# Feature Relevance

In the context of feature attribution, we address the following question:

> **How relevant is a feature for my target?**

There are multiple approaches to tackle this question. In this context, we utilize the Shapley values concept with specifically tailored set functions, which can be changed depending on the problem at hand. In the most common setting, this is similar to the SHAP package. However, we provide a simple and flexible method to modify the set function, thus customizing the explanations.

We approach this with a causal framework where the noise (i.e., unexplained) is also included as one of the input features of a model. In other words, we can examine the causal mechanism, which includes the parents of a node and the noise as input.

Refer to the following paper for further insights into how Shapley values can be interpreted as causal explanations of a given model:

> Dominik Janzing, Lenon Minorics, Patrick Blöbaum. Feature relevance quantification in explainable AI: A causal problem International Conference on Artificial Intelligence and Statistics, 2907-2916, 2021.

## How to use it

Although there are non-causal use cases as well, and our tools can be utilized for them too, we will first look at an example where we use the causal mechanism of a node. As usual, we first generate some example data:

Skip to main content

```
>>> import numpy as np, pandas as pd, networkx as nx
>>> from dowhy import gcm
```

```
>>> X = np.random.normal(loc=0, scale=1, size=1000)
>>> Z = np.random.normal(loc=0, scale=1, size=1000)
>>> Y = X + 3 * Z + np.random.normal(loc=0, scale=1, size=1000)
>>> data = pd.DataFrame(data=dict(X=X, Y=Y, Z=Z))
```

Now, there are different options to analyze this. Let's initially model this as a causal model:

```
>>> causal_model = gcm.InvertibleStructuralCausalModel(nx.DiGraph([('X', 'Y'), ('Z
>>> gcm.auto.assign_causal_mechanisms(causal_model, data)
>>> gcm.fit(causal_model, data)
```

We can now evaluate the global relevance (population/distribution level) of the parents:

```
>>> parent_relevance, noise_relevance = gcm.parent_relevance(causal_model, target_
>>> parent_relevance, noise_relevance
    {('X', 'Y'): 1.1211907746332785, ('Z', 'Y'): 8.92516062224172}, [1.0313637]
```

in the context of global relevance, the inputs' relevance is evaluated with respect to their contribution to variance by default. Here, we can see that the parent $Z$ is clearly the most relevant feature. We can also see the noise's relevance, which, in this case, is equivalent to the relevance of $X$.

In the above example, we leveraged the causal structure directly and estimated global relevance. However, we can also use this method with a black box predictor. For this, let's use a linear model as an example:

```
>>> from sklearn.linear_model import LinearRegression
>>> from dowhy.gcm.util.general import variance_of_deviations
```

```
>>> mdl = LinearRegression()
>>> mdl.fit(data[['X', 'Z']].to_numpy(), Y)
>>> relevance = gcm.feature_relevance_distribution(mdl.predict, data[['X', 'Z']].t
>>> relevance
    [0.98705591 8.95981759]
```

Skip to main content

In this case, we directly defined our subset scoring function as the variance of deviations and obtained similar results.

Finally, let's look at the relevance for an individual observation, rather than for the entire population. For this, suppose we want to explain the relevance for the following observation:

```
>>> single_observation = np.array([[2, 1]])
```

Here, $X = 2$ and $Z = 1$, resulting in $Y = 5$. Let's evaluate the relevance for this single observation:

```
>>> from dowhy.gcm.util.general import means_difference
```

```
>>> relevance = gcm.feature_relevance_sample(mdl.predict, data[['X', 'Z']].to_nump
>>> relevance
    [[2.01652995 3.04522823]]
```

As expected, even though $X$ has a larger value than $Z$, $Z$ is more relevant for the model's outcome due to its coefficient.

Note: Similar to the other methods, while we demonstrated and validated the method using a linear relationship here, it can also handle arbitrary non-linear relationships.

Previous Next