Uczenie Maszynowe

Projekt
Dokumentacja końcowa

Marcin Latawiec (311031),
Jan Sosnowski (311094),

Politechnika Warszawska

Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska
Warszawa
Sunday 22nd January, 2023

Spis treści

1 Temat Projektu

Studium projektu obejmuje zadanie klasyfikacji danych z brakującymi wartościami przy wykorzystaniu algorytmu uczenia maszynowego do tworzenia drzewa decyzyjnego.

2 Wstęp

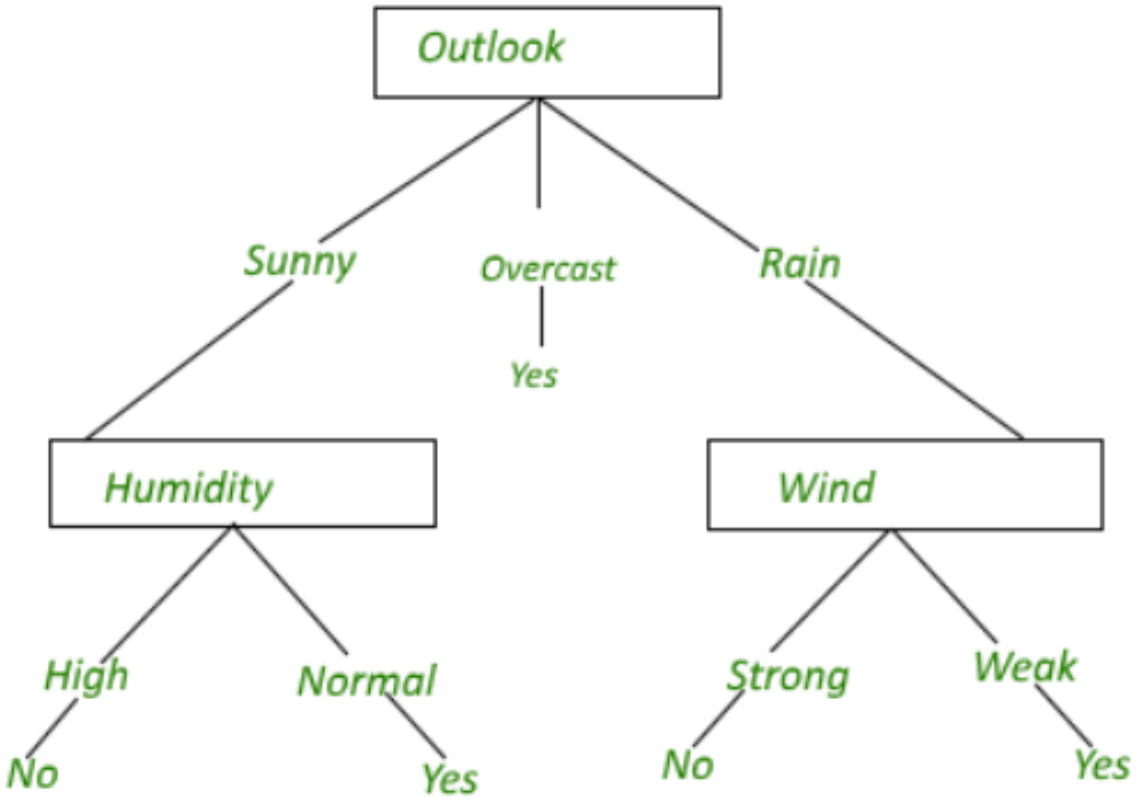
Klasyfikacja danych jest powszechnie stosowaną metodą, aby na podstawie określonych danych przypisać jakiś przykład/element do konkretnej podgrupy(klasy). W zadaniach klasyfikacji często stosuje się drzewa decyzyjne, których istotną zaletą jest czytelność dla człowieka, a także praktycznie nieograniczona rozszerzalność - korzeń drzewa znajduje się na poziomie 0 - a jego węzły w ogólnym przypadku znajdują się na poziomie l+1, gdzie $l \in \mathsf{N}$.

Problem Brakujących Wartości

Dla rzeczywistych zbiorów danych - tych pochodzących z obserwacji, ankiet oraz różnych pomiarów - niekompletność jest powszechną właściwością. Często zdarza się, że w przykładach brakuje niektórych wartości. Taka sytuacja może mieć miejsce zarówno dla przykładów ze zbioru trenującego, jak i dla nowych przykładów, które mają być klasyfikowane za pomocą uzyskanego klasyfikatora. Istnieje wiele różnych metod uodparniających drzewa decyzyjne na problem brakujących wartości, w taki sposób że możliwym staje się przetwarzanie zbioru danych przez drzewo - co więcej jakość takiej klasyfikacji może być oceniana wysoko. W kolejnych sekcjach zostanie zawarty opis budowy drzewa decyzyjnego na potrzeby naszych eksperymentów, opis metod służących do uodpornienia drzewa na problem brakujących wartości, zostaną przedstawione przeprowadzone eksperymenty na wybranych zbiorach danych, na podstawie których sformułowane zostaną ostateczne wnioski.

