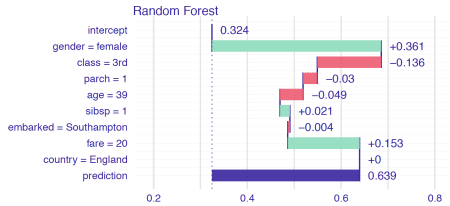
**TL;DR:** If you are going to explain predictions for a black box model you should combine statistical charts with natural language descriptions. This combination is more powerful than SHAP/LIME/PDP/Break Down charts alone. During this summer Adam Izdebski implemented this feature for explanations generated in R with DALEX library. How he did it? Find out here:

**Is text better than charts for explanations?**  
Packages from DrWhy.AI toolbox generate lots of graphical explanations for predictive models. Available statistical charts allow to better understand how a model is working in general (global perspective) or for a specific prediction (local perspective).  
Yet for domain experts without training in mathematics or computer science, graphical explanations may be insufficient. Charts are great for exploration and discovery, but for explanations they introduce some ambiguity. Have I read everything? Maybe I missed something?  
To address this problem we introduced the describe() function, which  
automatically generates textual explanations for predictive models.

**Insufficient interpretability**  
Domain experts without formal training in mathematics or computer science can often find statistical explanations as hard to interpret. There are various reasons for this. First of all, explanations are often displayed as complex plots without instructions. Often there is no clear narration or interpretation visible. Plots are using different scales mixing probabilities with relative changes in model’s prediction. The order of variables may be also misleading. See for example a Break-Down plot.



The figure displays the prediction generated with Random Forest that a selected passenger survived the Titanic sinking. The model’s average response on titanic data set (intercept) is equal to 0.324. The model predicts that the selected passenger survived with probability 0.639. It also displays all the variables that have contributed to that prediction. Once the plot is described it is easy to interpret as it posses a very clear graphical layout.  
However, interpreting it for the first time may be tricky.

**Properties of a good description**  
Effective communication and argumentation is a difficult craft. For that reason, we refer to [winning debates strategies](https://debate.uvm.edu/dcpdf/Steven_Johnson_Winning_Debates_2009.pdf) as for guiding in generating persuasive textual explanations. First of all, any description should be  
intelligible and persuasive. We achieve this by using:

* **Fixed structure:** Effective communication requires a rigid structure. Thus we generate descriptions from a fixed template, that always includes a proper introduction, argumentation and conclusion part. This makes the description more predictable, hence intelligible.
* **Situation recognition:** In order to make a description more trustworthy, we begin generating the text by identifying one of the scenarios, that we are dealing with. Currently, the following scenarios are available:
  + The model prediction is significantly higher than the average model prediction. In this case, the description should convince the reader why the prediction is higher than the average.
  + The model prediction is significantly lower than the average model prediction. In this case, the description should convince the reader why the prediction is lower than the average.
  + The model prediction is close to the average. In this case the description should convince the reader that either: variables are contradicting each other or variables are insignificant.

Identifying what should be justified, is a crucial step for generating persuasive descriptions.

**Description’s template for persuasive argumentation**  
As noted before, to achieve clarity we generate descriptions with three separate components: an introduction, an argumentation part, and a summary.

**An introduction** should provide a claim. It is a basic point that an arguer wishes to make. In our case, it is the model’s prediction. Displaying the additional information about the predictions’ distribution helps to place it in a context — is it low, high or close to the average.

**An argumentation part** should provide evidence and reason, which connects the evidence to the claim. In normal settings this will work like that: This particular passenger survived the catastrophe (claim) because it was a child (evidence no. 1) and children were evacuated from the ship in the first order as in the phrase women and children first. (reason no. 1) What is more, the children were traveling in the 1-st class (evidence no. 2) and first-class passengers had the best cabins, which were close to the rescue boats. (reason no. 2).

The tricky part is that we are not able to make up a reason automatically, as it is a matter of context and interpretation. However what we can do is highlight the main evidence, that made the model produce the claim. If a model is making its’ predictions for the right reason, evidences should make much sense and it should be easy for the reader to make a story and connect the evidence to the claim. If the model is displaying evidence, that makes not much sense, it also should be a clear signal, that the model may not be trustworthy.

**A summary** is just the rest of the justification. It states that other pieces of evidence are with less importance, thus they may be omitted. A good rule of thumb is displaying three most important evidence, not to make the picture too complex. We can refer to the above scheme as to creating relational arguments as [in winning debates guides](https://debate.uvm.edu/dcpdf/Steven_Johnson_Winning_Debates_2009.pdf).

**Implementation**  
The logic described above is implemented in ingredients and iBreakDown packages.

For generating a description we should pass the explanation generated by ceteris\_paribus() or break\_down() or shap() to the describe() function.

describe(break\_down\_explanation)

# Random Forest predicts, that the prediction for the selected instance is 0.639 which is higher than the average.

# The most important variables that increase the prediction are gender, fare.

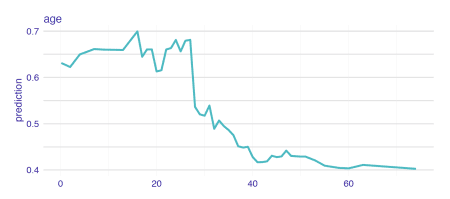
# The most important variable that decrease the prediction is class.

# Other variables are with less importance. The contribution of all other variables is -0.063 .

There are various parameters that control the display of the description making it more flexible, thus suited for more applications. They include:

* generating a short version of descriptions,
* displaying predictions’ distribution details,
* generating more detailed argumentation.

While explanations generated by iBreakDown are feature attribution explanations that aim at providing interpretable reasons for the model’s prediction, explanations generated by ingredients are rather speculative. In fine, they explain how the model’s prediction would change if we perturb the instance being explained. For example, ceteris\_paribus() explanation explores how would the prediction change if we change the values of a single feature while keeping the other features unchanged.



describe(ceteris\_paribus\_explanation, variables = “age”)

# For the selected instance Random Forest predicts that , prediction is equal to 0.699.

# The highest prediction occurs for (age = 16), while the lowest for (age = 74).

# Breakpoint is identified at (age = 40).

# Average model responses are \*lower\* for variable values \*higher\* than breakpoint.

**Applications and future work**

Generating natural language explanations is a sensitive task, as the interpretability always depends on the end user’s cognition. For this reason, experiments should be designed to assess the usefulness of the descriptions being generated. Furthermore, more vocabulary flexibility could be added, to make the descriptions more human alike. Lastly, descriptions could be integrated with a chatbot that would explain predictions interactively, using the framework described here. Also, better discretization techniques can be used for generating better continuous ceteris paribus and aggregated profiles textual explanations.