A Experimental Details

In this appendix, we provide an overview of additional experimental details. First, we provide additional details about the grammar. Next, we describe how we generate our parsing training sets. Further, we provide more details about the explanation selection procedure. After, we provide examples of the sentences we had mturk workers revise to create our gold parsing datsets. Finally, we provide additional details about the user study.

A.1 Grammar Details

In this subsection, we provide additional details about the grammar. First, we describe the design of the grammar. After, we provide details about how we update the grammar for new datasets.

Design Recall from the main paper that the grammar serves as a logical form of user utterances, which TalkToModel can execute. Here, we provide more details about the grammar design. The grammar defines relations between the `operations` and the acceptable values `arguments` in Table 2. For example, the grammar defines different acceptable values for the `comparison` argument in the `filter` operation, such as `less than or equal to` or `greater than`. In addition, we structure the grammar to make parses appear closer to natural language text instead of a formal programming language like SQL or Python. The reason is that language models tend to perform better at translating utterances into grammars that are more similar to natural language instead of a programming language [56]. Consequently, we design the grammar, so that parses appear more like natural language text, without unnecessary parentheses and commas, and omitting unnecessary arguments where possible. In general, we found that simplifying the grammar and making it more like natural language as much as possible considerably improved performance. For example, the question "What are the three most important features for people older than thirty-five?" would simply translate to `filter age greater than 35 and topk 3`. Note, here, because TalkToModel is applied to only one dataset at a time, the `dataset` argument can be omitted for simplicity. Practically, we implement the grammar in Lark [57] because the implementation supports interactive parsing, simplifying the process of implementing the guided decoding strategy.

Updating the Grammar For Datasets and User Utterances In the main paper, we discussed how we update the grammar based on the dataset. Here, we provide more description about how we update the grammar. We update the grammar based on the feature names and categorical feature values in the dataset. In particular, the acceptable values for the `feature` argument (Table 2) in the grammar becomes all the feature values in the dataset. Further, the `value` argument for a categorical `feature` becomes the set of categorical feature values that appear in the data for that feature. Because there are many possible values of numeric features, we instead extract potential numeric values from user utterances as they are provided to the system. Specifically, we set the `value` argument in the grammar for numeric features to contain the set of numeric values that appear in the user utterance. We additionally support string based numeric values (e.g., "fifty-five" or "twelve") to make the system handle a wider variety of cases.

A.2 Training Dataset Generation

In this subsection, we provide details about the generation of the (utterance, parse) pair training dataset. To ensure that we generate a diverse and comprehensive set of (utterance, parse) pairs for training, we compose a total of 687 templates that use 6 different wildcard types. The templates

consist of a diverse set of utterances that encompass the different operations permitted in the system. The wildcards include categorical feature names, numeric feature names, class names, numeric feature values, categorical feature values, explanation types, and common filtering expressions (e.g., "{NUMERIC_FEATURE} above {NUMERIC_VALUE}"). Because templates can have potentially many wildcards, we recursively enumerate all the wildcard values for each parse. Further, we also limit the number of values for certain wildcards to ensure the number of training pairs generated does not become extremely large. In particular, we limit the number of numeric values to 2 values per feature. In addition, to prevent templates with many wildcards from dominating the training dataset, we also downsample the number of values per wildcard to 2 values after the initial recursion. In this way, we the training dataset does not get dominated by a few templates that have many wildcards, which we found improves performance.

As an example, an utterance template is "Explain the predictions on data with {NUMERIC_FEATURE} greater than {NUMERIC_VALUE}" and the corresponding parse template is `filter {NUMERIC_FEATURE} greater than {NUMERIC_VALUE} and explain feature importance`. From there, we enumerate the numeric features in the dataset and a selection of numeric feature values, substituting these into {NUMERIC_FEATURE} and {NUMERIC_VALUE} respectively to generate data.

