

UMB W05: przetwarzanie wstępne

```
import pandas as pd
import matplotlib.pyplot as plt

import umb_tools as umb
```

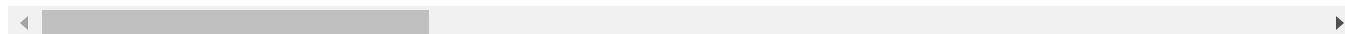
```
# konfiguracja
plt.rcParams["figure.figsize"] = [5, 4]
pd.set_option("display.float_format", lambda x: "%.4f" % x)
```

1. Wczytanie zbioru danych

```
# odczyt pliku TSV (zwracane są: zbiór danych w postaci DataFrame biblioteki Pandas oraz lista kolumn)
(df, c_names) = umb.read_data("data/BreastCancer.txt")
df
```

| #BreastCancer | labels | radius_mean | texture_mean | perimeter_mean | area_mean | smoothness_mean | cor |
|---------------|--------|-------------|--------------|----------------|-----------|-----------------|-----|
| 842302 | 1 | 17.9900 | 10.3800 | 122.8000 | 1001.0000 | 0.1184 | |
| 874858 | 1 | 14.2200 | 23.1200 | 94.3700 | 609.9000 | 0.1075 | |
| 875263 | 1 | 12.3400 | 26.8600 | 81.1500 | 477.4000 | 0.1034 | |
| 87556202 | 1 | 14.8600 | 23.2100 | 100.4000 | 671.4000 | 0.1044 | |
| 875938 | 1 | 13.7700 | 22.2900 | 90.6300 | 588.9000 | 0.1200 | |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 8910720 | 2 | 10.7100 | 20.3900 | 69.5000 | 344.9000 | 0.1082 | |
| 8910506 | 2 | 12.8700 | 16.2100 | 82.3800 | 512.2000 | 0.0943 | |
| 8910499 | 2 | 13.5900 | 21.8400 | 87.1600 | 561.0000 | 0.0796 | |
| 8912055 | 2 | 11.7400 | 14.0200 | 74.2400 | 427.3000 | 0.0781 | |
| 92751 | 2 | 7.7600 | 24.5400 | 47.9200 | 181.0000 | 0.0526 | |

569 rows × 31 columns



2. Normalizacja danych

```
# pobranie dwóch cech z macierzy danych
data = df.iloc[:, [1, 10]]

data.describe()
```

| #BreastCancer | radius_mean | fractal_dimension_mean |
|---------------|-------------|------------------------|
| count | 569.0000 | 569.0000 |
| mean | 14.1273 | 0.0628 |
| std | 3.5240 | 0.0071 |
| min | 6.9810 | 0.0500 |
| 25% | 11.7000 | 0.0577 |
| 50% | 13.3700 | 0.0615 |
| 75% | 15.7800 | 0.0661 |
| max | 28.1100 | 0.0974 |

Standaryzacja

```
from sklearn.preprocessing import StandardScaler
```

```
data = df.iloc[:, [1, 10]]

norm_data = StandardScaler().fit_transform(data)

pd.DataFrame(norm_data, columns=data.columns).describe()
```

| #BreastCancer | radius_mean | fractal_dimension_mean |
|---------------|-------------|------------------------|
| count | 569.0000 | 569.0000 |
| mean | -0.0000 | 0.0000 |
| std | 1.0009 | 1.0009 |
| min | -2.0296 | -1.8199 |
| 25% | -0.6894 | -0.7226 |
| 50% | -0.2151 | -0.1783 |
| 75% | 0.4694 | 0.4710 |
| max | 3.9713 | 4.9109 |

Skalowanie min-max

```
from sklearn.preprocessing import MinMaxScaler
```

```
data = df.iloc[:, [1, 10]]
```

```
norm_data = MinMaxScaler().fit_transform(data)
```

```
pd.DataFrame(norm_data, columns=data.columns).describe()
```

| #BreastCancer | radius_mean | fractal_dimension_mean |
|---------------|-------------|------------------------|
| count | 569.0000 | 569.0000 |
| mean | 0.3382 | 0.2704 |
| std | 0.1668 | 0.1487 |
| min | 0.0000 | 0.0000 |
| 25% | 0.2233 | 0.1630 |
| 50% | 0.3024 | 0.2439 |
| 75% | 0.4164 | 0.3404 |
| max | 1.0000 | 1.0000 |

Znaczenie normalizacji danych

```
# pobranie etykiet klas
```

```
labels = df.iloc[:, 0].to_numpy()
```

```
# pobranie macierzy danych
```

```
data = df.iloc[:, 1:].to_numpy()
```

```
# standaryzacja
```

```
data_norm = StandardScaler().fit_transform(data)
```

```
# wykresy PCA
```

```
fig, ax = plt.subplots(1, 2, figsize=(10, 4))
```

```
umb.pca_plot(data, labels, c_names, ax[0], "Without standarization")
```

```
umb.pca_plot(data_norm, labels, c_names, ax[1], "With standarization")
```

```
plt.show()
```

