Data Science with LLMs and Interpretable Models

Sebastian Bordt[1]*, Ben Lengerich[2], Harsha Nori[3], Rich Caruana[3]

[1]University of Tübingen, Tübingen AI Center
[2]Massachusetts Institute of Technology, Broad Institute of MIT and Harvard
[3]Microsoft Research
sebastian.bordt@uni-tuebingen.de, blengeri@mit.edu, hanori@microsoft.com, rcaruana@microsoft.com

Abstract

Recent years have seen important advances in the building of interpretable models, machine learning models that are designed to be easily understood by humans. In this work, we show that large language models (LLMs) are remarkably good at working with interpretable models, too. In particular, we show that LLMs can describe, interpret, and debug Generalized Additive Models (GAMs). Combining the flexibility of LLMs with the breadth of statistical patterns accurately described by GAMs enables dataset summarization, question answering, and model critique. LLMs can also improve the interaction between domain experts and interpretable models, and generate hypotheses about the underlying phenomenon. We release https://github.com/interpretml/TalkToEBM as an open-source LLM-GAM interface.

Introduction

Recent years have seen important advances in the building of interpretable models (Caruana et al. 2015; Lin et al. 2020). On tabular machine learning problems, interpretable glass-box models are now frequently on par with black box methods (Rudin 2019). In other words, today's interpretable models act as powerful pattern detectors.

In this work, we show that combining interpretable models with large language models (LLMs) holds broad potential to improve many different tasks in data science. Using an interpretable model allows LLMs to handle complex statistical patterns, enabling them to perform some data science tasks semi-autonomously (Hollmann, Müller, and Hutter 2023). For example, we can ask the LLM whether the statistical relationships learned by an interpretable model are surprising given what would be expected from prior knowledge in the underlying real-world problem.

We explore this approach with the interpretable model class of Generalized Additive Models (GAMs) (Hastie and Tibshirani 1990; Caruana et al. 2015). GAMs represent complex outcomes as a sum of univariate component functions (graphs). This means that the model is additive, and we can make use of this additivity to scale our approach to large datasets with many features. By analyzing the graphs of a GAM one at a time, the LLM can analyze large models

with many different features, without ever having the entire model fit into the context window (Liu et al. 2023). This is the advantage of using interpretable models: they have short textual description lengths. In contrast, black-box models are monolithic and have long textual descriptions, prohibiting their use as contexts for LLMs.[1]

In addition to serving as a useful tool for data science, the combination of LLMs with interpretable models also holds the potential to improve domain expert's interaction with machine learning. This is because it provides domain experts with textual access to the models, and allows them to formulate questions to the model in natural language (Slack et al. 2022).

This paper is organized as follows. We first provide a brief overview of our approach. We then provide simple baseline evaluations (is the LLM able to read a value from a graph?), and continue with examples of more complex qualitative tasks (summarization, anomaly detection). We then discusses the problem of hallucination, and presents some simple experimental evidence on the grounding of model responses in the interpretable model.

Our Approach

The short technical description of our approach is as follows. We first train a GAM. Then we convert the individual graphs of the GAM to text, representing them as key-value pairs in JSON. The graphs are then provided as context to an LLM, together with a general introductory prompt about the task, and a description of the dataset. We engage in chain-of-thought (CoT) prompting (Wei et al. 2022), first asking the LLM to describe the general pattern of the graph, then asking more specific questions (these questions depend on the task that we want the LLM to perform). We also request succinct summarizations of entire conversations about graphs. To elevate the discussion to the level of the entire model and dataset, the summaries of the conversations about individual graphs are aggregated and provided as context in a new