3 Drzewo decyzyjne

Poprzez drzewo decyzyjne rozumieć należy strukturę, na którą składają się węzły oraz liście. Węzły przechowują testy, które sprawdzają wartości atrybutów przykładów. Liście przypisują przykładom konkretne kategorie. Dla każdego z możliwych wyników testu, z węzła prowadzi odpowiadająca mu gałąź do dalszego poddrzewa. Zwizualizowane działanie drzewa decyzyjnego zamieszczone poniżej ukazuje istotę przetwarzania przykładów - w zależności od przkładu: pogody, badana jest możliwość gry w tenisa.



Zrzut ekranu 1: Drzewo decyzyjne - klasyfikator - gra w tenisa

3.1 Budowa drzewa decyzyjnego

Tworzenie drzewa decyzyjnego jest zasadniczym etapem projektu, koniecznym do przeprowadzenia eksperymentów badających metody obsługi brakujących wartości w zadaniach klasyfikacji. Przed rozpoczęciem tego procesu, należy podjąć kilka ważnych decyzji: jak zostanie określone kryterium zatrzymania, jakie etykiety zostaną przypisane liściom, gdy kryterium zatrzymania zostanie spełnione i jakie testy będą wybierane w każdym węźle. Najprostszymi kryteriami zatrzymania są: brak różnych etykietowaniach, brak przykładów lub brak dostępnych testów. W każdym z tych przypadków, liściowi przydzielana jest odpowiednia etykieta. W pierwszym przypadku, braku przykładów, liściowi przydzielana jest kategoria, która występuje najczęściej na bezpośrednim poziomie rekursji. W drugim przypadku, liściowi przydzielana jest etykieta, którą mają wszystkie przykłady, które do niego dotarły. W trzecim przypadku, liściowi przydzielana jest wartość, którą reprezentuje większość przykładów, które do niego dotarły. Przy budowie algorytmu budowy drzewa decyzyjnego konieczne jest wybranie testu, na podstawie którego dane będą klasyfikowane do kolejnych węzłów. Decyzją projektową zastosowane zostały testy nierównościowe - okazują się być najefektywniejsze przy pracy z zadanymi(liczbowymi) zbiorami danych. Kryterium wyboru testu to przyrost informacji: obliczany według wzoru:

$$g_t(P) = I(P) - E_t(P) \tag{1}$$

$$I(P) - \text{informacja zawarta w zbiorze etykietowanych przykładów P} \tag{2}$$

$$I(P) = \sum_{d \in \mathbb{C}} -\frac{|P^d|}{|P|} log \frac{|P^d|}{|P|} \tag{3}$$

$$E_t(P) - \text{średnia ważona entopia zbiorów przykładów P ze względu na test t} \tag{4}$$

$$E_t(P) = \sum_{r \in \mathbb{R}} -\frac{|P_{tr}|}{|P|} E_{tr}(P) \tag{5}$$

$$\text{dla } E_{tr}(P) = \sum_{d \in \mathbb{C}} -\frac{|P_{tr}^d|}{|P_{tr}|} log \frac{|P_{tr}^d|}{|P_{tr}|} \tag{6}$$

Zaimplementowany algorytm budowy drzewa jest inspirowany algorytmem ID3. Drzewo ma postać rekurencyjnego słownika. W z każdego węzła odchodzą maksymalnie dwa poddrzewa dla określonej wartości oraz (w zależności do wybranego sposobu budowania) dodatkowo jedna dla brakującej wartości. Wybór atrybutu i progu podziału jest zaimplementowany tak, że przechodzimy po wszystkich atrybutach, po wszystkich wartościach i wybieramy tą parę atrybut wartość, dla której entropia podziału jest największa. Funkcja **def_create_tree:** zwraca drzewo decyzyjne reprezentujące hipotezę przybliżającą c na zbiorze P, zgodnie z poniższym pseudokodem:

```
if kryteriumstopu (P, S) then
    utworz lisc 1 ;
    d1 := kategoria (P, d) ;
    return 1 ;
end if
utworz wezel n ;
tn := wybierztest (P, S) ;
d := kategoria (P, d) ;
for r in Rtn
    n [ r ] := createtree(Ptnr , d , S      { tn }) ;
end for
return n ;
```

