|  |  |
| --- | --- |
| 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.

Qualitative evaluation

## 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.

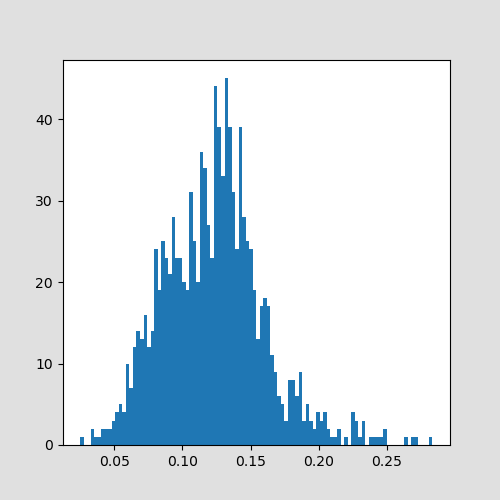
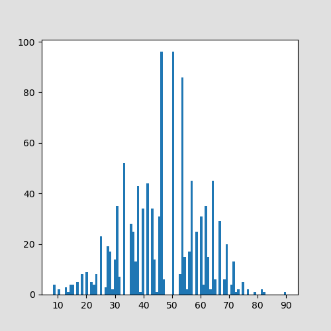
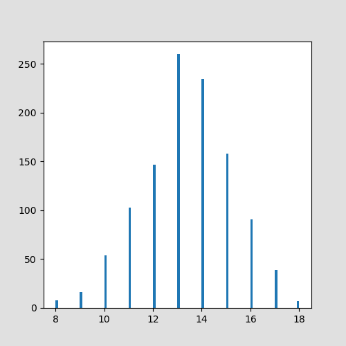
We noticed this issue mainly before we sampled the extra 5000 data points, showing how it could have mainly been caused by the sparsity of the data. It has improved since sampling.

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.

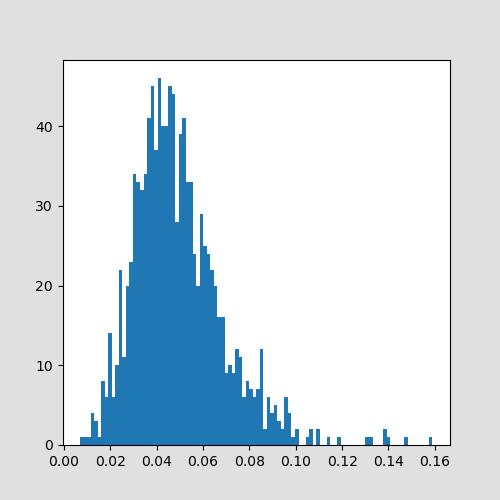
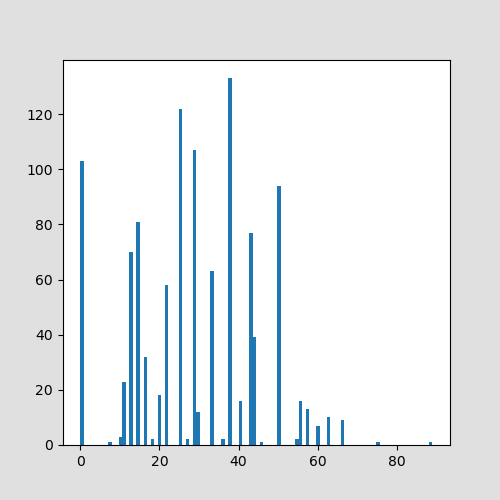
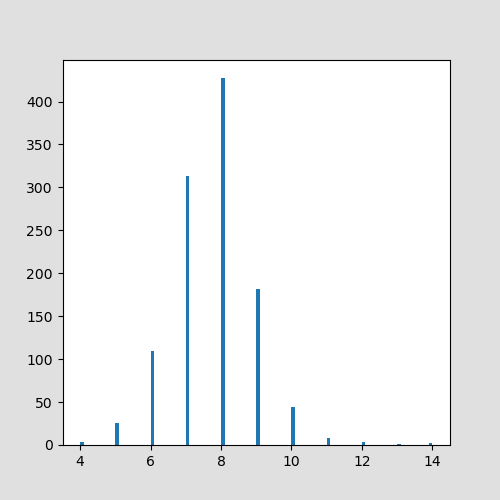
|  |  |
| --- | --- |
| Metric | Score |
| Average number of features changed | 13 (/19) |
| Average number of features changed with counterintuitive SHAP changes | 9 (/19) |
| Average distance to the profile | 0.123 (/1) |
| Average distance to the closest training point | 0 |



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.