---

---

[1]The textual description length of the GAM on the pneumonia dataset is 43,592 GPT-4 tokens, too many to be directly encoded in GPT-4's 32k context window. The individual graphs are much more compact: the maximum description length of a single graph in the pneumonia GAM is 2,345 GPT-4 tokens. This easily fits within GPT-4's context window.
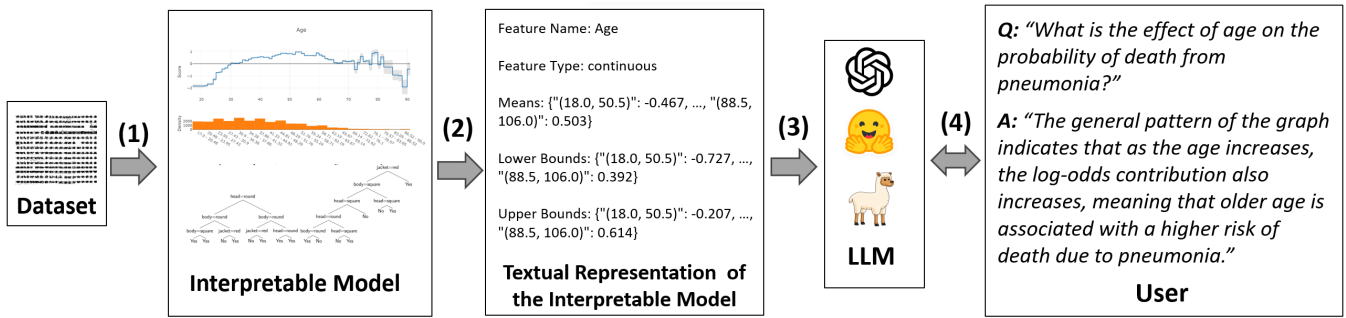
Figure 1: Conceptual overview of our approach. (1) We first train an interpretable model. (2) Next, we convert the different components of the interpretable model to text. (3) Then we provide these components as inputs to an LLM. (4) The user asks the LLM questions about the model or the dataset. The LLM can provide model-level summaries, or answer specific questions about particular aspects of the model. The LLM can also be used to automatically generate hypotheses about the real-world phenomenon that underlies the model and data.

query to the LLM. A graphical overview of this approach is given in Figure 1. The task of describing an individual graph is depicted in Figure 2. The prompt structure is given in the Supplement.

Experimental Results

This sections provides the results of different experiments. We begin by investigating the ability of GPT-3.5 and GPT-4 to perform baseline tasks with graphs. Performance on these tasks can be quantified. This leads us to the conclusion that our approach requires relatively powerful LLMs. We then demonstrate the impressive abilities of GPT-4 to perform complex qualitative tasks with GAMs.

Datasets. We use the California Housing (Kelley Pace and Barry 1997), OpenML Diabetes (Smith et al. 1988; Vanschoren et al. 2014), Iris, Kaggle Titanic[2], Kaggle Spaceship Titanic[3], and Pneumonia (Caruana et al. 2015) dataset. The Spaceship Titanic dataset is from 2022, and the Pneumonia dataset has never been released on the Internet, meaning that GPT-3.5 and GPT-4 have not seen these datasets during training (the pneumonia dataset has been described in papers).

Interpretable Models. We train Generalized Additive Models (GAMs) using the `interpretml` software package (Nori et al. 2019).

LLMs. We perform our experiments with `gpt-3.5-turbo-16k-0613` and `gpt-4-0613`, using the OpenAI API (OpenAI 2023).

Baseline Tasks

We conduct the following experiments using 31 different graphs, obtained from training GAMs on the different datasets.

Reading a Value from a Graph. As a test of whether the LLM is able to perform basic tasks with the `JSON` format in which we provide the graph, we ask the LLM to read a value from the graph ("What is the mean value of the graph

at 3.528?"). This means that we sample a random point on the x-axis and ask the LLM about the value of the graph at that point. The results of this experiment are depicted in Table 1. GPT-4 correctly reads 64 out of 75 values from the different graphs. In fact, the success rate of GPT-4 on some graphs is much higher, and the mistakes frequently occur on graphs that require many decimal places to encode. GPT-3.5, in contrast, is not able to perform this simple task (of course, this can also depend on prompting).