3.1.1 Testy nierównościowe

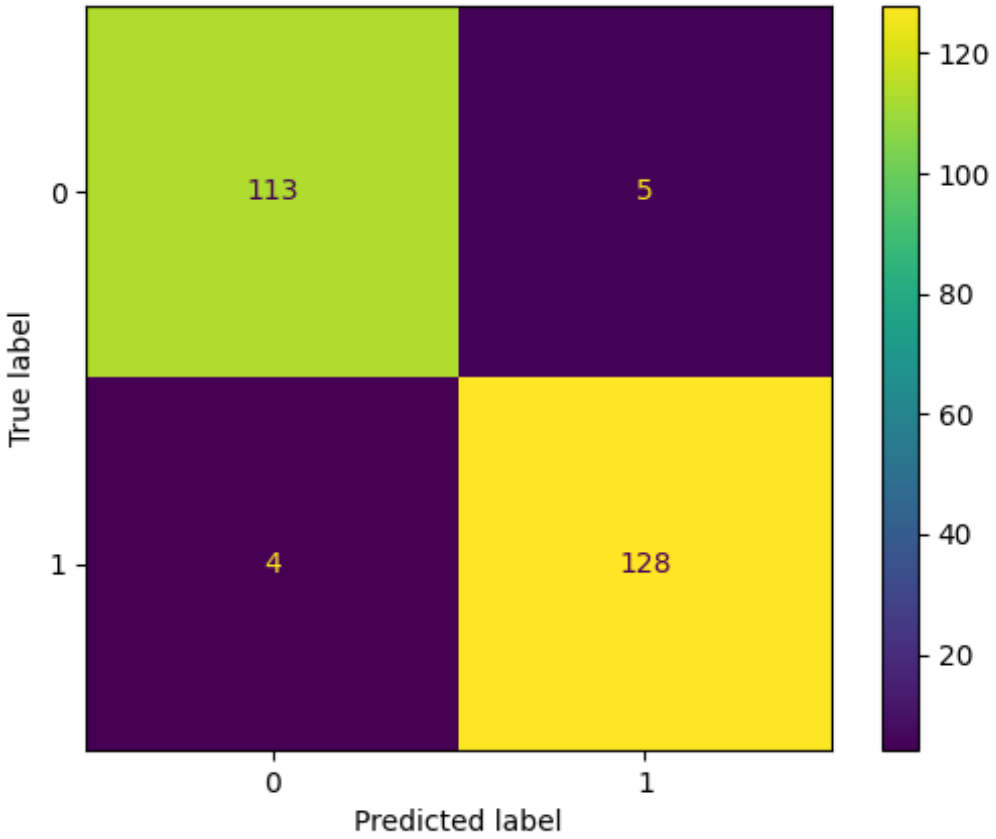
Ustalenie zbioru możliwych testów nierównościowych dla każdego atrybutu wymaga wybrania wartości progowych, które mają być używane do porównań w nierównościach. Dla każdego takiego atrybutu może być tyle różnych testów nierównościowych, ile możliwych wartości progowych wybranych ze zbioru jego wartości. Przyjęty został model gdzie w zależności od parametru *information_gain* ustawiany jest próg testów.

3.1.2 Algorytm referencyjny budowy drzewa decyzyjnego

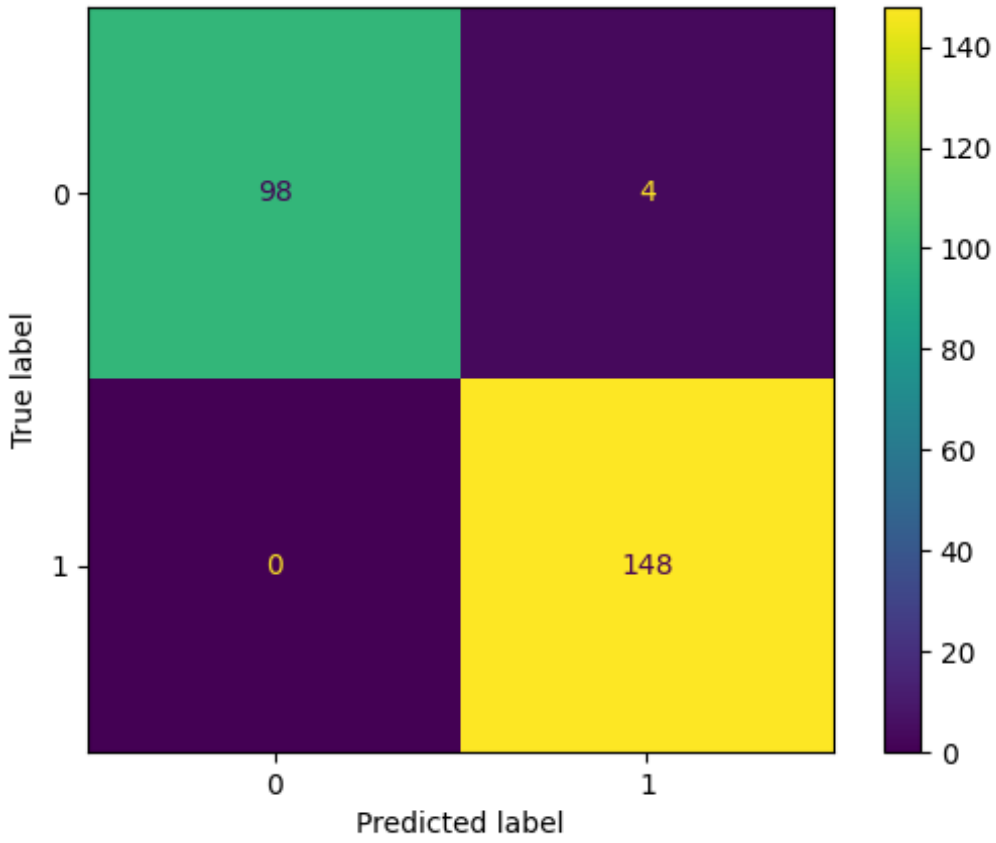
Jako algorytm referencyjny przyjęto Klasyfikator drzewa decyzyjnego wbudowany w moduł **python sklearn**(klasyfikator nie jest uodporniony na brakujące dane). Zbiór testujący(obejmujący 25% zbioru danych) wybierany jest losowo, z tego powodu wykonano kilka prób testowych, aby utrzymać najbardziej miarodajne wyniki. Wyniki zostały zebrane w tabeli 1. Porównując wyniki 2 algorytmów będziemy w stanie zweryfikować poprawność własnej implementacji. Testy służące do weryfikacji klasyfikatora drzewa decyzyjnego znajdują się w pliku *validation.py*.