A.3 Explanation Selection

Here, we provide additional details about the explanation selection algorithm. For our explanation selection algorithm, we set $\sigma = 0.05$, K to floor($\frac{d}{2}$), and $N = 10,000$. Further, because it would not make sense to perturb categorical features using Gaussian noise, we permute these features to another value occurring in the feature 30% of the time. We additionally use both LIME [50] and KernelSHAP [37] with their default settings while computing explanations. In addition, we include LIME with the following kernel widths: $[0.25, 0.50, 0.75, 1.0]$, to generate explanations with a variety of different localities.

In addition, if the different in fidelity between the top two explanation methods is small, so it is hard to determine which explanation is most accurate, we use the explanation stability metric proposed by Alvarez-Melis and Jaakkola [6] to break ties, because it is more desirable for the explanation to robust to perturbations [59, 3]. In order to use the *stability* metric proposed by Alvarez-Melis and Jaakkola [6] to break ties if the explanations fidelities are quite close (less than $\delta = 0.01$), we compute the jaccard similarity between feature rankings instead of the $l2$ norm as is used in their work. The reason is that the norm might not be comparable between explanation types, because they have different ranges, while the jaccard similarity should not be affected. Further, we compute the area under the top k curve using the jaccard similarity stability metric, as in Equation 3, to make this measure more robust.

A.4 Gold Parsing Dataset

In this subsection, we provide examples of 10 of the sentences provided to mturk workers to revise during our data generation process for each dataset. The examples are provided in Table 6 and illustrate the different types of utterances revised by mturkers to create a comprehensive testing set. Also, we provide the number of (utterance, parse) pairs in the IID and compositional splits in Table 7.

A.5 User Study Questions

In this subsection, we provide the questions participants answered in the user study. The questions are provided in Table 8. One of the two question blocks, Block 1 or Block 2, is shown for TalkToModel

Table 6: Examples of 10 sentences out of 50 from each dataset provided to mturk workers to revise.

COMPAS	What is your reasoning for determining if people older than 20 are likely to commit crimes?
	How likely are people that are younger than 25 or have committed at least 1 crime in the past to commit a crime in future?
	what are top 3 most important features you use for prediction for people if they were to decrease their prison terms by 10 months?
	For people that are 18 years old and black, how often are you correct in predicting whether they will commit crimes in the future?
	let's look at those in the data with 3 or more prior crimes on record. what are some common mistakes the model makes on these people?
	For this subset in the data, how accurate is the model?
	For people that are 18 years old and black, how often are you correct in predicting whether they will commit crimes in the future?
	how likely would the person with the id number of 33 in the data be to a commit a crime if they were 5 years younger?
	But what if they were twenty years older?
	Could you show me some data for people who are black?
Diabetes	How likely are people that either (1) have had two pregnancies or (2) are older than 20 and younger than 30 to have diabetes?
	What are the top five most important features for the model's predictions on people with a bmi over 40?
	Show the data for people older than 20. Then, could you show me the predictions on this data?
	what would happen to the likelihood of having diabetes if we were to increase glucose by 100 for the data point with id 33
	What's the average age in the data?
	For this subset in the data, how accurate is the model?
	What does patient number 34 need to do in order to be diagnosed as unlikely to have diabetes?
	What are the reasons why the model predicted data point number 100 and what could you do to change it?
	How would the predictions change if age were decreased by 5 years for people with a bmi of 30?
	How do the features of the data interact in the model's predictions on this particular data?
German	If people in the data were unemployed, how important would the age and loan amount features be for predicting credit risk?
	what would happen to the likelihood of being bad credit risk if we were to increase the loan amount by 250 for the data point with id 89
	what are top three most important features for determining whether those who are applying for furniture loans are good credit risk?
	What is the average loan amount for people with no current loans and that do not own a house?
	How accurate are you at predicting whether people who are asking for loans for home appliances are good credit risk?
	In the dataset, if the loan duration were to be increased by 2 years, what would the predictions for the data be?
	Could you show me some examples the model predicts incorrectly and how accurate the model is on the data?
	What do you predict on the instances in the data? Also, could you show me an example of a few mistakes you make in these predictions?
	But why did you think these people are bad credit risk?
	If these people were not unemployed, what's the likelihood they are good credit risk? Why?

Table 7: The number of gold (utterance, parse) pairs in the IID and Compositional splits for the datasets. There are relatively more IID questions in each dataset.

