenting ML visualization tools with explanations.

An immediate next step is to invite data scientists to use TeleGam, investigating (1) how well the verbalizations summarize the different aspects of the model, and (2) if the combination of visualizations and verbalizations aid interpretability and help people ask or answer new types of questions. Based on this study feedback, we hope to refine TeleGam's verbalization generation, possibly exploring additional verbalizations to help users identify regions of error (i.e., parts of the model that produce highly uncertain predictions) [16] or methods to increase its transparency.

Complementing visualizations with verbalizations also opens up new avenues for interactive exploration of ML models. For example, by treating verbalizations as search targets, systems can more easily allow people to use natural language queries to search for particular types of data instances based on patterns they exhibit. In other words, when looking at an instance's prediction, one could simply type "*show instances with similar predictions*" and the system could directly compare such a query against the possible verbalizations to identify the similar instances. However, exploring the feasibility and practical utility of such interface affordances in the context of a more complete workflow involving model building and model evaluation remains an open question for future work. Ultimately, we hope that this work will help further realize the emerging idea of combining visualizations and verbalizations in the context of ML systems and encourage the design of future interfaces for explanation.

Acknowledgments

This work is supported in part by a NASA Space Technology Research Fellowship.

References

[1] InterpretML. *Microsoft*, 2019.

[2] People + ai guidebook: Designing human-centered ai products. *Google People + AI Research (PAIR)*, 2019.

[3] A. Abdul, J. Vermeulen, D. Wang, B. Y. Lim, and M. Kankanhalli. Trends and trajectories for explainable, accountable and intelligible systems: an hci research agenda. In *ACM Conference on Human Factors in Computing Systems*, p. 582. ACM, 2018.

[4] S. Amershi, M. Chickering, S. M. Drucker, B. Lee, P. Simard, and J. Suh. Modeltracker: redesigning performance analysis tools for machine learning. In *ACM Conference on Human Factors in Computing Systems*, pp. 337–346. ACM, 2015.

[5] S. Amershi, D. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, et al. Guidelines for human-ai interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2019.

[6] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias. *ProPublica, May*, 23, 2016.

[7] J. Buolamwini and T. Gebru. Gender shades: intersectional accuracy disparities in commercial gender classification. In *ACM Conference on Fairness, Accountability and Transparency*, pp. 77–91, 2018.

[8] A. Caliskan, J. J. Bryson, and A. Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

[9] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. Intelligible models for healthcare: predicting pneumonia risk and hospital 30-day readmission. In *ACM International Conference on Knowledge Discovery and Data Mining*, pp. 1721–1730. ACM, 2015.

[10] Ç. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati. Foresight: Recommending visual insights. *Proceedings of the VLDB Endowment*, 10(12):1937–1940, 2017.

[11] F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[12] K. Eckelt, P. Adelberger, T. Zichner, A. Wernitznig, and M. Streit. Tourdino: A support view for confirming patterns in tabular data. *EuroVis Workshop on Visual Analytics*, 2019.

[13] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal. Explaining explanations: an approach to evaluating interpretability of machine learning. *arXiv preprint arXiv:1806.00069*, 2018.

[14] D. Gunning. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA)*, 2017.

[15] T. J. Hastie and R. Tibshirani. Generalized additive models. In *Chapman & Hall/CRC*. 1990.

[16] F. Hohman, A. Head, R. Caruana, R. DeLine, and S. M. Drucker. Gamut: A design probe to understand how data scientists understand machine learning models. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2019.

[17] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau. Visual analytics in deep learning: an interrogative survey for the next frontiers. *IEEE Transactions on Visualization and Computer Graphics*, 2018.

[18] K. Jones and S. Almond. Moving out of the linear rut: the possibilities of generalized additive models. *Transactions of the Institute of British Geographers*, pp. 434–447, 1992.

[19] N. Kong, M. A. Hearst, and M. Agrawala. Extracting references between text and charts via crowdsourcing. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pp. 31–40. ACM, 2014.

[20] J. Krause, A. Perer, and K. Ng. Interacting with predictions: visual inspection of black-box machine learning models. In *ACM Conference on Human Factors in Computing Systems*, pp. 5686–5697. ACM, 2016.

[21] B. C. Kwon, F. Stoffel, D. Jäckle, B. Lee, and D. Keim. Visjockey: Enriching data stories through orchestrated interactive visualization. In *Poster Compendium of the Computation+ Journalism Symposium*, vol. 3, 2014.

[22] S. Latif, D. Liu, and F. Beck. Exploring interactive linking between text and visualization. In *Proceedings of the Eurographics/IEEE VGTC Conference on Visualization: Short Papers*, pp. 91–94. Eurographics Association, 2018.

[23] Z. C. Lipton. The mythos of model interpretability. *ICML Workshop on Human Interpretability in Machine Learning*, 2016.

[24] Y. Lou, R. Caruana, and J. Gehrke. Intelligible models for classification and regression. In *ACM International Conference on Knowledge Discovery and Data Mining*, pp. 150–158. ACM, 2012.

[25] Y. Lou, R. Caruana, J. Gehrke, and G. Hooker. Accurate intelligible models with pairwise interactions. In *ACM International Conference on Knowledge Discovery and Data Mining*, pp. 623–631. ACM, 2013.

[26] J. Lu, W. Chen, Y. Ma, J. Ke, Z. Li, F. Zhang, and R. Maciejewski. Recent progress and trends in predictive visual analytics. *Frontiers of Computer Science*, 11(2):192–207, 2017.

[27] Y. Lu, R. Garcia, B. Hansen, M. Gleicher, and R. Maciejewski. The state-of-the-art in predictive visual analytics. In *Computer Graphics Forum*, vol. 36, pp. 539–562. Wiley Online Library, 2017.

[28] G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

[29] G. PAIR. What-if tool. 2018.

[30] Parliament and C. of the European Union. General data protection regulation. 2016.

[31] D. Ren, S. Amershi, B. Lee, J. Suh, and J. D. Williams. Squares: supporting interactive performance analysis for multiclass classifiers. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):61–70, 2017.

[32] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: explaining the predictions of any classifier. In *ACM International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144. ACM, 2016.

[33] M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[34] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206, 2019.

[35] D. Sacha, M. Sedlmair, L. Zhang, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. What you see is what you can change: human-centered machine learning by interactive visualization. *Neurocomputing*, 268:164–175, 2017.

[36] D. Servén and C. Brummitt. pygam: generalized additive models in python, Mar. 2018. doi: 10.5281/zenodo.1208723

[37] R. Sevastjanova, F. Beck, B. Ell, C. Turkay, R. Henkin, M. Butt, D. A. Keim, and M. El-Assady. Going beyond visualization: Verbalization as complementary medium to explain machine learning models. In *Workshop on Visualization for AI Explainability at IEEE VIS*, 2018.

[38] A. Srinivasan, S. M. Drucker, A. Endert, and J. Stasko. Augmenting visualizations with interactive data facts to facilitate interpretation and communication. *IEEE transactions on visualization and computer graphics*, 25(1):672–681, 2018.

[39] S. Stumpf, V. Rajaram, L. Li, W.-K. Wong, M. Burnett, T. Dietterich, E. Sullivan, and J. Herlocker. Interacting meaningfully with machine learning systems: three experiments. *International Journal of Human-Computer Studies*, 67(8):639–662, 2009.

[40] D. S. Weld and G. Bansal. Intelligible artificial intelligence. *arXiv preprint arXiv:1803.04263*, 2018.

[41] S. N. Wood. *Generalized additive models: an introduction with R*. Chapman and Hall/CRC, 2006.