Tabela 1: Porównanie algorytmu budowy drzewa decyzyjnego z metodą referencyjną

Nazwa zbioru - numer próby	**Dokładność - własny klasyfikatora**	**Dokładność ref. klasyfikatora**
Ryż 1	0.98	0.98
Ryż 2	0.98	0.98
Ryż 3	0.98	0.98
Ryzyko kredytowe 1	0.75	0.75
Ryzyko kredytowe 2	0.75	0.75
Ryzyko kredytowe 3	0.75	0.75
Rak piersi 1	0.95	0.92
Rak piersi 2	0.91	0.93
Rak piersi 3	0.93	0.91



Zrzut ekranu 2: Macierz pomyłek dla klasyfikacji ryżu - własny klasyfikator(drzewo decyzyjne)



Zrzut ekranu 3: Macierz pomyłek dla klasyfikacji ryżu - referencyjny klasyfikator - biblioteka sklearn

4 Zbiory danych

Do eksperymentów wykorzystane zostaną zbiory danych ze stron archive.ics.uci.edu oraz kaggle.com
Wybrane zostały następujące zbiory:

- Ryż - (18185 przykładów, 10 atrybutów, 2 klasy: '0' - 8200 przykładów, '1' - 9985 przykładów
- Rak piersi - (569 przykładów, 30 atrybutów, 2 klasy: 'M' - 212 przykładów, 'B' - 357 przykładów
- Ryzyko kredytowe - (1125 przykładów, 11 atrybutów, 2 klasy: '0' - 900 przykładów, '1' - 225 przykładów

Każdy ze zbiorów danych posiada wyłącznie atrybuty liczbowe, domyślnie każdy przykład zawiera wszystkie wartości - na początku eksperymentów będziemy stosować skrypt *data_deleter*, który będzie miał za zadanie w sposób pseudolosowy usunąć określone wartości z kolumn badanego zbioru.

4.1 Zbiory trenujące

Założeniem implementacyjnym jest wyodrębnienie spośród wyżej ukazanych zbiorów danych zbiorów trenujących, którymi będzie uczony algorytm budowy drzewa. Do trenowania zostało wydzielone 75 % całego zbioru, na zbiór testowy składać się będzie pozostała część. Przykłady z każdej grupy zostaną dobrane tak, aby każda wartość był adekwatnie przedstawiony, unikając nadmiernej polaryzacji danych. Wybrane zbiory różnią się miedzy sobą ilością przykładów i atrybutów - zbiór z danymi o raku jest dość mały i posłuży do weryfikacji poprawności działania implementowanych algorytmów. Natomiast pozostałe zbiory zawierają więcej przykładów i atrybutów, co jest idealne do głębszego zbadania właściwości implementowanych algorytmów.

5 Brakujące wartości atrybutów

5.1 Wdrożone mechanizmy radzenia sobie z brakującymi wartościami

5.1.1 Usuwanie przykładu

Najprostszym sposobem radzenia sobie z brakującymi danymi w danych przykładach jest usuwanie tych przykładów. Taki przetworzony zbiór danych trenujących pozwala na zbudowanie drzewa decyzyjnego algorytmem, w którym nie występują żadne inne formy uodpornień na brakujące dane. Wadą takiego rozwiązania jest to że badając zbiór, w którym wiele próbek jest niekompletnych nakładamy znaczne ograniczenie na budowane drzewo.

5.1.2 Wypełnianie brakujących wartości

Drzewo decyzyjne zostało uodpornione wykorzystując mechanizm, na który składa się zastępowanie brakującej wartości alternatywnie: średnią/medianą/modą/wartością stałą - z danego atrybutu. Wypełnienia dokonywano na zbiorze trenującym, za pomocą którego budowane zostało drzewo.

5.1.3 Klasyfikacja probabilistyczna

W sytuacji gdy nie ma możliwości ustalenia wyniku testu t w danym węźle z powodu braku wartości atrybutu, można skorzystać z podejścia probabilistycznego. Za pomocą zbioru trenującego, w którym elementy mają znany wynik testu t, można określić prawdopodobieństwo dotarcia do konkretnych liści przez element o nieznanym wyniku testu t (co można zrobić podczas tworzenia drzewa). Prawdopodobieństwo to można oszacować zgodnie z podanym wzorem:

$$Pr(c(x_*) = d) = \sum_{l \in T} Pr(l|x_*) * Pr_{x \in \Omega}(c(x) = d|l) \tag{7}$$

$Pr\_x \in \Omega(c(x) = d|l)$ - prawdopodobieństwo tego, że kategoria przykładu x wybranego z dziedziny zgodnie z rozkładem Ω i zaliczonego do liścia l jest d. Prawdopodobieństwo to jest zgodne ze wzorem:

$$Pr(c(x) = d|l) = \frac{P^d_{T,l}}{P_{T,l}} \tag{8}$$

5.1.4 Podziały zastępcze i przykłady ułamkowe

Metoda podziału zastępczego pozwala na zastąpienie każdego przykładu z brakującą wartością atrybutu a, przykładami ułamkowymi dla różnych wartości tego atrybutu występujących w zbiorze P, w proporcji zgodnej z częstością ich występowania.

