Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Uczenie Maszynowe

Dokumentacja wstępna

Piotr Patek, Jan Potaszyński

Warszawa, 2024

Spis treści

1. Dokumentacja wstępna

1.1. Ogólne założenia

Treść zadania

W ramach projektu z przedmiotu Uczenie Maszynowe przydzielone zostało następujące zadanie:

„Las losowy z naiwnym klasyfikatorem bayesowskim (NBC) w zadaniu klasyfikacji. Postępujemy tak jak przy tworzeniu lasu losowego, tylko co drugi klasyfikator w lesie to NBC. Jeden z klasyfikatorów (NBC lub drzewo ID3) może pochodzić z istniejącej implementacji. Przed rozpoczęciem realizacji projektu proszę zapoznać się z zawartością: `https://staff.elka.pw.edu.pl/~rbiedrzy/UMA/index.html`."

Na podstawie treści zadania autorzy zdecydowali się zaimplementować zarówno klasyfikator w postaci drzewa ID3, jak i naiwny klasyfikator bayesowski. Na podstawie stworzonych modeli lokalnych tworzony będzie model w postaci lasu losowego, gdzie co drugie drzewo decyzyjne zostanie zastąpione naiwnym klasyfikatorem bayesowskim.

Stworzony kod zostanie odpowiednio pokryty testami jednostkowymi, a do realizacji projektu wykorzystany zostanie język Python.

1.2. Implementacja lasu losowego z drzewami ID3 i NBC

1.2.1. Drzewo decyzyjne ID3

Drzewo decyzyjne ID3 to klasyczne drzewo decyzyjne tworzone za pomocą algorytmu o nazwie ID3. Algorytm ten polega na takim rozgałęzianiu drzewa, by sumaryczna entropia $H(S)$ definiowana następującym wzorem:

$$H(S) = -\sum_{x \in X} p(x) \cdot \log_2 p(x) \tag{1.1}$$

gdzie:

— S - Dane, dla których liczona jest entropia,
— X - Zbiór klas w zbiorze S,
— $p(x)$ - Stosunek liczby elementów w zbiorze S z przypisaną klasą $x \in X$ do liczby wszystkich elementów w zbiorze S.

była z każdym następnym rozgałęzieniem możliwie najbardziej zmniejszana. Innaczej, żeby przyrost wiedzy $G(S, S_a, S_b)$ definiowany wzorem:

$$G(S, S_a, S_b) = H(S) - (H(S_a) + H(S_b)) \tag{1.2}$$

gdzie:

— S - Dane, dla których liczona jest entropia,
— S_a, S_b - Dychotomiczny podział danych S.

był jak największy.

Najprostszą metodą podziału danych jest porównanie jej do pewnej wybranej wartości. Na przykład:

$$\begin{gathered} s \in S \\ \lambda_s \geqslant value \implies s \in S_a \\ \lambda_s < value \implies s \in S_b \end{gathered} \tag{1.3}$$

gdzie λ_s jest wartością atrybutu, na podstawie którego dokonujemy podziału, a *value* wartością wybraną przez algorytm, na podstawie której zachodzi podział.

Rozgałęzienia są dokonywane aż nie zajdzie jeden z tzw. warunków stopu. W tym przypadku jeden z następujących trzech:

— Gdy rozpatrywany zbiór S jest 1 elementowy (jest to szczególny przypadek następnego warunku),
— Gdy wszystkie elementy rozpatrywanego zbioru S należą do tej samej klasy $x \in X$,
— Gdy osiągana jest maksymalna głębokość drzewa (ustalana arbitralnie przez trenującego)

W takim wypadku tworzony jest liść, czyli ostateczna klasyfikacja $x \in X$ (w przypadku, gdy zachodzi trzeci warunek liść przyjmuje klasę częściej występującą w zbiorze)

Algorytm ID3 tworzenia drzewa decyzyjnego w pseudokodzie wygląda następująco:

