|  |  |
| --- | --- |
| Desiderata | How to test |
| Model-agnostic | - (by design) |
| Valid | Check that new prediction is of the target class |
| Robust | - (by design: deterministic) |
| Sparse | Count how many features are shown |
| Similar | Calculate the distance to the data point |
| Realistic | Calculate the distance to training data point |
| Informative | Show that the proposed feature changes are in line with the SHAP values |

Smart way to generate sparser explanations;

* Only show feature changes that are in line with the SHAP values.

## Distance kernel idea

How? Add the feature values of each training point together, weighted by the training point’s distance to the profile we are trying to explain. This creates a new data point which is not guaranteed to be realistic. If we then search for the most similar training data point to this average, we require a second run through all the training data points which is slow.

This was brought up because the initial implementation often resulted in counter-intuitive explanation; e.g. adjusting a feature value in the opposite direction than expected. For instance, decreasing the TAC value when increasing the NOC. We tried to come up with ideas to counteract this problem. E.g. by creating clusters and presenting the median as a datapoint. However, clustering comes with extra parameters that are difficult to accurately optimize. A simpler approach would be to not change features that are against the median of the closest *n* features, which would again require the optimization of *n*, and no guarantees that the prediction is still valid.

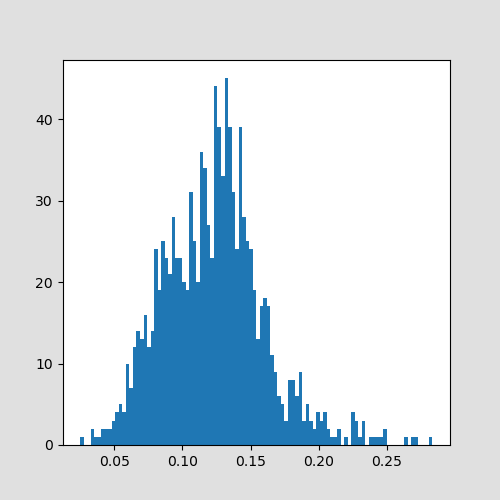
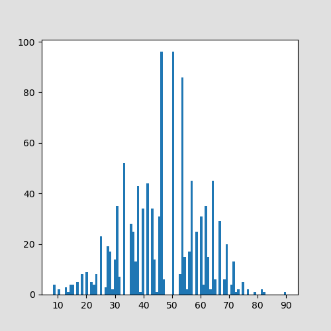
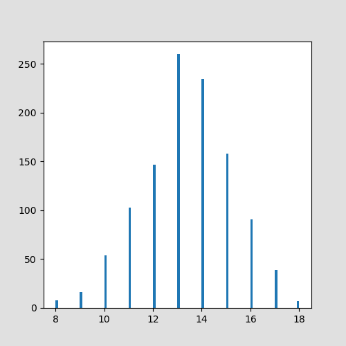
We noticed this issue mainly before we sampled the extra 5000 data points, showing how it could have mainly been caused by the sparsity of the data. It has improved since sampling.

These also stray from the concept of a counterfactual explanation i.e. finding *the most similar world* to the original profile, yet with the target prediction. By simply presenting an average data point, this loses all the quirks of what makes a profile unique. Presenting a profile that has similar quirks as the profile to be explained, even though those do not fit the average, is preferred.

For the following two approaches, we use a heuristically chosen value for the weight of the two scores of distance and number of features changed. (10 \* distance + 1 \* num\_features\_changed).

Using the weighted kernel approach, (1 – distance scores) we obtain the following measures on 1118 test samples.

|  |  |
| --- | --- |
| Metric | Score |
| Average number of features changed | 13 (/19) |
| Average number of features changed with counterintuitive SHAP changes | 9 (/19) |
| Average distance to the profile | 0.123 (/1) |
| Average distance to the closest training point | 0 |



All in all, the CF profiles picked here have a high number of changed features, a high distance to the profile we want to explain, and a large number of feature changes that do not move the prediction in the right direction.

For each proposed change:

Check if that new scaled value exists in training data in combination with a marge of 5% of the other, unchanged scaled features

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | DiCE random | DiCE genetic | WhatIf | Pareto min | Pareto min+ filter | Pareto med | Pareto med + filter | Weighted | Weighted + filter |
| Avg. features changed | 1.73 | 11.5 | 8.41 | 7.69 | 3.95 | 7.81 | 4.02 | 7.88 | 4.02 |
| Avg. distance to profile | 0.0308 | 0.0840 | 0.0394 | 0.0453 | 0.02246 | 0.0442 | 0.0219 | 0.0415 | 0.0204 |
| Avg. distance to training point | 0.0574 | 0.0194 | 0.00 | 0.00 | 0.02188 | 0.00 | 0.02149 | 0.00 | 0.02079 |
| Avg. realism score | 0.26 | 0.55 | 1.00 | 1.00 | 0.90 | 1.00 | 0.90 | 1.00 | 0.90 |
| Targets missed | 18 | 21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Hours | 0.3 | 0.8 | 3.0 | 3.0 | 7.1 | 3.0 | 7.4 | 3.0 | 7.2 |

Good to find out if users:

* Understand the model better
* Will change their choice or not
* Can identify when the model might be wrong

Write down the overall user experience (soft evaluation is important)

When looking at the training data, to find the closest CF, we need to optimize two scores

Distance and number of features changed

They both lie between 0 and 1, but they are differently distributed.

To make sure both scores contribute somewhat equally, we assign a weight according to the means of the scores on the training data

Distance mean: 0.12644057709891549

Distance median: 0.12299491805042531

Distance minimum: 0.003678513596551171

Distance maximum: 0.4208296031944699

Features changed mean: 0.7044902436521877

Features changed median: 0.6842105263157895

Features changed minimum: 0.15789473684210525

Features changed maximum: 1.0

To obtain the same mean, we multiply the distance score by a weight of 5. Transforming the means to be equal would require a weight of 5.8, but we want to penalize the number of features change a bit more heavily, we round the distance weight down to 5.

Fixing the distance measure so that we take the range of the feature and not the max.

Distance mean: 0.12836387164377064

Distance median: 0.12488870016401843

Distance minimum: 0.003700578219583392

Distance maximum: 0.4264238466522581

Features changed mean: 0.7044902436521877

Features changed median: 0.6842105263157895

Features changed minimum: 0.15789473684210525

Features changed maximum: 1.0

MAD (half the training data)

Distance mean: 3593649679.6645336

Distance median: 2369742.224960396

Distance minimum: 0.08198083035227413

Distance maximum: 414599452940.98975

Features changed mean: 0.7044250880823222

Features changed median: 0.6842105263157895

Features changed minimum: 0.15789473684210525

Features changed maximum: 1.0

Completely dominated by one feature that has large variance.