Omawiana metoda wymaga z związania z każdym przykładem trenującym liczby jego egzemplarzy. Przykłady z nieznaną wartością atrybutu zostaje zastąpiony zbiorem przykładów ułamkowych ze wszystkimi wartościami tego atrybutu występującymi w zbiorze P.

6 Eksperymenty

6.1 Procedura przeprowadzania eksperymentów

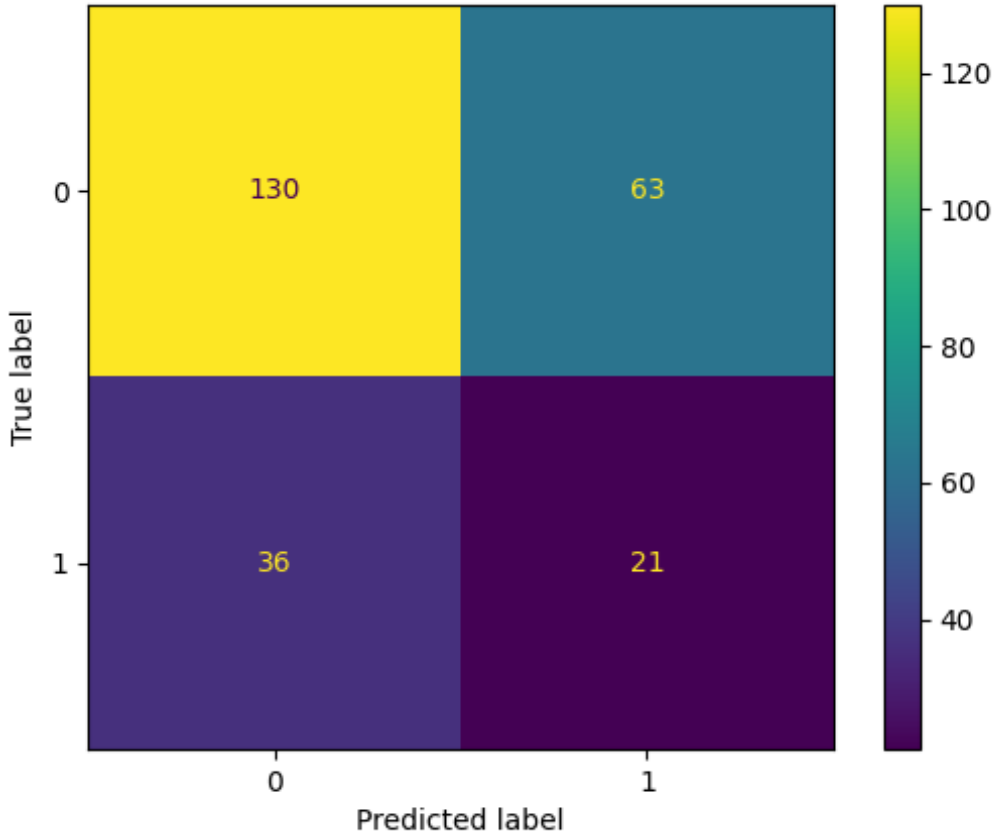
Eksperymenty, dalej określane jako testy, były przeprowadzane w określonych warunkach, tak aby dały jak najbardziej miarodajne wyniki. Została poświęcona wyraźna uwaga, aby mierzona jakość klasyfikacji naszego drzewa decyzyjnego była kilkukrotnie powtórzona, celem takiego procesu było jak największe wykluczenie odchyleń każdorazowej klasyfikacji. Testowane były wszystkie 3 zbiory, dzięki czemu można było zweryfikować wszechstronność i uniwersalność zaimplementowanego drzewa decyzyjnego jak i metod uodparniających algorytm. Oprócz standardowego mierzenia dokładności klasyfikacji, generowane były również macierze pomyłek, które w graficzny sposób prezentowały skuteczność programu.

6.1.1 Metoda 5.1.1 - Usuwanie przykładu

Jest to najbardziej naiwna z metod i należy postawić hipotezę, że jest również najmniej efektywna: usuwając cały przykład, w którym w jednym atrybucie brakuje wartości, pozbawiamy się innych, ważnych danych. Poniższa tabela ukaże skuteczność takiej metody.

Tabela 2: Badany zbiór danych - Ryż: Zmierzona referencyjna jakość klasyfikatora - w przypadku bez brakujących danych = 0.98

Procent brakujących danych	**Liczba brakujących atrybutów**	**Średnia dokładność**
20%	2	0.62
20%	4	0.48
40%	2	0.55
40%	4	0.54
60%	2	0.66
60%	4	0.56
80%	2	0.62
80%	4	0.39



Zrzut ekranu 4: Macierz pomyłek dla klasyfikacji z wykorzystaniem metody usuwania przykładu - zbiór danych - ryż

Zgodnie z oczekiwaniami dokładność klasyfikatora stoi na mało zadowalającym poziomie - trudno uznać ten fakt za dziwny. Wobec braku wysokiej dokładności klasyfikacji, testy zostały przeprowadzone tylko dla jednego