Deciding Monotonicity. One of the most important properties that a graph in an interpretable model can have is monotonicity. If a graph is monotone, it would be important to include this in a summarization of the graph, and later the summary of the full model. We test the ability of GPT-3.5 and GPT-4 to decide whether a graph is monotone ("Please tell me whether the graph is monotone increasing, monotone decreasing, or not monotone."). By using both the original and y-axis inverted graphs, we conduct this experiment with a total of 62 different graphs (if the original graph is monotone increasing, then the y-axis inverted graph is monotone decreasing). The result of this experiment is depicted in the second row of Table 1. GPT-4 correctly identifies 11 of 12 monotone graphs. It also claims that 9 out of 50 non-monotone graphs are monotone. However, most of these graphs 9 are "honest mistakes", meaning that they are almost, but not perfectly, monotone. We thus find that GPT-4 has a surprisingly strong ability to decide the monotonicity of a graph purely from its textual representation. GPT-3.5, in contrast, is hardly able to identify the monotone graphs.

Finding the Largest Jump. Inspired by the strong performance of GPT-4 on the previous two tasks, we decided to ask the model to identify the largest jump in a graph. The graphs of EBMs are piecewise constant, which means that they have jumps in between the different intervals (Nori et al. 2019). The learned jumps often have real-world interpretations (Lengerich et al. 2022). However, this task is challenging for the LLMs since a perfect solution requires taking the max absolute value of the subtraction of adjacent values (Dziri et al. 2023). The result of the experiment is depicted in the third column of Table 1. We ask the LLMs to

---

[2]https://www.kaggle.com/competitions/titanic

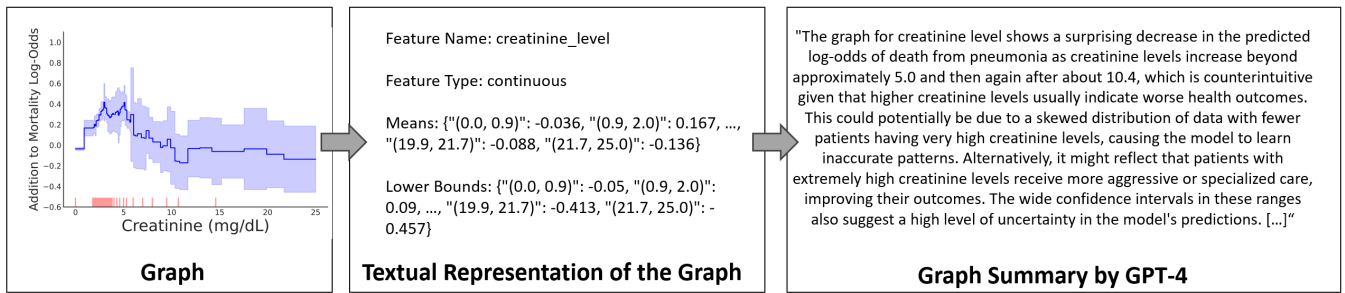[3]https://www.kaggle.com/c/spaceship-titanic

Figure 2: The basic building block of our framework is the ability of LLMs to describe and summarize the individual graphs of Generalized Additive Models (GAMs). Considering a GAM one graph at a time allows the LLM to work with interpretable models even on large-scale datasets with hundreds of features while staying within confined context windows.

identify the biggest jumps in our 31 different graphs. Using chain-of-thought prompting (first listing a number of important jumps in the graph, then listing the largest one) GPT-4 is able to achieve an impressive success rate of 27 correctly identified jumps out of a total of 31 graphs. Again, the number of jumps identified by GPT-3.5 is much smaller.

Qualitative Tasks

In the previous Section, we saw that GPT-4 shows good performance on simple tasks with graphs. In this Section, we provide different examples that demonstrate the impressive capabilities of GPT-4 on more complex qualitative tasks.

Summarizing Graphs. A basic building block of our framework is the ability of GPT-4 to describe and summarize individual graphs. Here we ask the model to describe the influence of the categorical feature "Home Planet" in the fictional Spaceship Titanic task from Kaggle. The model predicts the probability of a passenger "being transported to an alternative dimension".