	COMPAS	Diabetes	German
IID	117	147	127
Compositional	29	43	74
Overall	146	190	201

and the other is shown for the dashboard (the ordering of TalkToModel and the dashboard is also randomized). The two question blocks include similar concepts to ensure a similar level of difficulty but include different numbers to discourage memorization between the question blocks.

Table 8: The question's participants answered during the user study. Participants are shown either the questions in Block 1 or Block 2 for TalkToModel and the other set of questions for dashboard.

Block 1	What are the three most important features for the model's predictions on people older than 30 with bmi's above 35?
	What is the feature importance ranking of the age feature for data point id 188?
	How many individuals in the dataset are predicted to be likely to have diabetes but are not actually likely to have it?
	If patient id 293 were to decrease their bmi by five, what's the prediction probability of the "likely to have diabetes" class?
	Is the "glucose" feature more important than the "age" feature for the model's prediction on data point 49?
Block 2	What are the three most important features for the model's predictions on people younger than 23 with glucose levels below seventy-five?
	What is the feature importance ranking of the insulin feature for data point id 57?
	How many individuals in the dataset are predicted not likely to have diabetes but actually are likely to have it?
	What's the likelihood of patient 57 having diabetes if they increased their glucose levels by 100 and bmi by 3?
	How important is the "diabetes pedigree function" feature compared to the "glucose" feature for the model's prediction on data point 55?

A.6 User Study Length & Compensation

The survey took around 30 minutes for participants to complete. We compensated the ML professionals by providing them with a $20 gift card. We paid the prolific workers $14.74/hour on average for completing the survey, considering their individual completion times.

Table 9: The improvement in fidelity using our explanation selection procedure described in Equation 3 in the main paper. We present the average value and (1 STD) for each metric across the 3 datasets. There are significant differences in the explanation fidelities, and the worst case explanation fidelity can be much worse than the one selected. These results demonstrate the importance of selecting explanations for end users, who may otherwise be left with comparatively lower fidelity explanations without this procedure.

	COMPAS	Diabetes	German
% Fidelity Decrease From Best Using Median Fidelity Explanation	54.9% (41.0)	23.0% (20.7)	5.0% (4.5)
% Fidelity Decrease From Best Using Worst Fidelity Explanation	68.5%(42.7)	55.2% (31.8)	57.1% (16.1)
Absolute Improvement Using Best Over Median Fidelity	0.001 (0.002)	0.035 (0.060)	0.015 (0.012)
Absolute Improvement Using Best Over Worst Fidelity	0.042 (0.071)	0.002 (0.002)	0.186 (0.08)

B Additional Experiments

In this appendix, we provide additional experimental results. First, we provide experimental results where we show the advantages of explanation selection. Second, we give results on the affects of the number of training templates. After, we provide error analysis for our parsing models. Last, we give additional user study results.

B.1 Advantages of Explanation Selection

In the main paper, we discussed how we select explanations based on those that have the highest fidelity in order to ensure the user receives the most accurate explanations possible (Equation 3). In this subsection, we provide more details about the advantages of explanation selection and show that without the procedure, users may receive comparatively lower fidelity explanations, demonstrating the importance of the technique.

Our evaluation procedure is as follows. For a data point, we compute the topk fidelity AUCs from all the explanations we consider (details about the explanation pool in Appendix A.3). Next, we compute several metrics that capture the improvement in fidelity using our selection technique. We compute the absolute improvement over the worst fidelity explanation, absolute improvement over the median fidelity explanation, percent decrease in fidelity using worst fidelity explanation, and percent decrease in fidelity using the median fidelity explanation. We perform this procedure for all the data points in a dataset.

We repeat this evaluation procedure for the three datasets we consider in the main paper: Compas, Diabetes, and German Credit. The results provided in Table 9 demonstrate that explanation fidelity can be much worse without selecting for explanations. For instance, if experts without sufficient data science experience were to run an explanation type arbitrarily, these results demonstrate they could have relatively much less faithful explanations.