Algorithm 1 Algorytm ID3

Definicja Podziel(k, λ) - dokonaj podziału zbioru S ze względu na wartość λ parametru k
Definicja Entropia(Podziel(k, λ)) - oblicz entropię tak dokonanego podziału
Definicja Warunek STOP:
- Jeśli Wszystkie elementy S są tej samej klasy
- Jeśli maksymalna głębokość drzewa została osiągnięta

Stworzenie pierwszego węzła dla zbioru S
Realizacja algorytmu:
if Warunek STOP **then**
    Zamiana tego węzła na liść, klasyfikujący wg najliczniejszej klasy w zbiorze S
end if
k_{best} = Dowolny atrybut k
for Każdy rodzaj atrybutu k **do**
    λ_{best} = Dowolna wartość λ atrybutu k
    for Wartość λ atrybutu k dla każdego z elementów z S **do**
        if Entropia(Podziel(k,λ)) < Entropia(Podziel(k,λ_{best})) **then**
            $\lambda_{\text{best}} = \lambda$
        end if
    end for
    if Entropia(Podziel(k,λ_{best})) < Entropia(Podziel(k_{best},λ_{best})) **then**
        $k_{\text{best}} = k$
    end if
end for
Dodanie k_{best} i λ_{best} jako parametrów decyzyjnych tego węzła
(S_a, S_b) = Podziel(k_{best}, λ_{best})
Stworzenie węzłów dla zbiorów S_a oraz S_b i dodanie ich jako węzłów potomnych tego węzła.
Realizacja algorytmu dla węzłów potomnych

Przykładowe obliczenia entropii dla podziału 3 elementowego zbioru S

Załóżmy, że posiadamy 3 elementowy zbiór S zawierający próbki o jednym atrybucie ciągłym. Każda próbka ma przydzieloną jedną z dwóch klas 0 lub 1. Naszym celem jest taki podział pierwotnego zbioru S na podzbiory, aby zmniejszać entropię zbioru po podziale i maksymalizować przyrost wiedzy (ang. information gain).

Wartość atrybutu	1	2	3
Klasyfikacja	0	0	1

Tab. 1.1. Wartości atrybutu i klasy przypisane próbkom z przykładowego zbioru S

Entropia dla niepodzielonego zbioru:

$$H(S) = -\frac{1}{3} \cdot \log_2 \frac{1}{3} - \frac{2}{3} \cdot \log_2 \frac{2}{3} \approx 0{,}918 \tag{1.4}$$

Entropia dla podziału wg kryterium (Atrybut $\geqslant 2$):

$$H(S_{21}) + H(S_{22}) = -1 \cdot \log_2 1 - \frac{1}{2} \cdot \log_2 \frac{1}{2} - \frac{1}{2} \cdot \log_2 \frac{1}{2} = 1 \tag{1.5}$$

Entropia dla podziału wg kryterium (Atrybut $\geqslant 3$):

$$H(S_{31}) + H(S_{32}) = -1 \cdot \log_2 1 - 1 \cdot \log_2 1 = 0 \tag{1.6}$$

Przyrost wiedzy dla podziału wg kryterium (Atrybut $\geqslant 2$):

$$G(S, S_{21}, S_{22}) = H(S) - (H(S_{21}) + H(S_{22})) = -0{,}082 \tag{1.7}$$

Przyrost wiedzy dla podziału wg kryterium (Atrybut $\geqslant 3$):

$$G(S, S_{31}, S_{32}) = H(S) - (H(S_{31}) + H(S_{32})) = 1 \tag{1.8}$$

Z obliczeń wynika, że podział według kryterium (Atrybut $\geqslant 3$) jest podziałem preferowanym przez algorytm ID3.