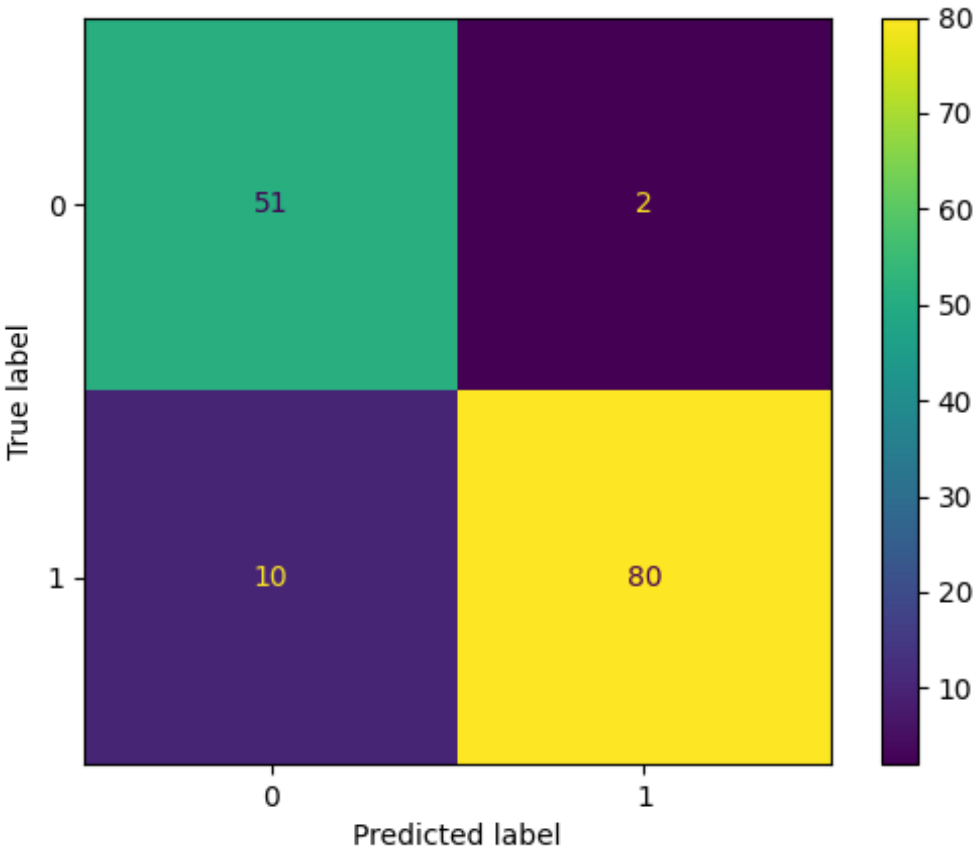
zbioru danych.

6.1.2 Metoda 5.1.2 - Wypełnianie brakujących wartości

W celu sprawdzenia jak algorytm radzi sobie z brakującymi wartościami na zbiorze trenującym, zastosowano różne metody ich uzupełnienia, takie jak średnia, mediana i najczęściej powtarzająca się wartość. W celu uodpornienia drzewa decyzyjnego na takie zdarzenia, losowo usunięto wartości atrybutów, aż do osiągnięcia założonego procentu "zdeformowanych" przykładów. Aby uzyskać obiektywne wyniki, kilkanaście prób zostało wykonanych dla jednej konfiguracji, z różnym procentem elementów z brakującymi wartościami na całym zbiorze trenującym oraz różną ilością brakujących atrybutów w wylosowanych przykładach. Skrypt użyty do przeprowadzenia tych testów znajduje się w pliku o nazwie *test_filler*.

Tabela 3: Badany zbiór danych - **Rak piersi**: Zmierzona referencyjna jakość klasyfikatora (w przypadku bez brakujących danych) = 0.94

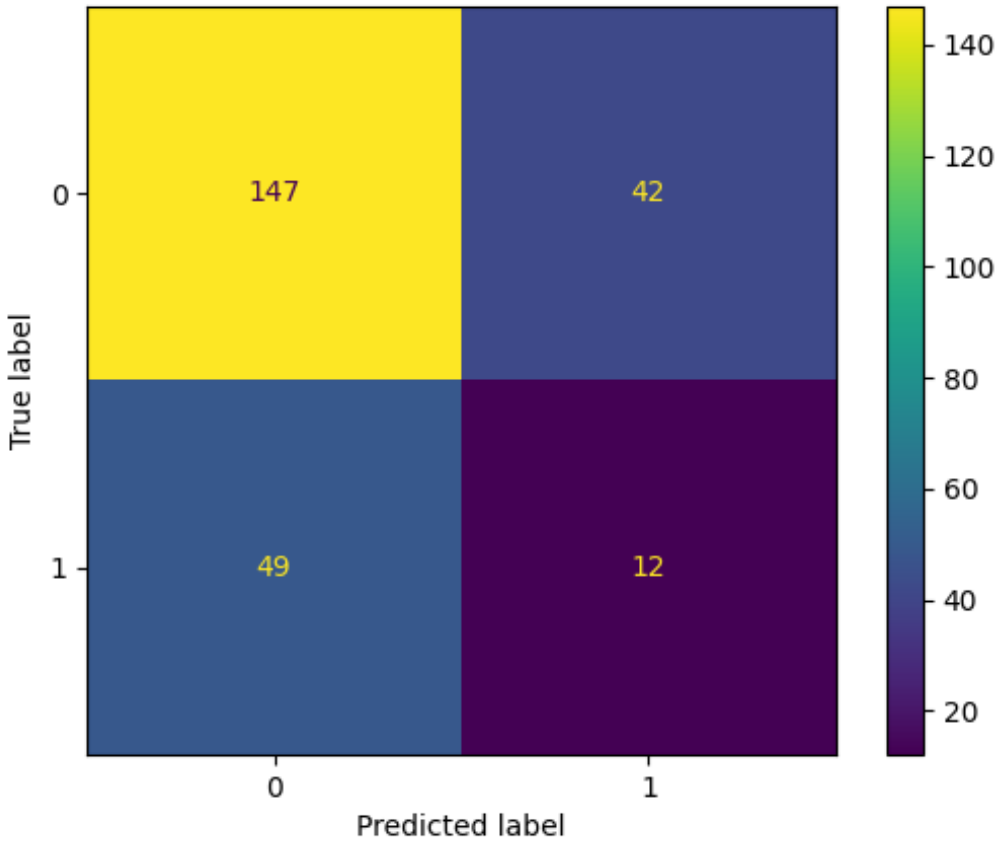
Procent brakujących danych	Liczba brakujących atrybutów	Średnia	**Mediana**	**Moda**
20%	1	0.91	0.92	0.94
20%	2	0.90	0.92	0.94
40%	1	0.88	0.92	0.94
40%	2	0.90	0.91	0.94
60%	1	0.91	0.92	0.94
60%	2	0.90	0.92	0.94
80%	1	0.90	0.92	0.94
80%	2	0.90	0.92	0.94