> **GPT-4:** *"The HomePlanet feature has a significant influence on the probability of a passenger being transported to an alternate dimension during the collision with the spacetime anomaly. Passengers from Europa have the highest positive effect, indicating a higher likelihood of being transported. On the other hand, passengers from Earth have a negative effect, suggesting a lower likelihood of being transported. [...]"*

Summarizing Models. Based on the ability of LLMs to describe individual graphs, we can use them to describe entire models. For this task, we additionally provide the LLM with the global feature importances. Consider the following description of a GAM trained to predict the likelihood of death from pneumonia (Caruana et al. 2015).

> **GPT-4:** *"The model predicts the likelihood of death from pneumonia based on several patient features. The most influential global feature is age, with increasing age associated with a higher likelihood of death, particularly after the age of 73. Respiration rate and BUN level also have substantial influences on the model's predictions, with higher levels of both associated with an increased risk of death.*

Table 1: Performance of GPT-3.5 and GPT-4 on baseline tasks with graphs. The table depicts the number of cases where the model succeeded versus the total number of cases in the task.

	GPT-3.5	GPT-4
Reading a Value from a Graph	3/75	64/75
Deciding Monotonicity	2/12	11/12
Finding the Largest Jump	9/31	27/31

> *Other significant features include the number of diseases a patient has, their systolic blood pressure, and their pO2 levels. More diseases, lower systolic blood pressure, and very low or very high pO2 levels are all associated with a higher likelihood of death.*
>
> *The model also shows that certain medical readings can increase or decrease the likelihood of death, depending on their levels. For example, both very low and very high glucose levels, pH levels, and potassium levels can increase the risk of death, while moderate levels are associated with a lower risk.*
>
> *Finally, the model shows a considerable amount of uncertainty in its predictions for certain features, such as age, heart rate, and albumin levels. This is likely due to a lack of data points in certain ranges or the complex, non-linear relationships these features may have with the outcome. [...]"*

This description was generated using the approach outlined above. Any aspects mentioned in the description are entirely correct and grounded in the actual model. (We discuss the problem of hallucination in the next Section).

Anomaly Detection. In order to show how the combination of LLMs and interpretable models can be useful for practitioners, we demonstrate the ability of GPT-4 at anomaly detection. In this task, we ask the LLM to critique the model and find any aspects that might be surprising or anti-causal (Lengerich et al. 2022). We again use the pneumonia model and ask the LLM to find surprising patterns in the graph for the feature "creatinine level".

> **GPT-4:** *The graph for creatinine level shows a sur-*

prising decrease in the predicted log-odds of death from pneumonia as creatinine levels increase beyond approximately 5.0 and then again after about 10.4, which is counterintuitive given that higher creatinine levels usually indicate worse health outcomes. This could potentially be due to a skewed distribution of data with fewer patients having very high creatinine levels, causing the model to learn inaccurate patterns. Alternatively, it might reflect that patients with extremely high creatinine levels receive more aggressive or specialized care, improving their outcomes. The wide confidence intervals in these ranges also suggest a high level of uncertainty in the model's predictions. [...]"

A clinician rated this response as accurate and potentially useful. For a more detailed analysis of how our approach can be useful for anomaly and surprise detection in healthcare, see (Lengerich et al. 2023).

Grounding and Hallucination

In this Section, we conduct an empirical investigation of the problem of hallucination. Our approach is to modify the learned graphs in a way that is counterfactual to that found in the original data. We then ask the LLM to describe both the original and counterfactual graph. In the absence of hallucination, the description of the counterfactual graph should reflect the modifications. We first discuss results with individual graphs. We then discuss the task of describing a counterfactual model.

Describing a Counterfactual Graph

We first discuss the example of the "Sex" feature on the Kaggle Titanic dataset. The reason why we chose this example because the LLM has likely learned about the role of sex in emergency evacuations during pre-training, which might lead it to ignore the graph and provide a generic response. Here is GPT-4 describing the graph of the "Sex" feature in a GAM that models the survival probability on the Titanic.

GPT-4: *"The 'Sex' feature significantly influenced the survival outcome in the Titanic disaster. Being female greatly increased the chances of survival, as indicated by a positive mean value of 1.397. [..] the overall trend suggests that females had a higher survival rate compared to males, aligning with the historical context of the "women and children first" maritime evacuation policy."*

We can see how the LLM provides an interpretation within the historical context. Now, we flip the values of 'male' and 'female' in the model, and see how the description of the LLM changes.