B.2 Effects of the number of training templates

In this subsection, we provide experimental details about the effects of the number of training templates on the parsing accuracy of the system. Because we use a template strategy to generate training data (Section 4), we must decide on how many prompts to write and include in the training scheme. This raises the question of how the number of templates affects model performance. To understand this behavior, we retrain the T5-Base model, randomly sampling the number of training templates over different percentages of the original templates set. In particular, we sweep over the following

	5-Shot		10-Shot	
	IID	Compositional	IID	Compositional
Compas	29.8	77.8	15.3	53.6
Diabetes	40.0	40.0	19.1	28.2
German	44.2	50.8	28.3	26.9

Table 10: Percentage of mistakes for few-shot GPT-J 6B where selected prompts *do not* include all the operations in the parse of the user utterance. We see that most of the time the operations for the parse of the user utterance are included in the prompts for the 10-Shot models, yet these methods still perform relatively poorly compared to finetuned T5.

percentages $[20\%, 40\%, 60\%, 80\%, 100\%]$ on the diabetes dataset, downsampling and retraining 5 times per template. We give the results in Figure 2. We see that there are clear accuracy gains over using a smaller number of templates. Further, the gains for the compositional model performance seem to somewhat level off, but this is not the case for the IID split, suggesting further templates may help IID performance.

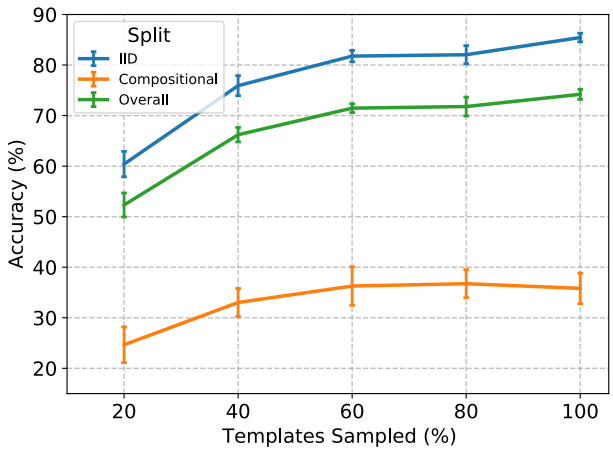

Figure 2: **Randomly sampling prompt templates and re-training T5-Base on the Diabetes dataset. For each down-sample %, the prompts are randomly down-sampled and the model is re-trained 5 times. The error bars are 1 standard deviation.**

B.3 Parsing Error Analysis

In the main text, we demonstrated that the fine-tuned T5 models performed considerably better than few-shot GPT-J (Table 3). In this subsection, we perform additional error analysis for why this occurs. This dynamic brings up the question: what is the cause of these poor few-shot results? Since the few-shot GPT-J models select the (utterance, parse) pairs from the synthetic dataset using nearest neighbors on a sentence embedding model, it could be possible these issues are due to the sentence embedding model failing to select good pairs. In particular, this nearest neighbors technique could fail to select pairs with the operations necessary to parse the user utterance, and not the model failing to learn from the examples. To evaluate whether this is the case, we compute the percent of mistakes that *do not* include the operations necessary to parse the user utterance for GPT-J. The results provided in Table 10 demonstrate that, especially for the 10-Shot case, the operations needed to parse the user

utterance *are* included in the prompts, indicating the issues are likely due to the model's capacity to learn few-shot, rather than the selection mechanism. In this work, we were limited by using up to 6-billion parameter GPT-J, but it could be possible to achieve better results with larger models, as results on emergent abilitiees suggest [70].

B.4 User Study Results: Per Question Likert Scores

In this subsection, we provide additional user study results. In addition to the questions asked at the end of the survey (Table 4 and Table 5), we also asked users to rate their experiences using both interfaces on a 1-7 Likert while they were taking the survey. In particular, we asked users how much they agreed with the following statements:

- I am confidence I completed my answer correctly.
- Completing this task took me a lot of effort.
- The interface was useful for completing the task.
- Based on my experience so far, I trust the interface to understand machine learning models.
- Based on my experience so far, I would use the interface again to understand machine learning models.

