

SIECI NEURONOWE

Raport z laboratorium 1

Student: Van Hien Le - 257795

Narzędzia i technologie

- Python, Pandas, Numpy
- Jupyter Notebook

Dataset

Zbiór danych używany w tym ćwiczeniu to zbiór danych chorób serca, do którego można uzyskać dostęp pod adresem <https://archive.ics.uci.edu/dataset/45/heart+disease>.

Analiza eksploracyjna

1. Import danych

Instalacja bibliotek i pobieranie zbioru danych:

```
!pip install ucimlrepo
!pip install scikit-learn
```

```
from ucimlrepo import fetch_ucirepo, list_available_datasets
list_available_datasets()
```

```
heart_disease = fetch_ucirepo(id=45)
X = heart_disease.data.features
y = heart_disease.data.targets
```

Pierwsze 10 przykładów zbioru danych:

```
dataset = pd.concat([X, y], axis=1)
dataset[0:10]
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
0	63	1	1	145	233	1	2	150	0	2.3	3	0.0	6.0	0
1	67	1	4	160	286	0	2	108	1	1.5	2	3.0	3.0	2
2	67	1	4	120	229	0	2	129	1	2.6	2	2.0	7.0	1
3	37	1	3	130	250	0	0	187	0	3.5	3	0.0	3.0	0
4	41	0	2	130	204	0	2	172	0	1.4	1	0.0	3.0	0
5	56	1	2	120	236	0	0	178	0	0.8	1	0.0	3.0	0
6	62	0	4	140	268	0	2	160	0	3.6	3	2.0	3.0	3
7	57	0	4	120	354	0	0	163	1	0.6	1	0.0	3.0	0
8	63	1	4	130	254	0	2	147	0	1.4	2	1.0	7.0	2
9	53	1	4	140	203	1	2	155	1	3.1	3	0.0	7.0	1

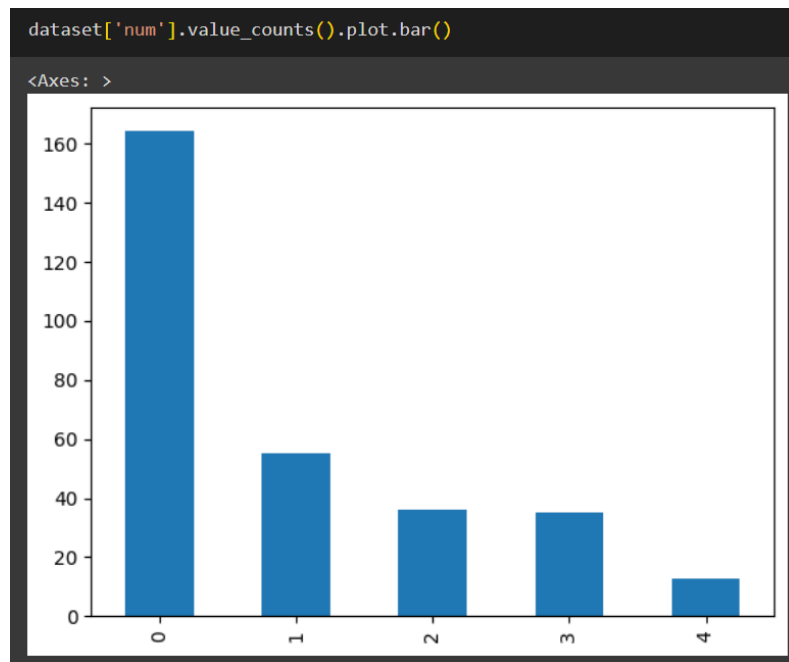
Kształt zbioru danych:

```
dataset.shape
```

```
(303, 14)
```

2. Balans klas

- Czy zbiór jest zbalansowany pod względem liczby próbek na klasy?



Zbiór danych nie jest zbalansowany pod względem liczby próbek na klasę. Liczba próbek klasy 0 jest znacznie wyższa niż pozostałych.

3. Cechy liczbowe

- Jakie są średnie i odchylenia cech liczbowych?

Cechy liczbowe w zbiorze danych to "age", "trestbps", "chol", "thalach" i "oldpeak".

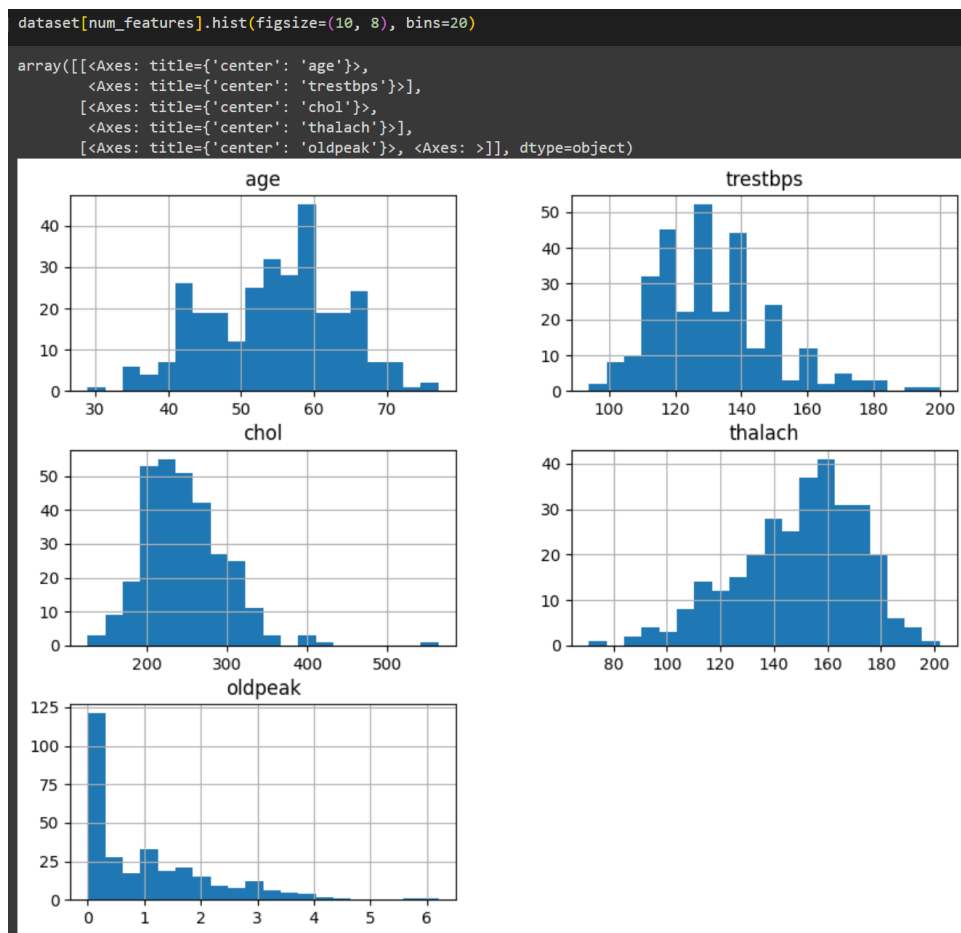
Średnie i odchylenia:

```
num_features = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
dataset[num_features].describe().loc[['mean', 'std']]
```

	age	trestbps	chol	thalach	oldpeak
mean	54.438944	131.689769	246.693069	149.607261	1.039604
std	9.038662	17.599748	51.776918	22.875003	1.161075

- Dla cech liczbowych: czy ich rozkład jest w przybliżeniu normalny?

Histogram cech liczbowych:

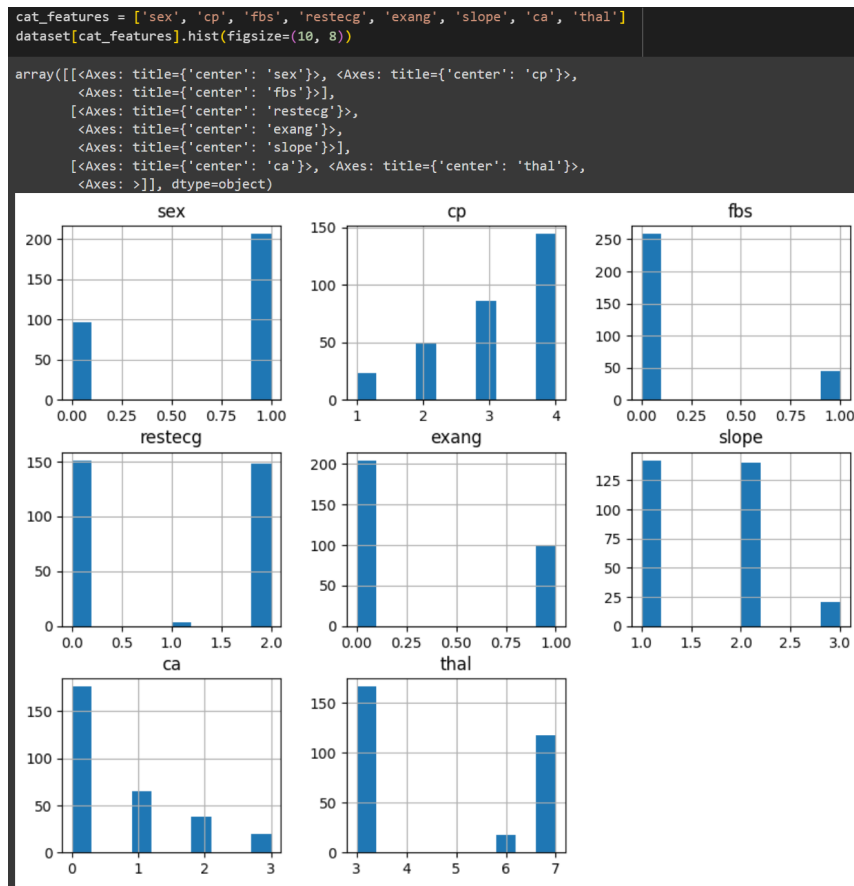


Patrząc na histogramy, możemy rozpoznać, że rozkład normalny istnieje w cechach "age", "chol" i "thalach", ale wydaje się, że nie istnieje w cechach "trestbps" i "oldpeak".

4. Cechy kateryczne

- Dla cech katerycznych: czy rozkład jest w przybliżeniu równomierny?
Cechy kateryczne w zbiorze danych to "sex", "cp", "fbs", "restecg", "exang", "slope", "ca" i "thal".

Histogram cech katerycznych:



Patrząc na histogramy, możemy zauważyć, że cechy katagoryczne tego zbioru danych nie mają równomiernego rozkładu.

5. Brakujące dane

- Czy w zbiorze danych brakuje jakichś danych?

```
dataset.isnull().sum()

age      0
sex      0
cp       0
trestbps 0
chol     0
fbs      0
restecg  0
thalach  0
exang    0
oldpeak  0
slope    0
ca       4
thal     2
num      0
dtype: int64
```

W zbiorze danych brakuje 6 wartości, 4 w kolumnie "ca" i 2 w kolumnie "thal".

- Jaką strategię można zastosować, aby zastąpić brakujące dane?

Dla cechy numerycznej "ca" możemy zastąpić brakujące dane wartościami mediany. Dla cechy katagorycznej "thal" możemy zastąpić brakujące wartości najczęściej występującą wartością tej cechy:

```

ca_median = dataset['ca'].median()
ca_median

0.0

thal_mode = dataset['thal'].mode()[0]
thal_mode

3.0

dataset['ca'].fillna(ca_median, inplace=True)
dataset['thal'].fillna(thal_mode, inplace=True)

```

Kod przekształcający zbiór danych w macierz cech liczbowych (przykłady x cechy)

W przypadku cech kategorycznych używam 'one hot encoder' do kodowania wartości:

```

mapped_column_names = {'cp_1': 'cp_typical_angina', 'cp_2': 'cp_atypical_angina', 'cp_3':
'cp_non_anginal_pain', 'cp_4': 'cp_asymptomatic',
                        'restecg_0': 'restecg_normal', 'restecg_1': 'restecg_wave_abnormality',
'restecg_2': 'restecg_definite_lvh',
                        'thal_3': 'thal_normal', 'thal_6': 'thal_fixed', 'thal_7':
'thal_reversible',
                        'slope_1': 'slope_upsloping', 'slope_2': 'slope_flat', 'slope_3':
'slope_downsloping'}

```

```

column_names = ['cp', 'restecg', 'thal', 'slope']
dataset[column_names] = dataset[column_names].astype('int64')
dataset_encoded = pd.get_dummies(dataset, columns=column_names)
dataset_encoded.rename(columns=mapped_column_names, inplace=True)

```

Kształt zbioru danych po przekształceniu to (303, 23)