Object-Oriented Programming

Applied Machine Learning in Engineering - Exercise 02

TU Berlin, Winter Term 2023/24

Prof. Dr.-Ing. Merten Stender – `merten.stender@tu-berlin.de`

This exercise covers data encoding and teaches how to use object-oriented programming.

Validate your implementation using a small example: One-hot encode the data contained in `bearning_faults.csv` provided in ISIS.

To read that file, use the following Numpy code:
`np.genfromtxt('bearing_faults.csv',dtype='str', delimiter=",")`.

If you want to know more about the bearing faults, search the web for *Bearing Failure: Causes and Cures* by Schaeffler.

Implement One-Hot-Encoding using Functional Programming

Implement one-hot encoding for nominal attribute types using functional programming. You can use test-driven development but it is not a must.

(a) Implement a function `fit()`, a function `encode()` (mapping categorical values to arrays), and a function `decode()` (mapping arrays back to categorical values).

(b) Use a dictionary for mapping categories to indices, like `class_map=dict()`, `class_map['class 1']=1`. You can get the value `class 1` by calling `class_map[1]` (i.e. the key of the dictionary).

(c) validate your implementation on the given data set

Implement One-Hot-Encoding using Object-Oriented Programming

Implement one-hot encoding for nominal attribute types using object-oriented programming. You can use test-driven development but it is not a must.

(a) Implement a class `OneHotEncoder()` that has methods `fit()`, `encode()` and `decode()` for performing the encoding and decoding.

(b) Re-use the implementations from the first task.

(c) validate your implementation on the given data set