In contrast, when we simply pick the closest profile according to this weighted sum, we obtain the following scores:

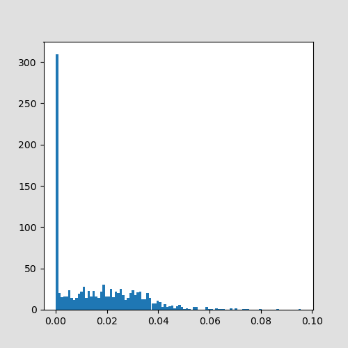
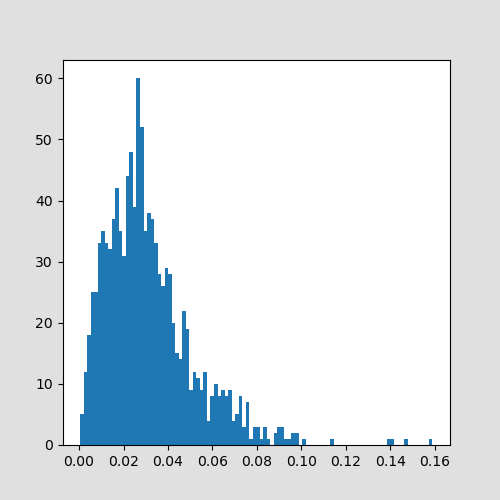
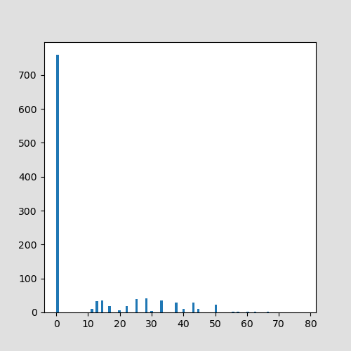
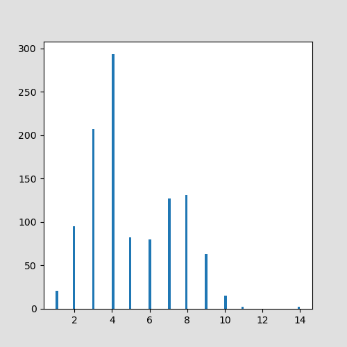
|  |  |
| --- | --- |
| Metric | Score |
| Average number of features changed | 8 (/19) |
| Average number of features changed with counterintuitive SHAP changes | 5 (/19) |
| Average distance to the profile | 0.049 (/1) |
| Average distance to the closest training point | 0 |



This weight was chosen according to the observed values of the training dataset to approximately center the means.

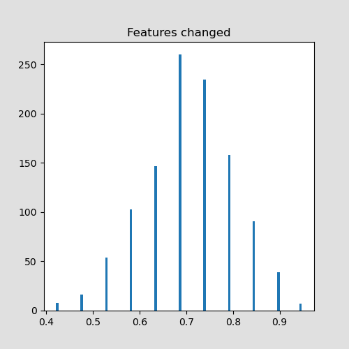
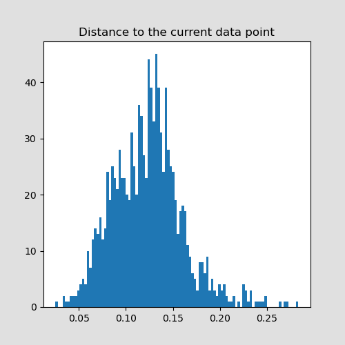
With tolerance:

|  |  |
| --- | --- |
| Metric | Score |
| Average number of features changed | 4 (/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 |  |