To evaluate these results, we compute the mean and standard deviation of the Likert score for the 1st through 5th question each user sees (question ordering is randomized so users see different questions first). We compute this for each statement and interface. The results for the medical workers are provided in Figure 3 and the ML professionals in Figure 4. Overall, the medical workers clearly prefer TalkToModel while answering the questions. Interestingly, they seem to gain trust in TalkToModel over time, going from "somewhat agree" to "agree" with the statement "Based on my experience so far, I trust the interface to understand ML models" by the end of the survey.

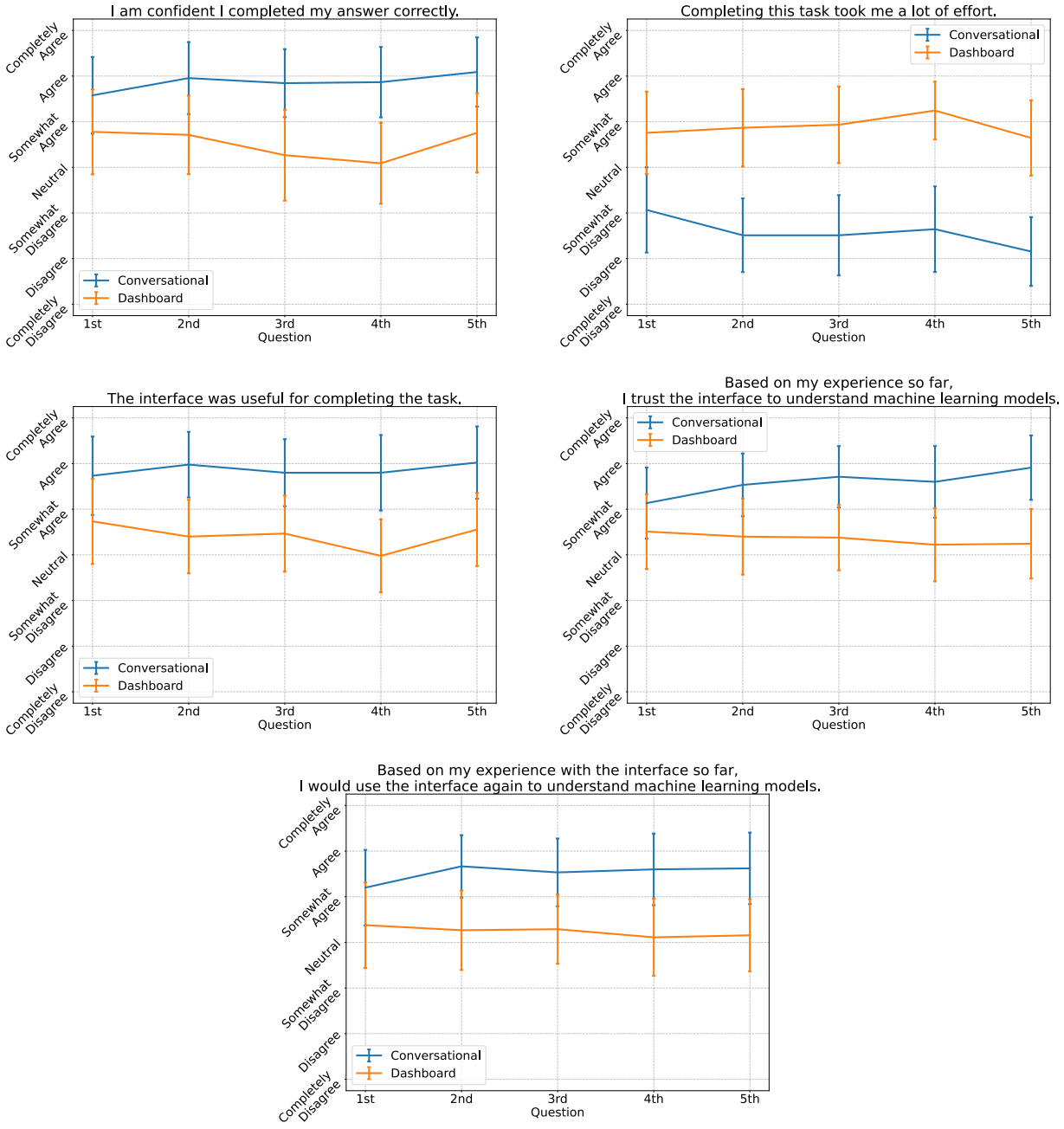

Figure 3: **Medical Worker per question likert results:** in general, these participants preferred TalkToModel over the dashboard to answer the questions.