Tak stworzone drzewo użyte może być do klasyfikacji obiektu w następujący sposób. Obiekt zaczyna od pierwszego wierzchołka (zwanego korzeniem). Obliczana jest wartość funkcji entropii dla zbioru, na podstawie której wybierany jest najlepszy atrybut wg którego powinien nastąpić następny podział i jego próg. Następne wierzchołki są tworzone tak długo jak nie zostanie spełnione kryterium stopu. Wtedy wierzchołek zostaje liściem, któremu przypisana jest konkretna klasa w klasyfikacji.

1.2.2. Naiwny klasyfikator bayesowski

Naiwny klasyfikator bayesowski to metoda klasyfikacji oparta na powszechnie znanym w statystyce twierdzeniu Bayesa.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1.9}$$

gdzie A, B to zdarzenia losowe.

Na potrzeby uczenia maszynowego, zmienne losowe A i B zastąpimy poprzez zmienne y i $x_1, \cdots, x_n$ oznaczające odpowiednio klasę przyporządkowaną danej próbce i wektor atrybutów danej próbki.

Po wspomnianych modyfikacjach wzór wygląda następująco:

$$P(y|x_1, \cdots, x_n) = \frac{P(x_1, \cdots, x_n|y)P(y)}{P(x_1, \cdots, x_n)} \tag{1.10}$$

W tej chwili możemy zastosować główne założenie algorytmu naiwnego klasyfikatora bayesowskiego, od którego pochodzi jego nazwa, a mianowicie założenie (naiwne i niepotwierdzone), że predyktory modelu są niezależne względem siebie.

$$P(x_1, \cdots, x_n|y) = \prod_{j=1}^{n} P(x_j|y) \tag{1.11}$$

W praktyce prawdopodobieństwo warunkowe, że atrybut x_i przyjmie daną wartość pod warunkiem, że próbka należy do danej klasy y estymuje się na podstawie zbioru trenującego $\mathbb{T}$ w następujący sposób:

$$P(x_i|y) = \frac{|\mathbb{T}_{y,x_i}|}{|\mathbb{T}_y|} \tag{1.12}$$

Prawdopodobieństwa *a priori* możemy obliczyć w bardzo prosty sposób na podstawie naszego zbioru trenującego, zliczając liczbę wystąpień danej klasy w tym zbiorze, a następnie licząc jej stosunek do wielkości całego zbioru trenującego.

$$P(y) = \frac{|\mathbb{T}_y|}{|\mathbb{T}|} \tag{1.13}$$

Ostatecznie wzór na predykcję klasy dla próbki $\mathbf{x}$ o określonych atrybutach wygląda następująco:

$$\hat{y} = \underset{y}{\operatorname{argmax}}\ P(y) \prod_{i=1}^{n} P(x_i|y) \tag{1.14}$$

gdzie:

— n - liczba atrybutów dla przykładów,
— x_i - i-ty atrybut przykładu,
— y - dowolna klasa ze zbioru danych,
— $P(y)$ - prawdopodobieństwo *a priori* dla danej klasy,
— $P(x_i|y)$ - prawdopodobieństwo warunkowe wystąpienia danej klasy dla danej wartości atrybutu,
— $\hat{y}$ - predykcja modelu.

Dodatkowo należy jeszcze wspomnieć, że naiwny klasyfikator bayesowski w podstawowej formie działa na atrybutach dyskretnych. W tym celu, przed etapem trenowania modelu należy dokonać dyskretyzacji wszystkich ciągłych argumentów.

1.2.3. Zmodyfikowany las losowy

Zmodyfikowany las losowy polega na stworzeniu wielu różnych klasyfikatorów (w tym przypadku drzew losowych oraz naiwnych klasyfikatorów bayesowskich, w stosunku 50/50)(ilość klasyfikatorów może być arbitralnie wybrana przez trenującego). Klasyfikatory tworzone są za pomocą tzw. próby bootstrap z liczebnością próby równą liczebności populacji. Innymi słowy z n-elementowego zbioru danych wybieramy n-elementową próbę za pomocą losowania ze zwracaniem z naszego zbioru danych. Dzięki temu klasyfikatory będą się od siebie różniły. Predykcja odbywa się przez głosowanie, czyli zliczana jest ilość poszczególnych klasyfikacji dokonanych przez każdy z klasyfikatorów w lesie i wybierana jest klasyfikacja częściej występująca.