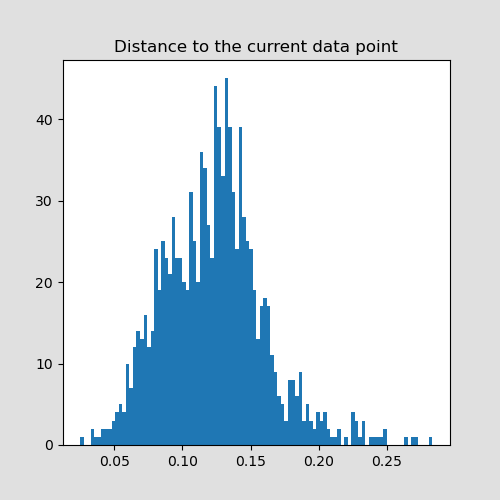
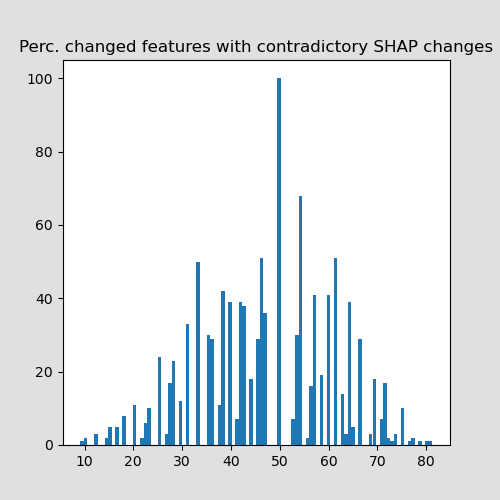
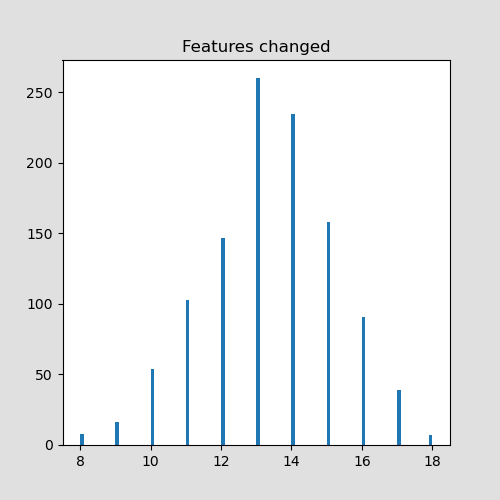
Now for weight == 6.

Why weight == 6? By multiplying by 6 we center the means of the two scores.

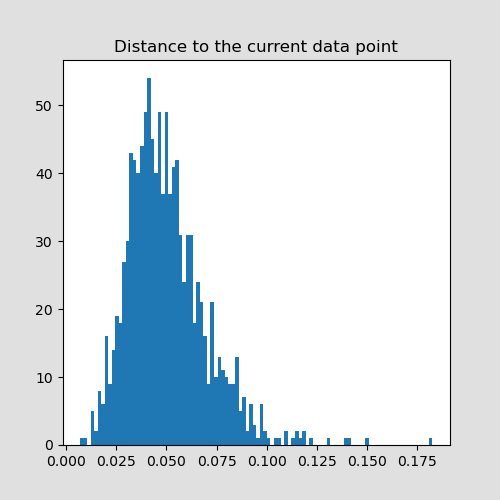
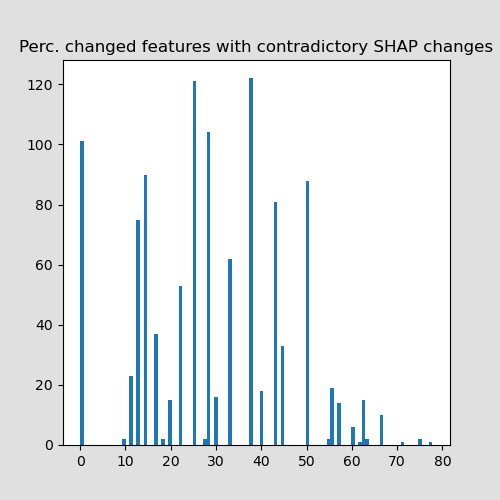
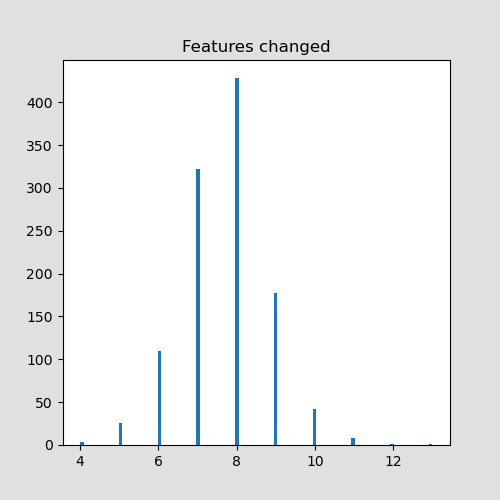
AVG