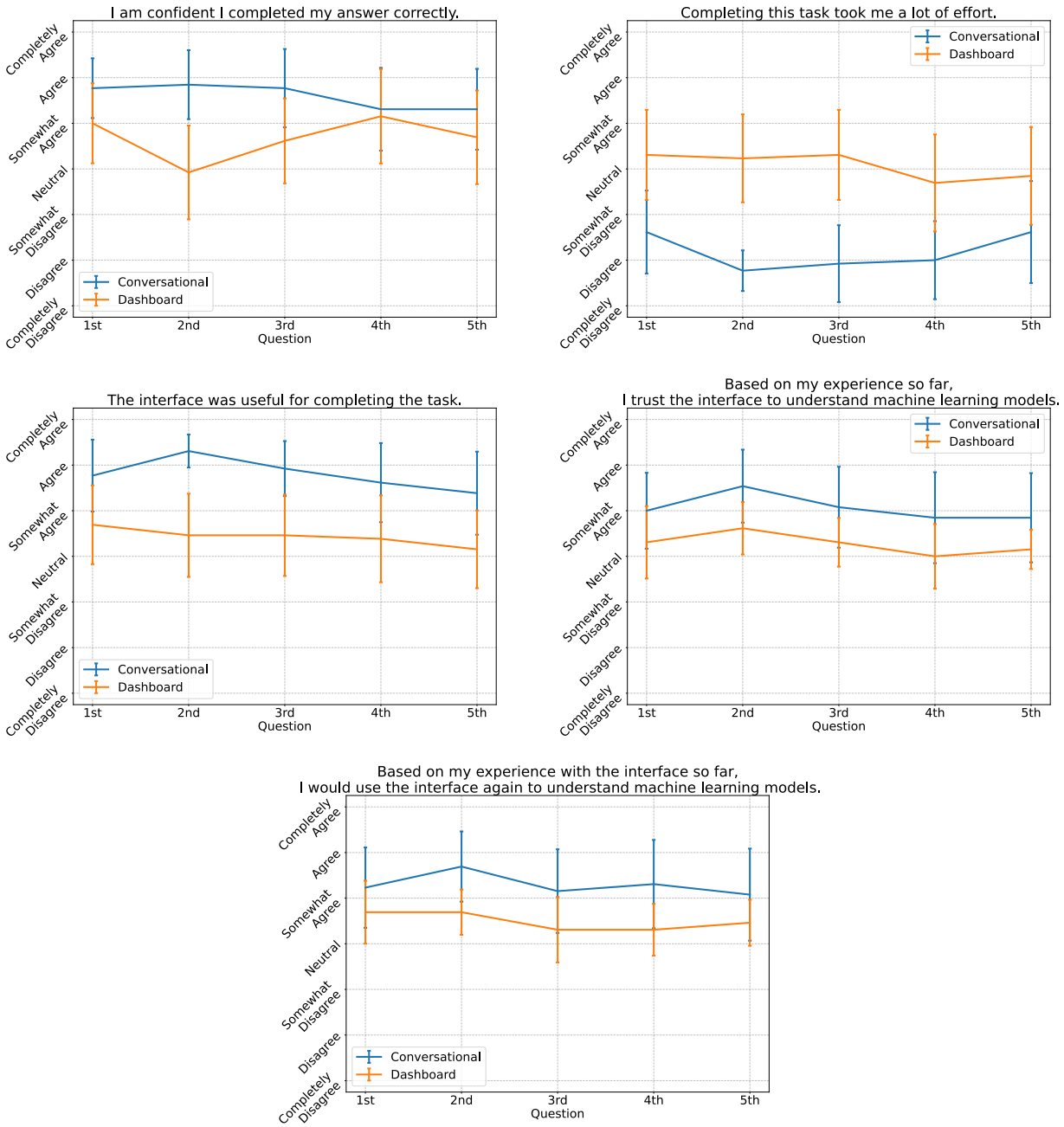

Figure 4: **ML professionals per question likert results:** these participants were more mixed about which interface they preferred while taking the survey. Interestingly, while the participants were much more accurate using TalkToModel, this population rated their answers at a similar confidence and said they trusted the interfaces similarly while taking the survey.