1.3. Eksperymenty

1.3.1. Opis eksperymentów

Następnym krokiem, równie ważnym jeżeli nie ważniejszym, są eksperymenty z wykorzystaniem uprzednio zaimplementowanych algorytmów. Na samym początku warto zwrócić uwagę, że główny algorytm, który ma być poddany eksperymentom - zmodyfikowany las losowy - składa się z już funkcjonujących algorytmów klasyfikujących. Dlatego też zdecydowano się, że aby mieć możliwość stwierdzenia, czy wprowadzona zmiana - zamiana połowy drzew w lesie losowym na naiwne klasyfikatory bayesowskie - prowadzi do poprawy lub pogorszenia możliwości klasyfikacyjnych algorytmu zdecydowano się testować wszystkie 4 algorytmy na wszystkich zbiorach danych:

— Pojedyncze drzewo ID3,
— Pojedynczy naiwny klasyfikator bayesowski,
— Klasyczny las losowy (100% ID3),
— Zmodyfikowany las losowy (50% ID3 + 50% NBC).

Eksperymenty będą wykonywane na zbiorach treningowych i testujących różnej wielkości, które będą wybierane losowo.

1.3.2. Wykorzystywane miary jakości

Zanim przejdziemy do miar jakości, które będą wykorzystywane w trakcie wykonywania eksperymentów, należy podkreślić, że wszystkie wartości miar jakości będą wartością uśrednioną z min. 25 uruchomień. Dodatkowo podawane będzie - wartość minimalna, wartość maksymalna i odchylenie standardowe będące wynikiem wielu uruchomień algorytmu.

Dla każdego zbioru danych określona została macierz pomyłek (ang. confusion matrix). Dzięki temu jasno jest określone jak powinny być oceniane próbki o określonych klasach rzeczywistych po etapie predykcji.

Błąd:

$$E = \frac{\mathrm{FP} + \mathrm{FN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{1.15}$$

Dokładność:

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{1.16}$$

Współczynnik prawdziwych pozytywów:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{1.17}$$

Współczynnik fałszywych pozytywów:

$$\mathrm{TPR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}} \tag{1.18}$$

Precyzja:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{1.19}$$

Ponadto dla każdego eksperymentu przeprowadzona zostanie analiza ROC.

1.3.3. Zbiór Wine Quality

Pierwszy zbiór danych jaki zostanie wykorzystany w ramach eksperymentów zawiera próbki opisujące różne wina.

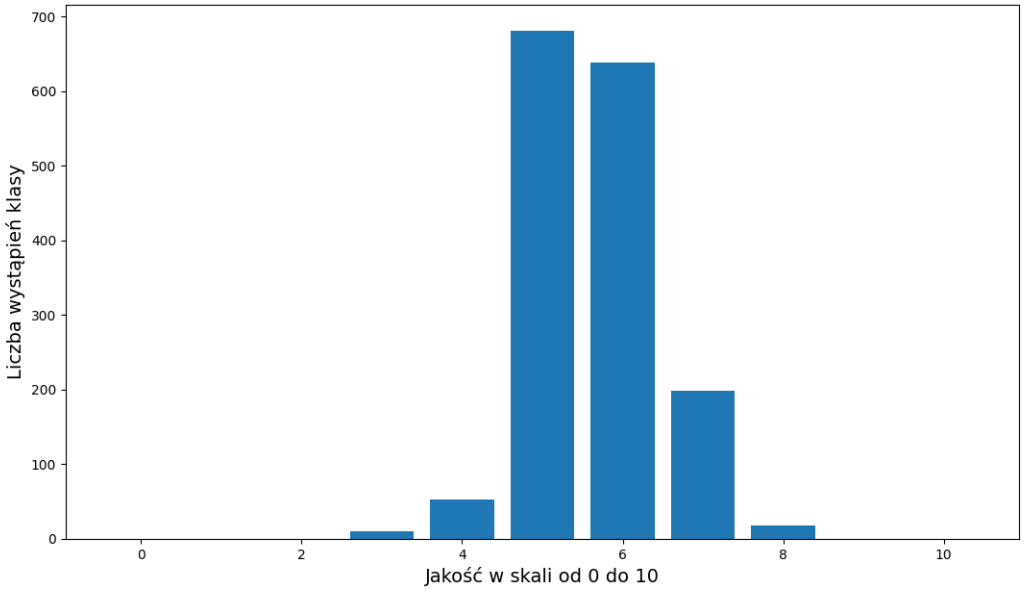
Liczba próbek	Liczba klas	Liczba atrybutów
6497	2	12