|  |  |
| --- | --- |
| Metric | Score |
| Average number of features changed | 13 (/19) |
| Average percentage of features changed with counterintuitive SHAP changes | 47% (~6/13) |
| Average distance to the profile | 0.123 (/1) |
| Average distance to the closest training point | 0 |



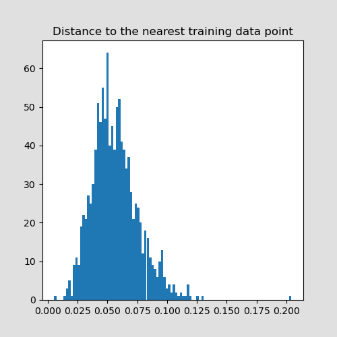
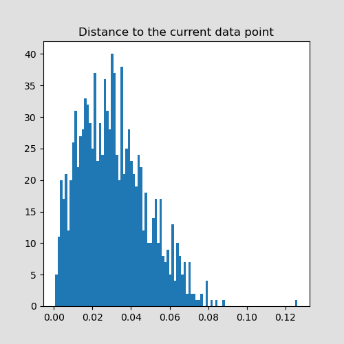
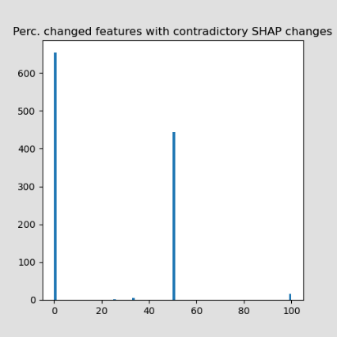
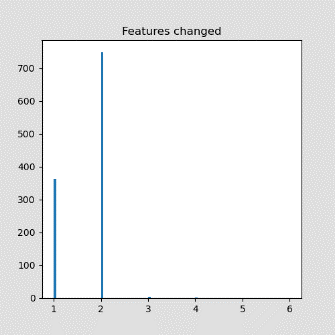
Pareto + no tolerance

|  |  |
| --- | --- |
| Metric | Score |
| Average number of features changed | 7 (/19) |
| Average percentage of features changed with counterintuitive SHAP changes | 29% (~2/7) |
| Average distance to the profile | 0.050 (/1) |
| Average distance to the closest training point | 0 |