Zrzut ekranu 5: Macierz pomyłek dla klasyfikacji z wykorzystaniem metody uzupełniania - zbiór danych - rak piersi

Tabela 4: Badany zbiór danych - **Ryzyko kredytowe**: Zmierzona referencyjna jakość klasyfikatora - w przypadku bez brakujących danych = 0.70

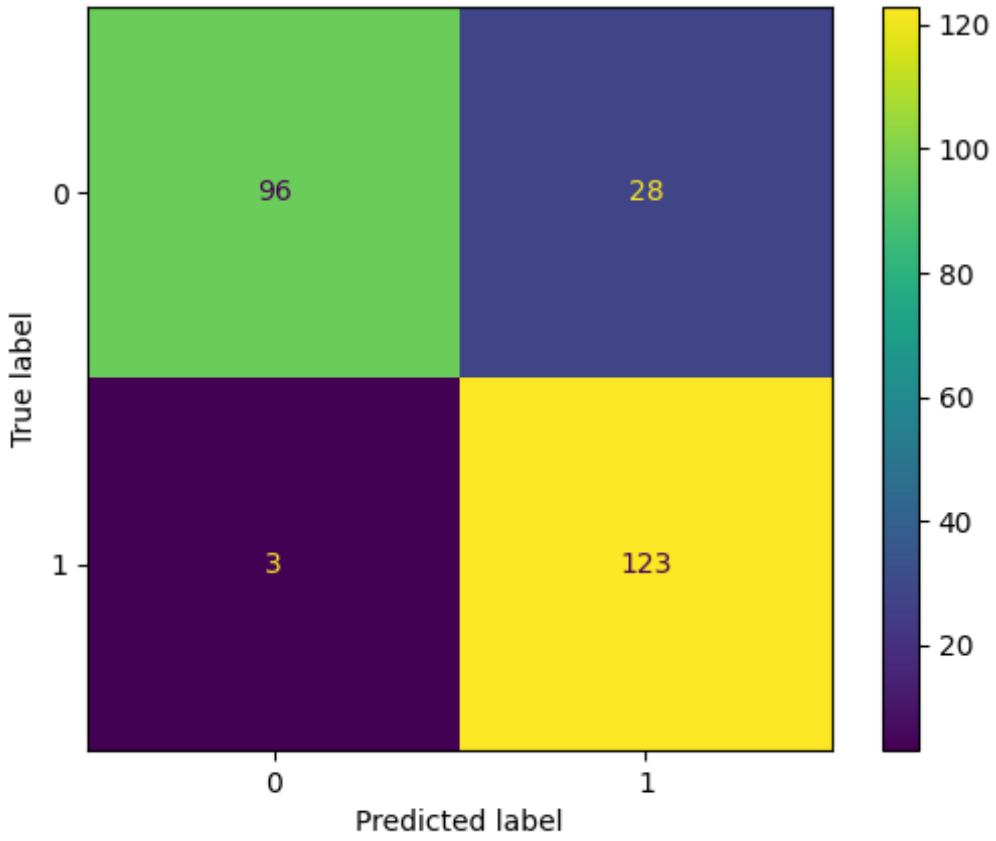
Procent brakujących danych	**Liczba brakujących atrybutów**	**Średnia**	**Mediana**	**Moda**
20%	2	0.70	0.72	0.7
20%	4	0.68	0.71	0.58
20%	6	0.69	0.74	0.66
40%	2	0.69	0.71	0.68
40%	4	0.69	0.65	0.7
40%	6	0.66	0.7	0.69
60%	2	0.69	0.72	0.72
60%	4	0.72	0.66	0.69
60%	6	0.7	0.732	0.7
80%	2	0.70	0.74	0.64
80%	4	0.67	0.71	0.72
80%	6	0.7	0.67	0.72