Tab. 1.2. Podstawowe informacje o zbiorze

Każda próbka posiada następujące atrybuty:

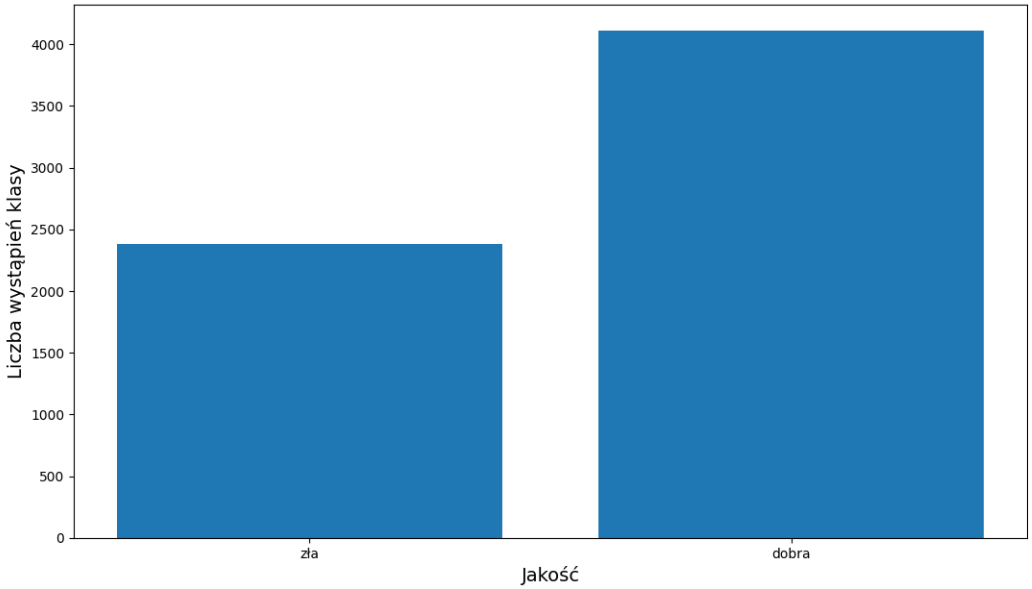
Polska nazwa atrybutu	Angielska nazwa atrybutu	Typ atrybutu
kwasowość stała	fixed acidity	ciągły
kwasowość ulotna	volatile acidity	ciągły
zawartość kwasu cytrynowego	citric acid	ciągły
zawartość cukru resztkowego	residual sugar	ciągły
zawartość chlorków	chlorides	ciągły
wolnego dwutlenku siarki SO_2	free sulfur dioxide	ciągły
ilość całkowitego dwutlenku siarki SO_2	total sulfur dioxide	ciągły
gęstość	density	ciągły
pH	pH	ciągły
zawartość siarczanów	sulphates	ciągły
zawartość alkoholu	alcohol	ciągły
kolor	color	dyskretny

Tab. 1.3. Atrybuty próbek ze zbioru Wine Quality



Rys. 1.1. Oryginalne klasy w zbiorze wine quality

W oryginalnej wersji zbiór jakości win posiada 10 klas jakości, jednakże jak można zauważyć Celem naszej klasyfikacji jest przydzielenie każdej próbce na podstawie wartości jej atrybutów klasy postaci jakości o wartościach dyskretnych od 0 do 10.



