Exercise 1: Gower Distance

Gower's distance is one of the most popular ways of measuring the similarity or dissimilarity between observations in the presence of mixed-type variables:

$$d_G(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = \frac{1}{p}\sum_{j=1}^{p} \delta_G(x_j^{(1)}, x_j^{(2)}) \text{ with } \delta_G(x_j^{(1)}, x_j^{(2)}) = \begin{cases} \frac{1}{\widehat{R}_j}|x_j^{(1)} - x_j^{(2)}| & \text{if } x_j \text{ is numerical} \\ \mathbb{I}_{x_j^{(1)} \neq x_j^{(2)}} & \text{if } x_j \text{ is categorical} \end{cases}.$$

Here $\widehat{R}_j$ stands for the range of this feature (the "diameter" of the feature): $\widehat{R}_j := \max_i(x_j^{(i)}) - \min_i(x_j^{(i)})$. Despite its popularity, the Gower distance has a major drawback. In the following exercises, you should identify this problem and construct a solution.

a) Let us consider the following example that should illustrate the problem: We have a dataset with two features - sex and age. Sex has the values M = male and F = female, while age could have values between 15 and 90, but in this example we only consider 3 different values (15, 58, 90). Overall, the following feature combinations are possible: (F, 15), (F, 58), (F, 90), (M, 15), (M, 58), (M, 90). Imagine that we have an observation $\mathbf{x}^{(1)}$ =($\text{Sex}^{(1)}$, $\text{Age}^{(1)}$) = (F, 15). For this observation we want to measure the distance to all other possible observations $\mathbf{x}^{(2)}$. Derive the Gower distance d_G and feature-wise distance δ_G per row.

$\text{Sex}^{(1)}$	$\text{Sex}^{(2)}$	$\delta_G(\text{Sex}^{(1)}, \text{Sex}^{(2)})$	$\text{Age}^{(1)}$	$\text{Age}^{(2)}$	$\delta_G(\text{Age}^{(1)}, \text{Age}^{(2)})$	$d_G(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$
F	F		15	15		
F	F		15	58		
F	F		15	90		
F	M		15	15		
F	M		15	58		
F	M		15	90		

b) Imagine that the observation $\mathbf{x}^{(1)}$ = ($\text{Sex}^{(1)}$, $\text{Age}^{(1)}$) = (F, 15) is the data point we want to find counterfactuals for and imagine that all the other observations ($\text{Sex}^{(2)}$, $\text{Age}^{(2)}$) are the counterfactual candidates. Given the fully filled out table, which counterfactuals would you prefer? Do you see a tendency towards a group of counterfactuals with respect to the type of features? Describe why this is a problem.

c) An enhancement of the Gower distance is to use different weights w_j for the different features $j \in \{1, ..., p\}$: $d_G(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = \frac{1}{p}\sum_{j=1}^{p} w_j \delta_G(x_j^{(1)}, x_j^{(2)})$. How would you choose w_j in order to solve the above problem? Think about possible weighting schemes.

Exercise 2: LIME: Example

An insurance company wants to calculate monthly premiums for a disability insurance ("Berufsunfähigkeitsversicherung") with an ML model based on the pension, age, job type and marital status. An appropriate model $\hat{f}$ was already fitted using a large customer dataset but in order to be launched it needs approval by regulators. The regulators put the model to test by using LIME (with exponential kernel) to provide an explanation for typical and critical customers. One of these test instances $\mathbf{x}$ is a 21 year old woman displayed in the first row of the following table.

a) The regulators already generated new instances to fit the surrogate model. For simplicity, we assume that three instances are enough to fit the model. Fill out the missing fields in the following table, where $\phi_\sigma(\mathbf{z}) = \exp(-d(\mathbf{x}, \mathbf{z})^2/\sigma^2)$ is the exponential kernel similarity measure for a specific σ and $d(\cdot)$ as the Gower's distance.

   How does the kernel width σ influence the proximity measure? What would happen if the kernel width is set too small?

	pension	age	job type	marital status	$\hat{f}$	$d(\mathbf{x}, \mathbf{z}.)$	$\phi_{\sigma=0.15}(\mathbf{z}.)$	$\phi_{\sigma=0.5}(\mathbf{z}.)$
$\mathbf{x}$	1800	21	sedentary	single	30.6	-	-	-
$\mathbf{z}_1$	1600	21	sedentary	married	25.8	0.25	0.06	
$\mathbf{z}_3$	2200	32	sedentary	married	85.2	0.32	0.01	
$\mathbf{z}_2$	1200	23	physically	single	74.9	0.49	0.00	0.38

b) The regulatories fit two different local surrogate models (g_1 and g_2) on the re-weighted data (here, three observations). The following table compares the prediction of $\hat{f}$ to the ones of the two surrogate models for the three instances.

	$\hat{f}$	g_1	g_2
$\mathbf{x}$	30.6	34.8	31.1
$\mathbf{z}_1$	25.8	28	26.1
$\mathbf{z}_3$	85.2	105	92.7
$\mathbf{z}_2$	74.9	90	68.9

Which of the two surrogate models do you prefer? Compute the local faithfulness for both surrogate models using the weights from a) with $\sigma = 0.15$.
Hint: consider $L(\hat{f}, g, \phi_{\mathbf{x}})$ from the lecture.

c) The surrogate model g_1 corresponds to a linear model using all three features, while g_2 corresponds to a random forest with 500 trees. Would you still prefer the model you chose in b)?

d) Discuss whether for the faithfulness assessment in b) it makes sense to use a new sampled dataset instead of the one the local surrogate model was fitted on.

Exercise 3: LIME: Implementation
In the following, you are guided to implement LIME to interpret a Support Vector Machine (SVM). We use **two** (numeric) features and explore LIME on a multiclass classification problem with only two (numerical) features. The associated files for this exercise are *lime.py* and *datasets.py* for Python or *lime.R* for R, depending on your preferred programming language, which you can find on Moodle. In these files, helper functions for plotting (`get_grid()`, `plot_grid()` and `plot_points_in_grid()`) were already implemented.

a) Inspect Implemented Functions

First of all, make yourself familiar with the already implemented functions in the template files.

- The function `get_grid()` prepares data to visualize the feature space. It creates a $N \times N$ grid, and every point in this grid is associated with a value. This value is obtained by the model's predict method.
- The function `plot_grid()`, visualizes the prediction surface.
- The created plot is an input to the function `plot_points_in_grid()`, which adds given data points to the plot.

b) Sample Points

Your first implementation task is to sample points, which are later used to train the local surrogate model. Complete `sample_points()` by randomly sampling from a uniform distribution. Consider the lower and upper bounds from the input datapoints.

Hint: In Python, you can use the method `dataset.get_configspace().get_hyperparameters_dict()` implemented in the file *dataset.py* to retrieve the lower and upper values. For an example, have a look at the already implemented function `get_grid()`.

c) Weight Points

Given a selected point $\mathbf{x}$ and the sampled points Z from the previous task, we now want to weight the points. Use the following equation with d as Euclidean distance to calculate the weight of a single point $\mathbf{z} \in Z$:

$$\phi_{\mathbf{x}}(\mathbf{z}) = \exp\left(-\frac{d(\mathbf{x}, \mathbf{z})^2}{\sigma^2}\right). \tag{1}$$

To make plotting easier later on, the weights should be normalized between zero and one. Finally, return the normalized weights in `weight_points()`.

d) Fit Local Surrogate Model

Finally, fit a decision tree with training data and weights. Return the fitted tree in the function `fit_explainer_model()`. What could be problematic?