C Interface Screenshots

In this appendix, we provide additional interface screen shots for both TalkToModel and the dashboard baseline.

C.1 TalkToModel Interface

We provide additional screenshots of the TalkToModel GUI interface in Fig. 5 and Fig. 6. The interface is a text chat interface, where users provide input and the system response. On the right side of the interface, users can "pin" messages they find interesting. In addition, we provide a feature where users can get help to generate a question in a particular category (Fig. 7). This feature inputs a random synthetic training utterance into the command bar from the category of choice.

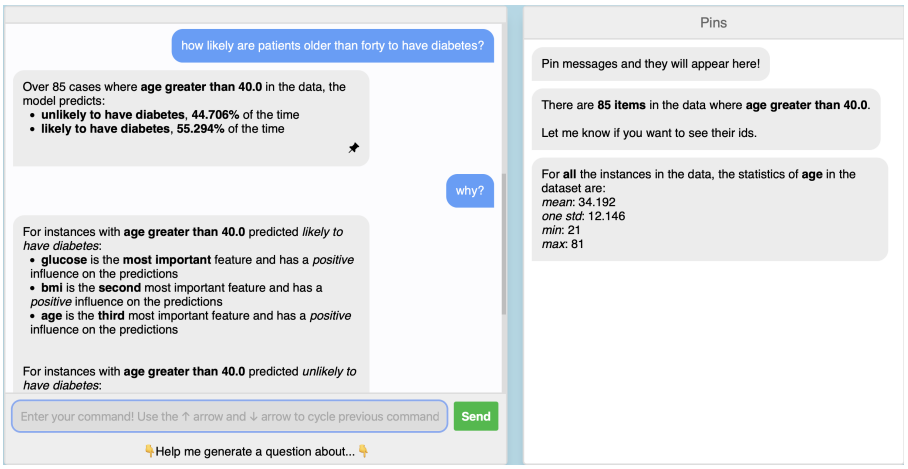

Figure 5: Discussing the model's predictions on patients older than 40. The system correctly understand's the context of the question "why?".

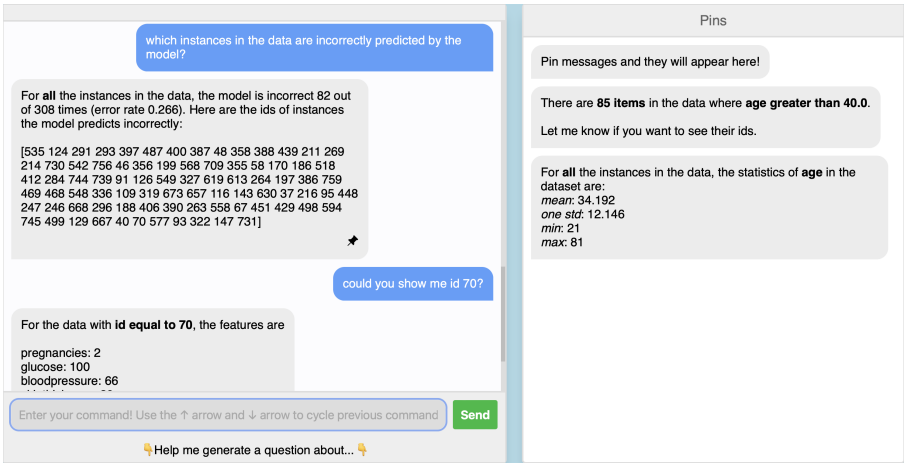

Figure 6: Beyond explanations, TalkToModel is also highly useful for classic error analysis, such as inspecting incorrectly predicted instances, as is shown here.