GPT-4: *"Being male contributes positively to the log-odds of survival, with a mean contribution of 1.397, while being female contributes negatively [...] it should be noted that these do not represent raw survival probabilities. This counterintuitive finding may be influenced by other confounding variables in the*

dataset. Further investigation is needed to fully understand this output."

We observe that the model response changes in accordance with the graph. Interestingly, the LLM also notes that the result is counterintuitive, even if it does not directly explain why this is the case (which would be even better and might be achieved with better prompting).

Describing a Counterfactual Model

We now discuss an experiment where GPT-4 describes a counterfactual model. We use the popular Iris dataset. The reason why we chose this dataset is because GPT-4 has seen it during pre-training (Bordt, Nori, and Caruana 2023), meaning that GPT-4's responses could potentially be based on the original data and not on our interpretable model. We first ask GPT-4 to describe a GAM that models the probability of a flower being Iris-setosa.

GPT-4: *"[...] Iris-setosa flowers typically have shorter and narrower petals compared to other species. [...] As sepal length increases beyond 5.55 units, the likelihood of a flower being Iris-setosa decreases. However, Iris-setosa flowers are more likely to have wider sepals, with the likelihood increasing drastically for widths greater than 2.85 units. Each feature exhibits an abrupt change in contribution at specific thresholds, suggesting potential distinct cutoffs that differentiate Iris-setosa from other species. [...]*

We then invert the graphs of the GAM on the y-axis and again pretend to the LLM that this GAM predicts the probability of a flower being Iris-setosa. Here is the description of the modified model.

GPT-4: *"[...] The model suggests that Iris Setosa is more likely to have longer sepals, smaller sepal widths, longer petal lengths, and wider petals. Notably, there are threshold effects at certain intervals for all the features. [...] Further validation with expert biological knowledge or additional data would be beneficial."*

We observe that the model summarization by the LLM changes in accordance with the model.

In summary, while the evidence from our simple qualitative experiments is far from conclusive, they do demonstrate that the responses of GPT-4 are frequently grounded in the provided interpretable model.

Related Work

Slack et al. (Slack et al. 2022) develop a natural language interface to give practitioners conversational access to model explanations. Their work is similar in spirit to ours but intends to provide access to generic (potentially black-box) models, precluding the LLM from direct access to model internals. Bisercic et al. (Bisercic et al. 2023) showed that LLMs can extract tabular datasets from unstructured text and then train interpretable models (linear regression and small decision trees) on top of this data. Various recent works have

explored the potential of LLMs for data science with tabular data by data wrangling and cleaning (Narayan et al. 2022; Vos, Döhmen, and Schelter 2022), or traditional supervised and unsupervised learning tasks like classification or density estimation (Borisov et al. 2022; Hegselmann et al. 2023; Wang et al. 2023; Onishi, Oono, and Hayashi 2023).

Previous work with GAMs has shown the ubiquity of hidden confounding in real-world datasets and the utility of GAMs to reveal surprising effects in datasets (Caruana et al. 2015; Lengerich et al. 2022). This prior work relied on human experts to design explicit statistical tests for surprising patterns in GAM feature graphs, limiting the potential applications.

Discussion

In this work we have explored synergies between LLMs and interpretable models and found this to be a powerful combination with broad potential to assist data scientists and practitioners.

As a preliminary investigation, we explored the capabilities of GPT-3.5 and GPT-4 to perform a number of different tasks with graphs generated by glass-box models. Although we gave only limited effort to prompt tuning, the results already are reasonably strong and probably could be further improved with more advanced prompting strategies. We observe that it would be unclear how to give an LLM the same degree of access to a black-box model. In principle, however, a related approach might be possible with post-hoc methods, for example, Shapley Values, which are known to be closely related to GAMs (Lundberg and Lee 2017; Bordt and von Luxburg 2023).

