Noise generation

The generation of noise can be characterized in different ways. Firstly, it can be characterized by its distribution, which can be, for example, normal (Gaussian). Secondly, noise can be characterized by where the noise is introduced, which can be in the **input attributes**, in the **output class**, in the **training data**, in the **test data**, or a combination of all of these. Finally, noise can be characterized by distinguishing if the magnitude of the generated noise values is relative to each data value of each variable, or if it is relative to the min, max and standard deviation for each variable.

I. Noise Generation in the input attributes of the training data

The noise is generated and introduced in the training data in the following manner. Firstly, generate $\boldsymbol{d_v = floor(d * R_n)}$ random numbers with a Gaussian distribution and within a range between the **max** and **min** of the corresponding variable, to modify the data values where **d** is the number of attributes, and $\boldsymbol{0 \leq R_n \leq 1}$ is the fraction of noisy attributes. Then generate $\boldsymbol{N_c = N_{tr} * R_C}$ random numbers with a uniform distribution within the range 1 to the number of cases where $\boldsymbol{N_{tr}}$ is the number of training data, $\boldsymbol{0 \leq R_n \leq 1}$ is the fraction of noisy training data. $\boldsymbol{N_c}$ indicates the case whose data value is to be modified.

Python code for generating noisy data with Gaussian distribution

```
import scipy.stats as ss
import numpy as np
# for feature f_i with values of {0,1}
# x = np.arange( np.min(f_i), np.max(f_i)+1 )
x = np.arange( 2 )
xU, xL = x + 0.5, x - 0.5
prob = ss.norm.cdf(xU, scale = 3) - ss.norm.cdf(xL, scale = 3)
prob = prob / prob.sum() #normalize the probabilities so their sum is 1
f_i[R_c] = np.random.choice(x, size = R_c, p = prob)
```

Matlab code for generating noisy data with Gaussian distribution

```
% for feature f_i with values of {0,1}
x = randperm(2)-1;
xU = x + 0.5;
xL = x - 0.5;
sigma = 3;
pd = makedist('Normal',0,sigma);
prob = cdf(pd, xU) - cdf(pd, xL);
prob = prob / sum(prob);
nums = randsrc(10,1,[x; prob]);
```

Apply the process of noise generation for $\boldsymbol{R_n} = 0.1, 0.2, 0.3\ , 0.4, 0.5$ and $\boldsymbol{R_C} = 0.1, 0.2, 0.3\ , 0.4, 0.5$**.** Report the performance of the model for each value of $\boldsymbol{R_n}$ and $\boldsymbol{R_c}$.

II. Generation of noise in the input attributes of the test data

The noise is generated and introduced in the test data in the following manner. Firstly, generate $\boldsymbol{d_v} = floor(\boldsymbol{d} * \boldsymbol{R_n})$ random numbers with a Normal distribution and within a range between the **max** and **min** of the corresponding variable, to modify the data values where **d** is the number of attributes, and $\mathbf{0} \leq \boldsymbol{R_n} \leq \mathbf{1}$ is the fraction of noisy attributes. Then generate $\boldsymbol{N_c} = \boldsymbol{N_{te}} * \boldsymbol{R_C}$ random numbers with a uniform distribution within the range 1 to the number of cases where $\boldsymbol{N_{te}}$ is the number of test data, $\mathbf{0} \leq \boldsymbol{R_n} \leq \mathbf{1}$ is the fraction of noisy training data. $\boldsymbol{N_c}$ indicates the case whose data value is to be modified.

Apply the process of noise generation for $\boldsymbol{R_n} = 0.1, 0.2, 0.3\ , 0.4, 0.5$ and $\boldsymbol{R_C} = 0.1, 0.2, 0.3\ , 0.4, 0.5$**.** Report the performance of the model for each value of $\boldsymbol{R_n}$ and $\boldsymbol{R_c}$.