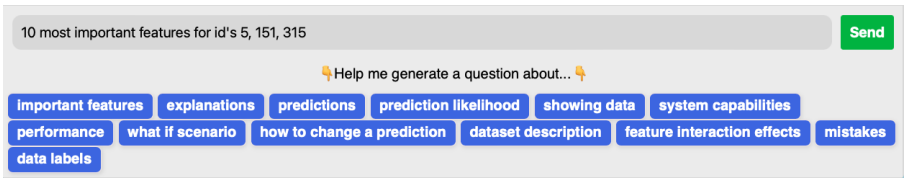

Figure 7: Screen shot of the help generate a question button, which will input a random question from the training data in the category when clicked..

C.2 Dashboard Interface

We provide example screenshots for the baseline dashboard interface in Fig. 8 and Fig. 9 [17]. In depth description can be found in the project's documentation https://explainerdashboard.readthedocs.io/en/latest/. The dashboard provides different ways to understand ML models using point and click interactions.

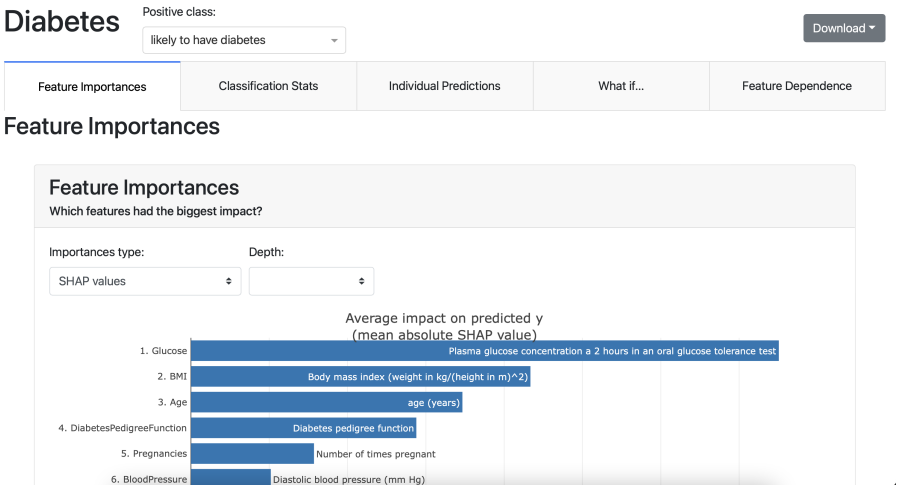

Figure 8: The dashboard interface lands on the global feature importances. Users can navigate to other tabs in the interface to see model predictions, metrics, feature importances, and compute what-if scenarios.

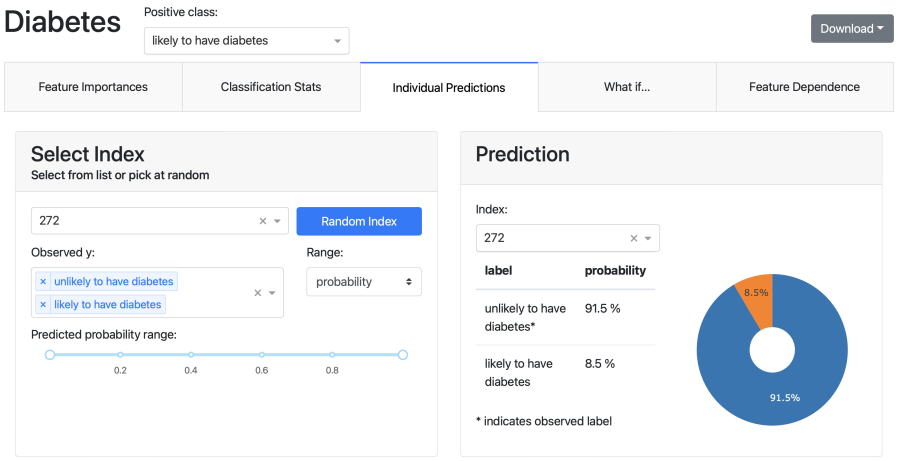

Figure 9: An example of the "individual predictions" page in the dashboard interface. Users need to navigate to this page and type in the data point index they want in order to inspect predictions.