Our preliminary experiments raise a variety of avenues of future study. For example, how does LLM performance change with graph complexity? Under what conditions is the LLM response grounded in the graph, and when does hallucination occur?

It is important to note that, given the current state of the technology, there are no guarantees that an answer from an LLM is a truthful response to the prompt. For this reason, all the answers and descriptions provided by the LLMs should be seen as helpful suggestions rather than final answers. Similarly, the answers provided by GPT-4 in the anomaly detection tasks should be seen as useful hypothesis that have to be verified by human experts.

All in all, we believe that the approach of combining LLMs with interpretable models holds broad potential to assist practitioners and domain experts across a broad range of tasks.

References

Bisercic, A.; Nikolic, M.; van der Schaar, M.; Delibasic, B.; Lio, P.; and Petrovic, A. 2023. Interpretable Medical Diagnostics with Structured Data Extraction by Large Language Models. *arXiv preprint arXiv:2306.05052*.

Bordt, S.; Nori, H.; and Caruana, R. 2023. Elephants Never Forget: Testing Language Models for Memorization of Tabular Data. In *NeurIPS 2023 Second Table Representation Learning Workshop*.

Bordt, S.; and von Luxburg, U. 2023. From Shapley values to generalized additive models and back. In *International Conference on Artificial Intelligence and Statistics*, 709–745. PMLR.

Borisov, V.; Seßler, K.; Leemann, T.; Pawelczyk, M.; and Kasneci, G. 2022. Language models are realistic tabular data generators. *arXiv preprint arXiv:2210.06280*.

Caruana, R.; Lou, Y.; Gehrke, J.; Koch, P.; Sturm, M.; and Elhadad, N. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 1721–1730.

Dziri, N.; Lu, X.; Sclar, M.; Li, X. L.; Jian, L.; Lin, B. Y.; West, P.; Bhagavatula, C.; Bras, R. L.; Hwang, J. D.; et al. 2023. Faith and Fate: Limits of Transformers on Compositionality. *arXiv preprint arXiv:2305.18654*.

Hastie, T. J.; and Tibshirani, R. J. 1990. *Generalized additive models*, volume 43. CRC press.

Hegselmann, S.; Buendia, A.; Lang, H.; Agrawal, M.; Jiang, X.; and Sontag, D. 2023. Tabllm: Few-shot classification of tabular data with large language models. In *International Conference on Artificial Intelligence and Statistics*, 5549–5581. PMLR.

Hollmann, N.; Müller, S.; and Hutter, F. 2023. GPT for Semi-Automated Data Science: Introducing CAAFE for Context-Aware Automated Feature Engineering. *arXiv preprint arXiv:2305.03403*.

Kelley Pace, R.; and Barry, R. 1997. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3): 291–297.

Lengerich, B. J.; Bordt, S.; Nori, H.; Nunnally, M. E.; Aphinyanaphongs, Y.; Kellis, M.; and Caruana, R. 2023. LLMs Understand Glass-Box Models, Discover Surprises, and Suggest Repairs. *arXiv preprint arXiv:2308.01157*.

Lengerich, B. J.; Caruana, R.; Nunnally, M. E.; and Kellis, M. 2022. Death by Round Numbers: Glass-Box Machine Learning Uncovers Biases in Medical Practice. *medRxiv*, 2022–04.

Lin, J.; Zhong, C.; Hu, D.; Rudin, C.; and Seltzer, M. 2020. Generalized and scalable optimal sparse decision trees. In *International Conference on Machine Learning*, 6150–6160. PMLR.

Liu, N. F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; and Liang, P. 2023. Lost in the Middle: How Language Models Use Long Contexts. *arXiv preprint arXiv:2307.03172*.

Lundberg, S. M.; and Lee, S.-I. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Narayan, A.; Chami, I.; Orr, L.; Arora, S.; and Ré, C. 2022. Can Foundation Models Wrangle Your Data? *arXiv preprint arXiv:2205.09911*.

Nori, H.; Jenkins, S.; Koch, P.; and Caruana, R. 2019. Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*.

Onishi, S.; Oono, K.; and Hayashi, K. 2023. TabRet: Pre-training Transformer-based Tabular Models for Unseen Columns. *arXiv preprint arXiv:2303.15747*.

OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

Rudin, C. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5): 206–215.

Slack, D.; Krishna, S.; Lakkaraju, H.; and Singh, S. 2022. Talktomodel: Understanding machine learning models with open ended dialogues. *arXiv preprint arXiv:2207.04154*.

Smith, J. W.; Everhart, J. E.; Dickson, W.; Knowler, W. C.; and Johannes, R. S. 1988. Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the annual symposium on computer application in medical care*, 261. American Medical Informatics Association.

Vanschoren, J.; Van Rijn, J. N.; Bischl, B.; and Torgo, L. 2014. OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2): 49–60.

Vos, D.; Döhmen, T.; and Schelter, S. 2022. Towards Parameter-Efficient Automation of Data Wrangling Tasks with Prefix-Tuning. In *NeurIPS 2022 First Table Representation Workshop*.

Wang, Z.; Gao, C.; Xiao, C.; and Sun, J. 2023. AnyPredict: Foundation Model for Tabular Prediction. *arXiv preprint arXiv:2305.12081*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837.

Appendix

System: You are an expert statistician and data scientist.

You interpret global explanations produced by a generalized additive model (GAM). GAMs produce explanations in the form of graphs that contain the effect of a specific input feature.

The user will first provide a general description of the dataset. Then you will be given graphs from the model, and the user will ask you questions about the graphs.

Answer all questions to the best of your ability, combining both the data contained in the graph, the data set description you were given, and your knowledge about the real world.

Graphs will be presented as a JSON object with keys representing the x-axis and values representing the y-axis. For continuous features, the keys are intervals that represent ranges where the function predicts the same value. For categorical features, each key represents a possible value that the feature can take. The y-axis depicts the contribution of the graph to the logprobs to the probability that the passenger survived.

The user will provide graphs in the following format:
- The name of the feature depicted in the graph
- The type of the feature (continuous, categorical, or boolean)
- Mean values
- Lower bounds of confidence interval
- Upper bounds of confidence interval

User: This is the titanic dataset from kaggle. The sinking of the Titanic is one of the most infamous shipwrecks in history.

On April 15, 1912, during her maiden voyage, the widely considered "unsinkable" RMS Titanic sank after colliding with an iceberg. Unfortunately, there weren't enough lifeboats for everyone onboard, resulting in the death of 1502 out of 2224 passengers and crew.

While there was some element of luck involved in surviving, it seems some groups of people were more likely to survive than others.

This dataset is used to answers the question: "what sorts of people were more likely to survive?" using passenger data (ie name, age, gender, socio-economic class, etc).

Assistant: Thanks for this general description of the data set. Please continue and provide more information, for example about the graphs from the model.

User: Consider the following graph from the model. This graph represents a continuous-valued feature. The keys are intervals that represent ranges where the function predicts the same value.

Feature Name: Age

Feature Type: continuous

Means: "(2.0, 2.5)": -0.308, "(2.5, 3.5)": 0.839, "(3.5, 12.5)": 0.91, "(12.5, 17.5)": 0.904, "(17.5, 20.0)": 0.035, "(20.0, 21.5)": 0.144, "(21.5, 25.5)": 0.304, "(25.5, 28.5)": 0.375, "(28.5, 31.5)": 0.254, "(31.5, 33.5)": 0.349, "(33.5, 36.25)": 0.399, "(36.25, 36.75)": 0.047, "(36.75, 37.5)": 0.038, "(37.5, 38.5)": 0.246, "(38.5, 39.5)": 0.2, "(39.5, 41.0)": 0.103, "(41.0, 43.0)": 0.086, "(43.0, 44.5)": -0.93, "(44.5, 46.5)": -1.153, "(46.5, 47.5)": -1.132, "(47.5, 50.5)": -0.301, "(50.5, 51.5)": 0.104, "(51.5, 52.5)": 0.121, "(52.5, 53.5)": 0.065, "(53.5, 55.0)": -0.637, "(55.0, 56.5)": -0.627, "(56.5, 57.5)": -0.648, "(57.5, 60.5)": -0.628, "(60.5, 62.5)": -0.977, "(62.5, 67.5)": -0.945, "(67.5, 80.0)": -0.887