Rys. 1.2. Zmodyfikowane klasy w zbiorze wine quality

		Rzeczywista klasyfikacja	
		Dobra	Zła
Predykowana klasyfikacja	Dobra	Prawdziwie dobra (TP)	Fałszywie dobra (FP)
	Zła	Fałszywie zła (FN)	Prawdziwie zła (TN)

Tab. 1.4. Macierz pomyłek

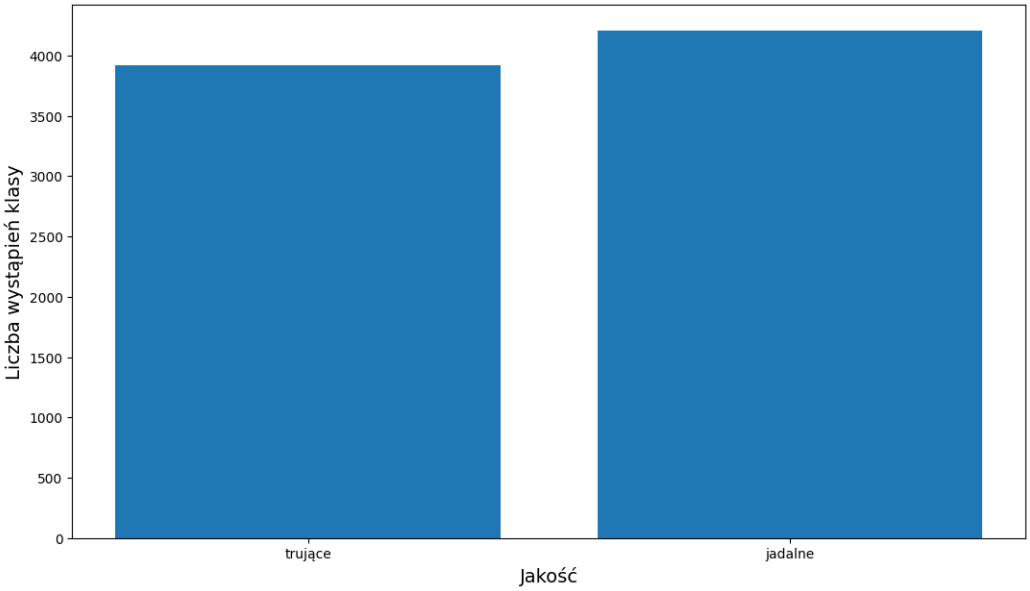
`https://archive.ics.uci.edu/dataset/186/wine+quality`

1.3.4. Zbiór Mushroom

Kolejnym rozważanym zbiorem danych będzie zbiór mushroom zawierający próbki sklasyfikowane w sposób binarny - na podstawie atrybutów należy stwierdzić czy grzyb jest jadalny czy nie.

Liczba próbek	**Liczba klas**	**Liczba atrybutów**
8124	2	22

Tab. 1.5. Podstawowe informacje o zbiorze



Rys. 1.3. Oryginalne klasy w zbiorze mushroom

		Rzeczywista klasyfikacja	
		Jadalny	Trujący
Predykowana klasyfikacja	Jadalny	Prawdziwie jadalny (TP)	Fałszywie jadalny (FP)
	Trujący	Fałszywie trujący (FN)	Prawdziwie trujący (TN)