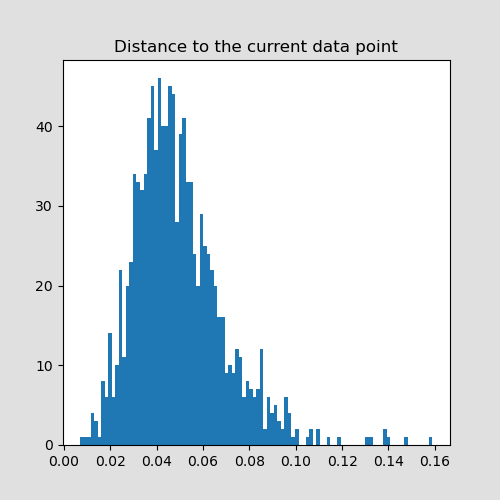
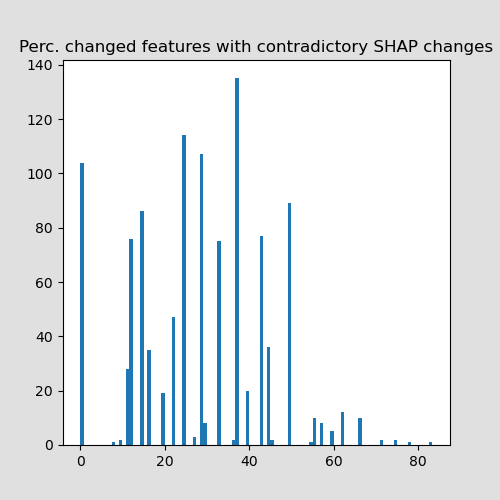
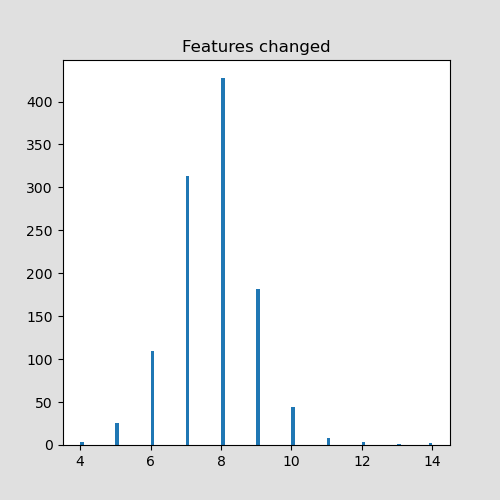
DiCE (33.5 minutes)

|  |  |
| --- | --- |
| Metric | Score |
| Average number of features changed | 2 (/19) |
| Average percentage of features changed with counterintuitive SHAP changes | 21% (~0/2) |
| Average distance to the profile | 0.031 (/1) |
| Average distance to the closest training point | 0.057 (/1) |
| Number of targets missed | 18 (/1118) |



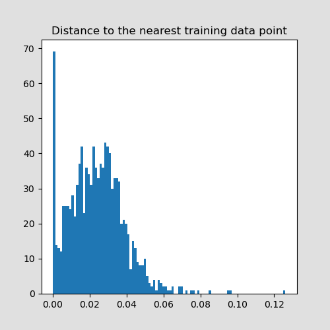
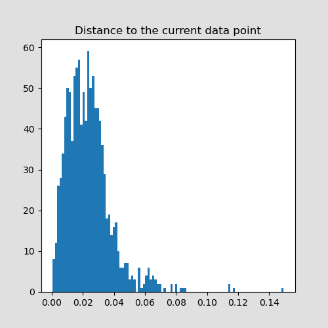
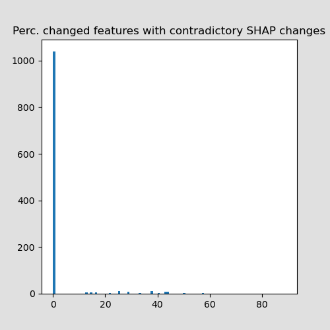
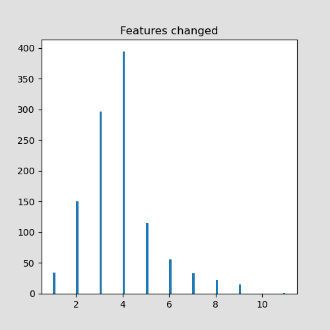
Weight (6) + no tolerance

|  |  |
| --- | --- |
| Metric | Score |
| Average number of features changed | 8 (/19) |
| Average percentage of features changed with counterintuitive SHAP changes | 29% (~2/8) |
| Average distance to the profile | 0.049 (/1) |
| Average distance to the closest training point | 0 (/1) |