Figure 3: (continues on the next page)

Lower Bounds (95%-Confidence Interval): ”(2.0, 2.5)”: -2.464, ”(2.5, 3.5)”: -0.518, ”(3.5, 12.5)”: -0.303, ”(12.5, 17.5)”: -0.314, ”(17.5, 20.0)”: -0.405, ”(20.0, 21.5)”: -0.299, ”(21.5, 25.5)”: 0.097, ”(25.5, 28.5)”: -0.12, ”(28.5, 31.5)”: 0.11, ”(31.5, 33.5)”: 0.153, ”(33.5, 36.25)”: 0.132, ”(36.25, 36.75)”: -0.329, ”(36.75, 37.5)”: -0.321, ”(37.5, 38.5)”: -0.285, ”(38.5, 39.5)”: -0.36, ”(39.5, 41.0)”: -0.514, ”(41.0, 43.0)”: -0.467, ”(43.0, 44.5)”: -2.148, ”(44.5, 46.5)”: -2.244, ”(46.5, 47.5)”: -2.242, ”(47.5, 50.5)”: -0.49, ”(50.5, 51.5)”: -0.522, ”(51.5, 52.5)”: -0.483, ”(52.5, 53.5)”: -0.59, ”(53.5, 55.0)”: -1.135, ”(55.0, 56.5)”: -1.135, ”(56.5, 57.5)”: -1.186, ”(57.5, 60.5)”: -1.165, ”(60.5, 62.5)”: -1.889, ”(62.5, 67.5)”: -1.804, ”(67.5, 80.0)”: -1.604

Upper Bounds (95%-Confidence Interval): ”(2.0, 2.5)”: 1.848, ”(2.5, 3.5)”: 2.196, ”(3.5, 12.5)”: 2.123, ”(12.5, 17.5)”: 2.122, ”(17.5, 20.0)”: 0.475, ”(20.0, 21.5)”: 0.588, ”(21.5, 25.5)”: 0.511, ”(25.5, 28.5)”: 0.869, ”(28.5, 31.5)”: 0.399, ”(31.5, 33.5)”: 0.544, ”(33.5, 36.25)”: 0.665, ”(36.25, 36.75)”: 0.423, ”(36.75, 37.5)”: 0.396, ”(37.5, 38.5)”: 0.778, ”(38.5, 39.5)”: 0.759, ”(39.5, 41.0)”: 0.72, ”(41.0, 43.0)”: 0.639, ”(43.0, 44.5)”: 0.288, ”(44.5, 46.5)”: -0.061, ”(46.5, 47.5)”: -0.023, ”(47.5, 50.5)”: -0.112, ”(50.5, 51.5)”: 0.73, ”(51.5, 52.5)”: 0.725, ”(52.5, 53.5)”: 0.719, ”(53.5, 55.0)”: -0.14, ”(55.0, 56.5)”: -0.118, ”(56.5, 57.5)”: -0.111, ”(57.5, 60.5)”: -0.09, ”(60.5, 62.5)”: -0.065, ”(62.5, 67.5)”: -0.087, ”(67.5, 80.0)”: -0.169

Please describe the general pattern of the graph.

Response: [First generated response]

User: Great, now please study the graph carefully and highlight any regions you may find surprising or counterintuitive. You may also suggest an explanation for why this behavior is surprising, and what may have caused it.

Response: [Second generated response]

User: Thanks. Now please provide a brief, at most 7 sentence summary of the influence of the feature on the outcome.

Response: [Third generated response]

Figure 4: The prompt structure to describe graphs that is the basis for the results in this paper. In the depicted example, we ask the model to describe the Age graph of a GAM trained on the Titanic dataset from Kaggle. Note that we provide the LLM with a description of the dataset and the meaning of the values on the y-axis in the graphs. The prompt structure is fairly general and can be easily amended to better fit various specific use cases.