Tab. 1.6. Macierz pomyłek

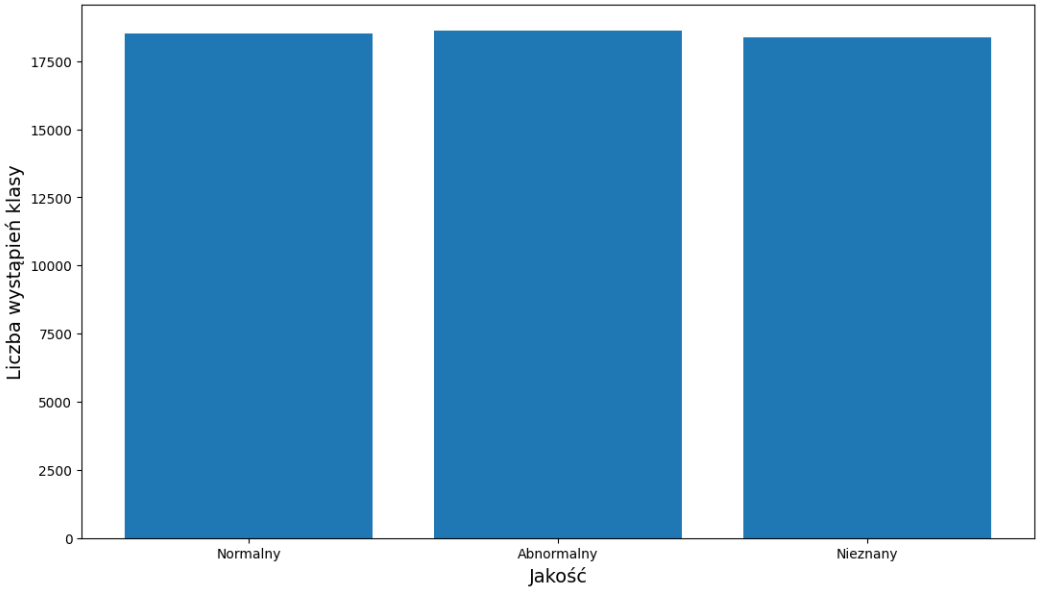
`https://archive.ics.uci.edu/dataset/73/mushroom`

1.3.5. Zbiór Healthcare

Ostatnim zbiorem danych na którym zostaną przeprowadzone eksperymenty był zbiór healthcare. Zawiera on próbki o atrybutach opisujących parametry medyczne pacjenta na podstawie których należy stwierdzić czy stan pacjenta jest normalny (pacjent jest zdrowy), czy stan pacjenta jest abnormalny (pacjent ma problemy zdrowotne) lub czy nie da się w sposób jednoznaczny stwierdzić jaki jest stan zdrowia pacjenta (Nieznany).

Liczba próbek	**Liczba klas**	**Liczba atrybutów**
49992	3	14

Tab. 1.7. Podstawowe informacje o zbiorze



Rys. 1.4. Oryginalne klasy w zbiorze healthcare

		Rzeczywista klasyfikacja	
		Abnormalny	Inny
Predykowana klasyfikacja	Abnormalny	Prawdziwie Abnormalny (TP)	Fałszywie abnormalny (FP)
	Inny	Fałszywie inny (FN)	Prawdziwie inny (TN)

Tab. 1.8. Macierz pomyłek dla klasy abnormalny

		Rzeczywista klasyfikacja	
		Normalny	Inny
Predykowana klasyfikacja	Normalny	Prawdziwie normalny (TP)	Fałszywie normalny (FP)
	Inny	Fałszywie inny (FN)	Prawdziwie inny (TN)

Tab. 1.9. Macierz pomyłek dla klasy normalny

		Rzeczywista klasyfikacja	
		Niejednoznaczny	Inny
Predykowana klasyfikacja	Niejednoznaczny	Prawdziwie nieznany (TP)	Fałszywie nieznany (FP)
	Inny	Fałszywie inny (FN)	Prawdziwie inny (TN)

Tab. 1.10. Macierz pomyłek dla klasy niejednoznaczny (ang. inconslusive)

`https://www.kaggle.com/datasets/prasad22/healthcare-dataset`

1.3.6. Inne potencjalne zbiory

https://www.kaggle.com/datasets/spscientist/students-performance-in-exams
https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset