Politechnika Wrocławska
Wydział Informatyki i Telekomunikacji

Sieci neuronowe

Raport ćw. 1-4

Autor

Szymon Sawczuk

nr albumu: **260287**

kierunek: **Informatyka Stosowana**

12 października 2023

Spis treści

1 Ćwiczenie 1 - Analiza danych

Celem ćwiczenia było zapoznanie (bądź przypomnienie) się z bibliotekami i narzędziami, które wykorzystywane są do uczenia maszynowego, eksploracji danych oraz ewaluacji sieci neuronowych, a także analiza zbioru danych wykorzystywanych do tego i dalszych ćwiczeń. (*Heart Disease Dataset*, 1988)

1.1 Biblioteki użyte w tym ćwiczeniu

W tym ćwiczeniu wykorzystałem 3 biblioteki dostępne dla języka python:

- pandas - biblioteka pozwalająca na łatwe tworzenie zbiorów danych oraz ich eksploracje i modyfikacje
- matplotlib - do tworzenia wykresów danych
- seaborn - dla zaawansowanych wizualizacji danych np. mapy ciepła

Zapoznałem się także z bibliotekami, które będą potrzebne do kolejnych ćwiczeń: numpy (biblioteka do operacji na wielowymiarowych tabelach/macierzach), Scikit-learn (dająca implementacje algorytmów do preprocessing'u oraz algorytmów uczenia maszynowego).

1.2 Eksploracja danych

Po załadowaniu danych poprzez prosty skrypt podany na stronie zbioru danych (*Heart Disease Dataset*, 1988), uzyskałem 14 kolumn w zbiorze danych:

- age (liczbowa) - wiek osoby (lata)
- sex (kategoryczna) - płeć osoby (0 - kobieta, 1 - mężczyzna)
- cp (kategoryczna) - typ bólu klatki piersiowej (wartości 1-4)
- trestbps (liczbowa) - ciśnienie krwi w spoczynku (mmHg)
- chol (liczbowa) - poziom cholesterolu w surowicy (mg/dl)
- fbs (kategoryczna) - poziom cukru we krwi na czczo (0 - nie, 1 - tak)
- restecg (kategoryczna) - wynik elektrokardiografii w spoczynku (0 - normalny, 1 - ST-T anormalność, 2 - hipertrofia)
- thalach (liczbowa) - maksymalne tętno osiągnięte podczas testu wysiłkowego
- exang (kategoryczna) - dławica wysiłkowa (0 - nie, 1 - tak)
- oldpeak (liczbowa) - depresja odcinka ST wywołana przez ćwiczenia w stosunku do odpoczynku
- slope (kategoryczna) - nachylenie odcinka ST podczas ćwiczeń (0 - wnoszące, 1 - płaskie, 2 - opadające)
- ca (liczbowa) - liczba głównych naczyń (0-3), podczas badania fluoroskopowego
- thal (kategoryczna) - rodzaj defektu (3 - normalny, 6 - uleczony defekt, 7 - odwracalny defekt)
- num - obecność choroby serca (0 - brak, 1,2,3,4 - obecność (czym większa wartość tym poważniejsza choroba))

Dane składają się z 303 próbek oraz 13 cech, kolumna num określa nam obecność choroby serca, albo jej brak.

```
heart_data.sample(10)
✓ 0.0s
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
261	58	0	2	136	319	1	2	152	0	0.0	1	2.0	3.0	3
169	45	0	2	112	160	0	0	138	0	0.0	2	0.0	3.0	0
40	65	0	4	150	225	0	2	114	0	1.0	2	3.0	7.0	4
241	41	0	2	126	306	0	0	163	0	0.0	1	0.0	3.0	0
12	56	1	3	130	256	1	2	142	1	0.6	2	1.0	6.0	2
259	57	1	2	124	261	0	0	141	0	0.3	1	0.0	7.0	1
201	64	0	4	180	325	0	0	154	1	0.0	1	0.0	3.0	0
246	58	1	4	100	234	0	0	156	0	0.1	1	1.0	7.0	2
99	48	1	4	122	222	0	2	186	0	0.0	1	0.0	3.0	0
255	42	0	3	120	209	0	0	173	0	0.0	2	0.0	3.0	0

Rysunek 1: 10 losowych przykładowych danych po wczytaniu

Pierwszą rzeczą jaką zbadałem było zbalansowanie danych względem liczby próbek w klasie.

```
heart_data["num"].value_counts()
[6]  ✓ 0.0s

...  num
     0    164
     1     55
     2     36
     3     35
     4     13
     Name: count, dtype: int64
```

Rysunek 2: Liczba próbek w klasach zbioru

Wyniki wskazują na brak zbalansowania danych pod względem liczby próbek na klasy. 164 próbki (około 54%) są próbkami zdrowych pacjentów (bez wykazanych problemów z sercem). Natomiast osób z zdiagnozowanymi najpoważniejszymi chorobami serca (klasa 4) jest tylko 4%. Rozwiązanie tego problemu wytłumaczone zostanie w następnym podrozdziale.

Następnym elementem badanym były wartości średnie oraz odchylenia standardowe cech liczbowych zbioru.

```
heart_data[num_features].mean()
✓ 0.0s

age          54.438944
trestbps    131.689769
chol        246.693069
thalach     149.607261
oldpeak       1.039604
ca            0.672241
dtype: float64
```

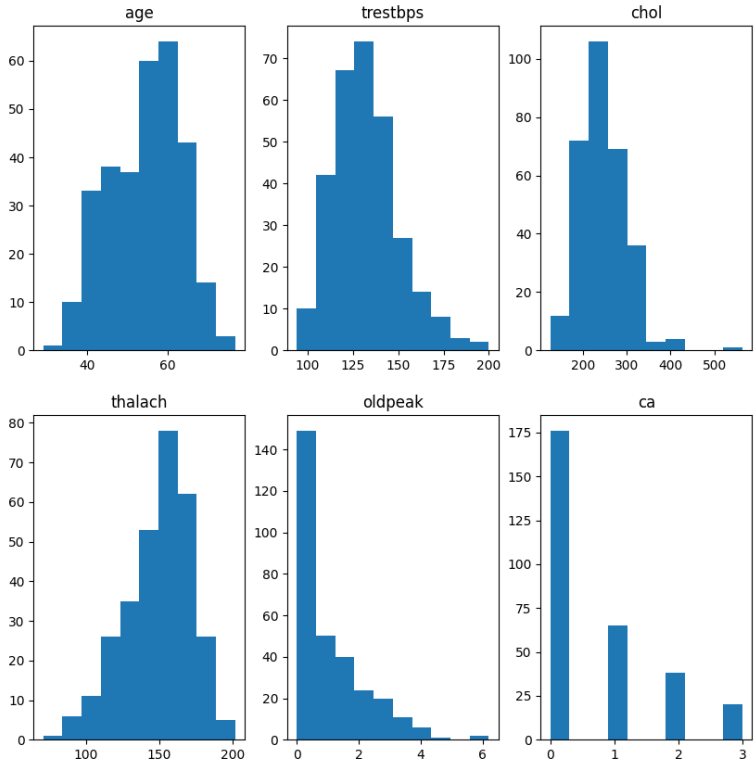
```
heart_data[num_features].std()
✓ 0.0s

age          9.038662
trestbps    17.599748
chol        51.776918
thalach     22.875003
oldpeak      1.161075
ca           0.937438
dtype: float64
```

Rysunek 3: Wartości średnie oraz odchylenia standardowe cech liczbowych

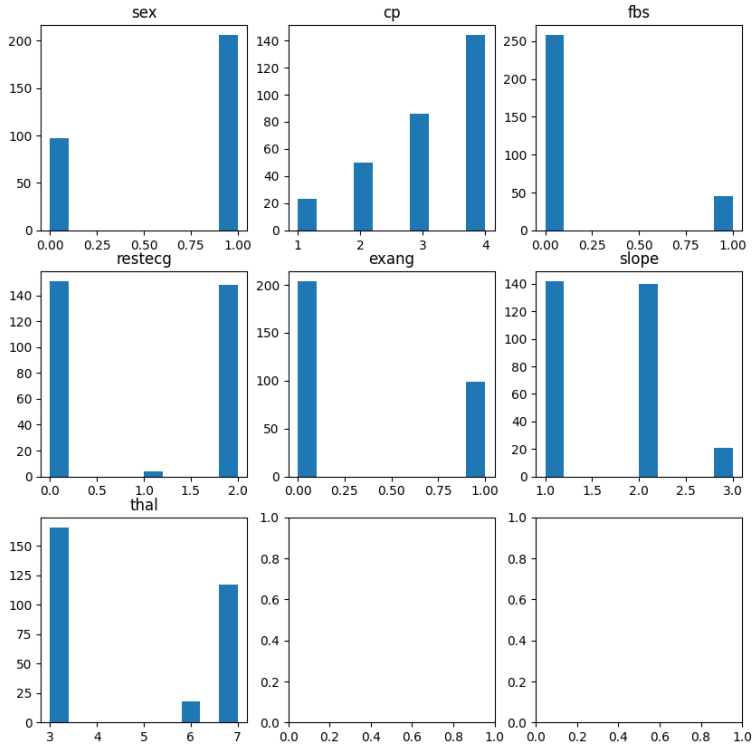
Dla przykładu średnia wartość wieku w zbiorze danych wynosi 54 lata, a około 70% danych mieści się między wiekiem 45, a 63. Widać już tutaj potencjalne rozkłady niektórych cech np. wieku. Natomiast histogramy wykazane poniżej wykazują, że cechy wieku, ciśnienia krwi w spoczynku, poziomu cholesterolu oraz maksymalnego osiągniętego tętna układają się w przybliżeniu zgodnie z wykresem

Gaussa, zatem posiadają one rozkłady normalne. Dane depresji odcinka ST (oldpeak) oraz liczba naczyń zaobserwowanych poprzez fluoroskopię (ca) nie wykazują rozkładu normalnego, bardziej rozkład wykładniczy.



Rysunek 4: Histogramy cech liczbowych

Weźmy teraz pod lupę cechy kategoryczne i czy są one w przybliżeniu równomierne.



Rysunek 5: Histogramy cech kategorycznych

Na powyższych histogramach cech kategorycznych nie widać, aby jakakolwiek cecha miała zrównoważone dane. Najbliżej jednak takiego rozkładu równomiernego są cechy danych elektrokardiograficznych (restecg) oraz nachylenie odcinka ST (slope). Z danych nierównomiernych widać np. że większą ilością badanych byli mężczyźni.

W zbiorze odnalazłem 2 cechy, które posiadają wartości puste jest to ca oraz thal. Łącznie wartości pustych jest 6.

```
heart_data[heart_data["ca"].isnull()]
✓ 0.0s
```

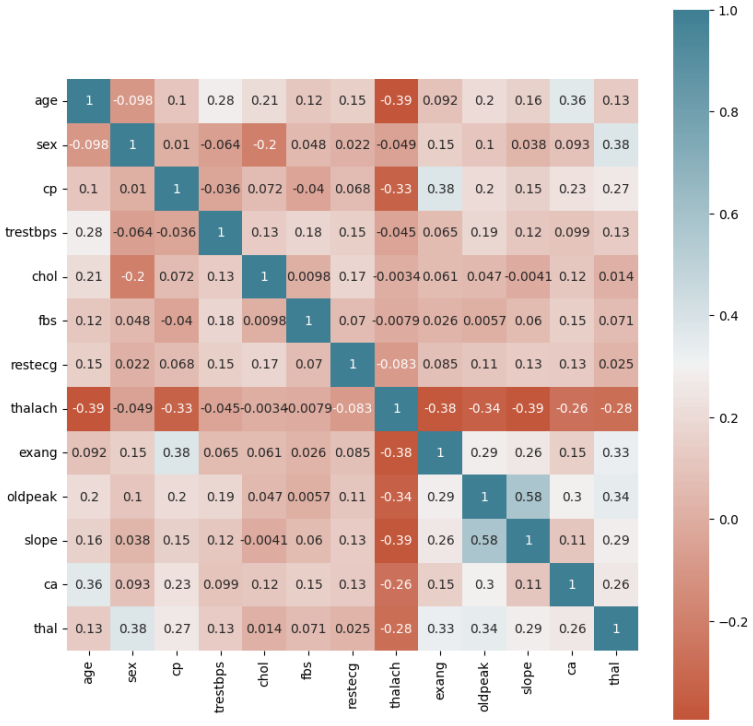
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
166	52	1	3	138	223	0	0	169	0	0.0	1	NaN	3.0	0
192	43	1	4	132	247	1	2	143	1	0.1	2	NaN	7.0	1
287	58	1	2	125	220	0	0	144	0	0.4	2	NaN	7.0	0
302	38	1	3	138	175	0	0	173	0	0.0	1	NaN	3.0	0

```
heart_data[heart_data["thal"].isnull()]
✓ 0.0s
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
87	53	0	3	128	216	0	2	115	0	0.0	1	0.0	NaN	0
266	52	1	4	128	204	1	0	156	1	1.0	2	0.0	NaN	2

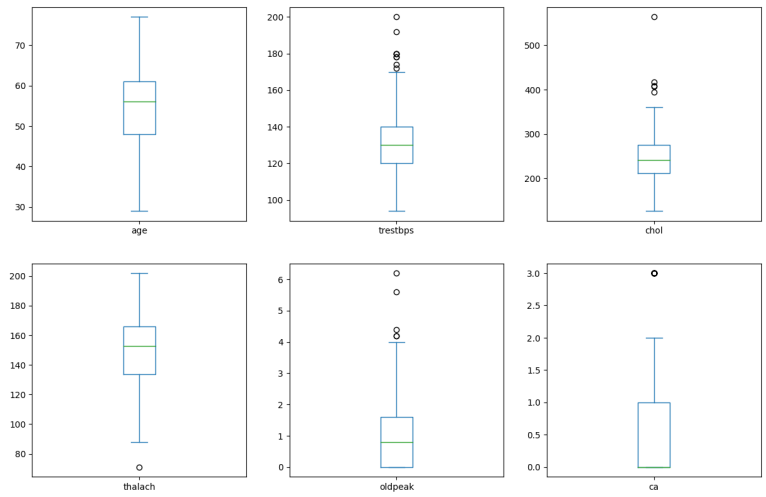
Rysunek 6: Wartości puste zbioru

Jest wiele sposobów na rozwiązanie tego problemu np. uzupełnienie brakujących danych w sposób sztuczny używając mediany, albo algorytmu k-najbliższych sąsiadów (*K najbliższych sąsiadów*, 2022). Natomiast z powodu małej ilości danych brakujących (około 2% danych), możemy najprościej usunąć te dane ze zbioru, bez znaczącej utraty informacji.



Rysunek 7: Wykres ciepła dla korelacji między cechami

Mapy ciepła zamieszczone powyżej pokazują poziom korelacji cech danych. Przy wysokich wartościach korelacji możnaby rozważyć usunięcie jednej z tych cech np. (slope i oldpeak), mogłoby to pomóc w uzykaniu lepszych wyników nauczania. Warto jednak rozważyć także sens merytoryczny tych dwóch cech, czy jednak nie są one znaczące dla całego modelu.



Rysunek 8: Wykresy pudełkowe dla cech liczbowych

Powyższe wykresy pudełkowe wskazują nam na rozłożenie wartości danych cech. Widzimi, że dane ca, oldepeak są w mniejszym zakresie niż np. wiek. Takie dane o małych zakresach mogą zostać przykryte w niektórych modelach przez cechy o większych zakresach. Warto też przyjrzeć się danym odbiegającym od kwartyli cechy (można rozważyć ich usunięcie).

1.3 Przygotowanie macierzy cech liczbowych

Po wyciągnięciu cech liczbowych ze zbioru danych zająłem się rozwiązaniem problemu braku zbilansowania próbek względem klasyfikacji zbioru. Zdecydowałem na naprawę braku zbalansowania próbek poprzez zmniejszenie klasyfikacji do klasyfikacji binarnej (0 - zdrowy, 1 - choroba serca), ponieważ klasy 1-4 oznaczały inne stopnie problemów z sercem, które można na potrzeby modelu budowanego zmniejszyć do tej samej klasyfikacji. Owe rozwiązanie pozwoliło także na zmniejszenie ilości danych potrzebnych do usunięcia, aby klasyfikacje były zbilansowane. W wyniku uzyskałem zmniejszony zbiór do 276 próbek, ale ze zbilansowanymi próbkami względem klas.

```
# repearing of the inbalnace in classification and removing null values
df["num"] = df["num"].replace([2, 3, 4], 1) #change classes to binary classification
print(df["num"].value_counts())

#get null values of ca and remove them
null_idx = df[heart_data["ca"].isnull()].index
print(null_idx)
df = df.drop(null_idx)
df = df.reset_index(drop=True)
print(df["num"].value_counts())

# balance classes to same amount 138
random_idx = df.query("num == 0").sample(df["num"].value_counts()[0] - df["num"].value_counts()[1]).index
df = df.drop(random_idx)
df = df.reset_index(drop=True)
print(df["num"].value_counts())
```

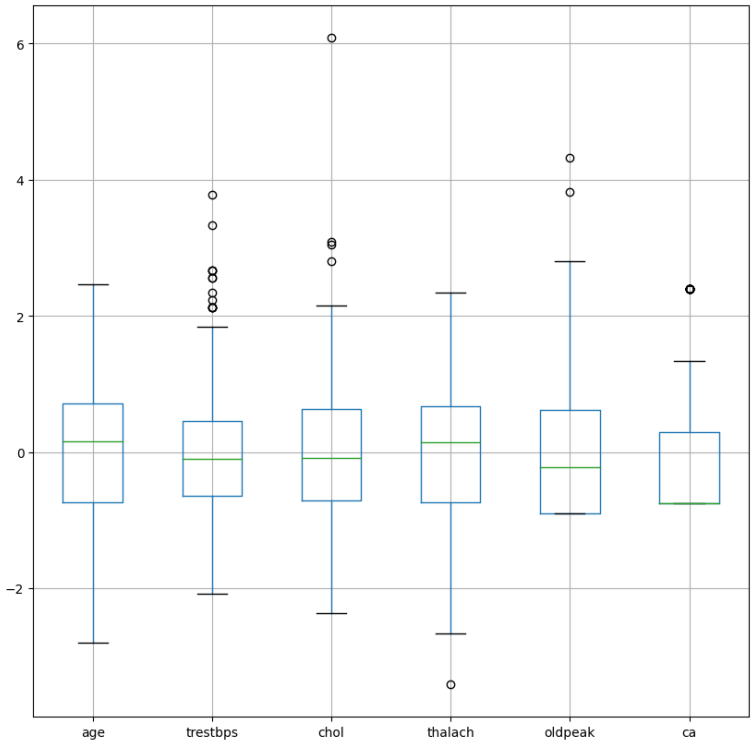
0.0s

Rysunek 9: Kod naprawiający zbilansowanie próbek

Usunięcie tych danych pozwoliło także na pozbycie się wartości pustych dla cechy ca.

Następnie rozwiązałem problem różnych zakresów cech liczbowych poprzez standaryzację cech, w taki sposób zachowane pozostały rozkłady owych cech. Standaryzację wykonałem za pomocą wzoru $z = \frac{x - \mu}{\sigma}$, gdzie:

- x – zmienna niestandaryzowana,
- μ – średnia z populacji,
- σ – odchylenie standardowe populacji.



Rysunek 10: Wykresy pudełkowe cech po standaryzacji

Wynikiem wszystkich tych operacji jest gotowa macierz cech liczbowych z przykładami, którą można wykorzystać do dalszych ćwiczeń.

	age	trestbps	chol	thalach	oldpeak	ca	heart_disease
0	0.932294	0.736946	-0.302354	0.055047	1.040180	-0.752472	0
1	1.371582	1.567138	0.720267	-1.789187	0.367248	2.394228	1
2	1.371582	-0.646707	-0.379533	-0.867070	1.292529	1.345328	1
3	-1.923080	-0.093246	0.025656	1.679729	2.049578	-0.752472	0
4	-1.483792	-0.093246	-0.861902	1.021074	0.283131	-0.752472	0
...	...	...	...	...	...	...	...
271	0.273362	0.460215	-0.147996	-1.130532	-0.726267	-0.752472	1
272	-1.044504	-1.200169	0.295783	-0.735339	0.114898	-0.752472	1
273	1.481405	0.681600	-1.074144	-0.340146	1.965461	1.345328	1
274	0.273362	-0.093246	-2.270418	-1.481814	0.114898	0.296428	1
275	0.273362	-0.093246	-0.244470	1.108895	-0.894500	0.296428	1

276 rows × 7 columns

Rysunek 11: Wynikowa macierz cech liczbowych

1.4 Wnioski

- Analiza danych pozwala nam na lepsze zrozumienie zbioru, a także naprawę problemów w zbiorze, które mogą spowodować gorsze wyniki naszego modelu.
- Warto zwrócić uwagę na zbalansowanie klasyfikacji w próbkach, gdyż brak owego zbalansowania może nauczyć model rozpoznawania poprawnej klasyfikacji tylko w kilku z nich (tych klas, których jest najwięcej).
- Innymi wartymi uwagi problemami jakie mogą pojawić się w zbiorze są brakujące dane, nierówne zakresy cech, bądź zbyt duża korelacja danych.
- Warto zwrócić uwagę na rozkłady cech w zbiorze, ponieważ może nam to pomóc w wyborze odpowiedniego modelu do rozwiązania naszego zadania.

- Analiza danych to proces iteracyjny, który nie raz wymaga wielu kroków, warto wspomóc się bibliotekami np. dla języka python, które pomagają nam na np. szybszą operację na danych oraz różne wizualizacje zbioru danych.

Literatura

Heart Disease Dataset. (1988, czerwiec). Retrieved from `https://archive.ics.uci.edu/dataset/45/heart+disease`

K najbliższych sąsiadów. (2022, czerwiec). Retrieved from `https://pl.wikipedia.org/wiki/K_najbli%C5%BCszych_s%C4%85siad%C3%B3w`