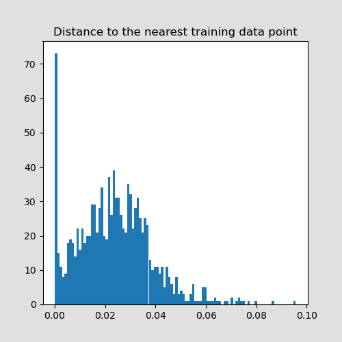
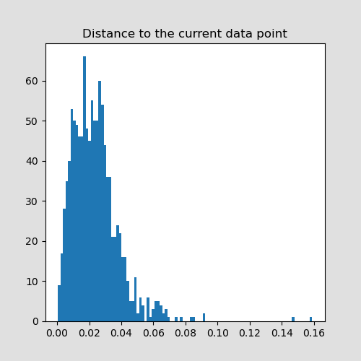
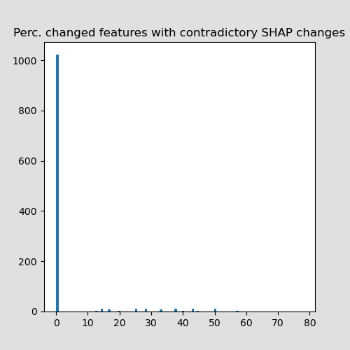
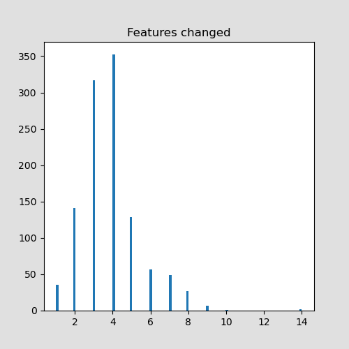
Pareto + filter

|  |  |
| --- | --- |
| Metric | Score |
| Average number of features changed | 4 (/19) |
| Average percentage of features changed with counterintuitive SHAP changes | 2% |
| Average distance to the profile | 0.024 (/1) |
| Average distance to the closest training point | 0.024 (/1) |



Weight (6) + filter

|  |  |
| --- | --- |
| Metric | Score |
| Average number of features changed | 4 (/19) |
| Average percentage of features changed with counterintuitive SHAP changes | 3% |
| Average distance to the profile | 0.024 (/1) |
| Average distance to the closest training point | 0.024 (/1) |



|  |  |  |
| --- | --- | --- |
| Metric |  | DiCE |
| Average number of features changed |  | 4 (/19) |
| Average percentage of features changed with counterintuitive SHAP changes |  | 3% |
| Average distance to the profile |  | 0.024 (/1) |
| Average distance to the closest training point |  | 0.024 (/1) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | DiCE | WhatIf | Pareto | Pareto + filter | Weighted | Weighted + filter |
| Avg. features changed | **1.73** | 8.43 | 7.70 | 3.93 | 7.88 | 4.05 |
| Avg perc. counter-intuitive | 21.3 | 27.7 | 27.9 | **2.93** | 27.1 | 2.98 |
| Avg. distance to profile | 0.03 | 0.04 | 0.04 | **0.02** | 0.04 | **0.02** |
| Avg. distance to training point | 0.06 | **0.00** | **0.00** | 0.02 | **0.00** | 0.01 |
| Avg. realism score | 0.25 | **1.00** | **1.00** | 0.90 | **1.00** | 0.90 |
| Targets missed | 14 | **0** | **0** | **0** | **0** | **0** |
| Minutes | 263 (4.4hrs) | 358 (6.0hrs) | 366 (6.1hrs) | 361 (6.0 hrs) | 362 (6.0 hrs) | 363 (6.0 hrs) |

Perhaps there could be a filter that shows the full example profile versus the feature changes that have helped towards the target prediction. In this way, we 1. Show a 100% real training data point, but also 2. Present the most influential changes.

We need to check if the proposed feature value change leads to a plausible profile. This is not achieved by the current L1/L2/MinMax-scaled/MAD-scaled distance measures, since they assume independence of features. When a CF is proposed with two relatively big feature value changes, this CF is scored to be further from the original profile, while a CF with single relatively small feature value change, is scored to be closer. What we miss here is a measure of how realistic these new feature combinations are.

might be more likely to occur in the training data, rather than a single relatively small feature value change.

For this reason, we deem perturbation methods unfit for this problem.

“*How likely is it to have feature value X1=v2 instead of X1=v1, in combination with all the other feature values of this profile?”*

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

Idea for the future: pick the CF profile that has changes in the least number of correlated features, because they are the most important to show.

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.