Zrzut ekranu 6: Macierz pomyłek dla klasyfikacji z wykorzystaniem metody uzupełniania - zbiór danych - ryzyko kredytowe

Tabela 5: Badany zbiór danych - **Ryż**: Zmierzona referencyjna jakość klasyfikatora - w przypadku bez brakujących danych = 0.98

Procent brakujących danych	**Liczba brakujących atrybutów**	**Średnia**	**Mediana**	**Moda**
20%	2	0.98	0.98	0.98
20%	4	0.98	0.98	0.98
40%	2	0.98	0.98	0.98
40%	4	0.98	0.98	0.98
60%	2	0.98	0.98	0.98
60%	4	0.96	0.98	0.97
80%	2	0.98	0.98	0.98
80%	4	0.94	0.98	0.97



Zrzut ekranu 7: Macierz pomyłek dla klasyfikacji z wykorzystaniem metody uzupełniania - zbiór danych - ryż

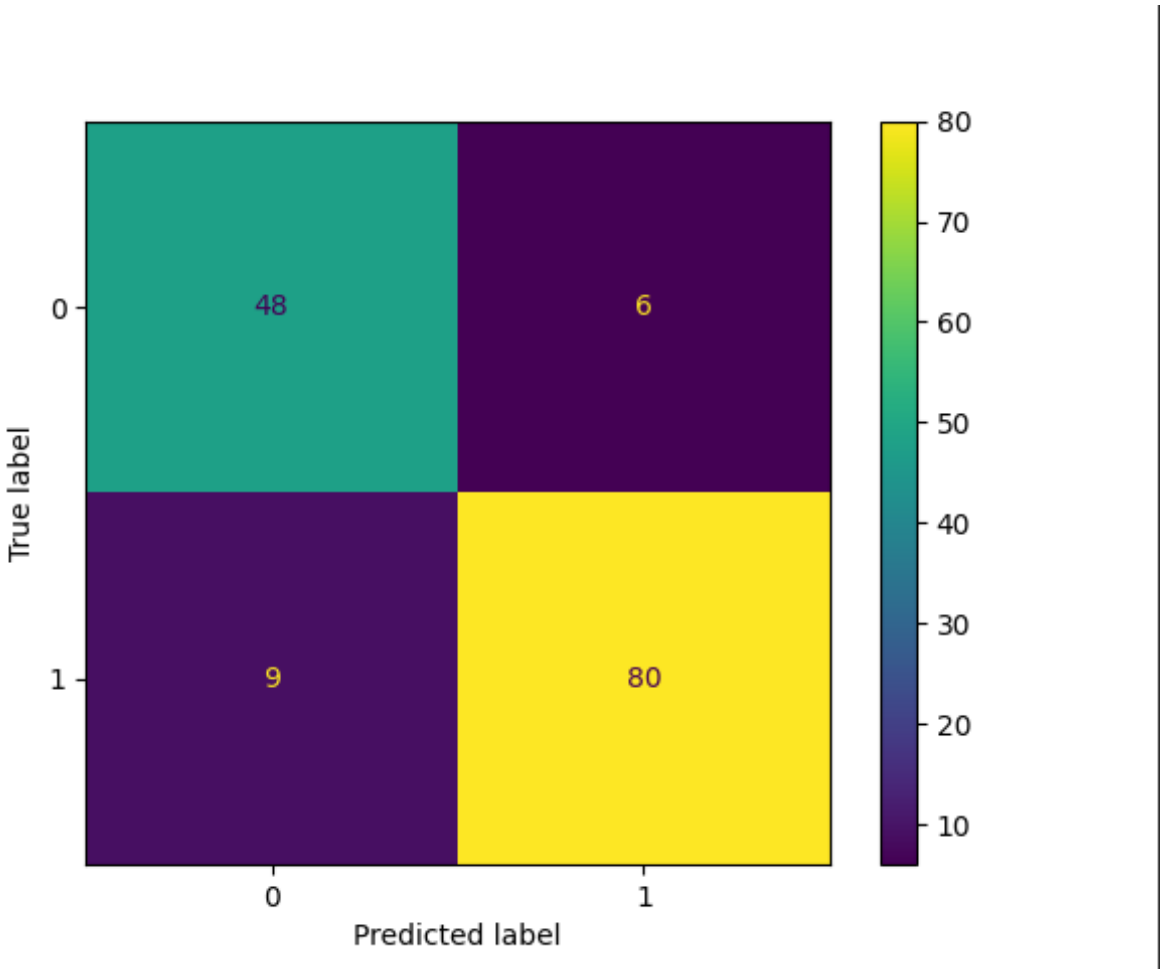
Komentarz: Metoda, w której uzupełniane są brakujące wartości pozwala efektywnie klasyfikować powyższe zbiory danych. Poziom klasyfikacji nie różnił się znacząco w sytuacji gdy zwiększany był odsetek brakujących danych w danym zbiorze. Nie zauważono większych różnić przy stosowaniu do wypełniania średniej arytmetycznej, mediany i mody - wyniki w przypadku kolejnych zbiorów danych są bardzo zbliżone.

6.1.3 Metoda 5.1.3 - Klasyfikacja probabilistyczna

Kolejne eksperymenty polegały na przetestowaniu drzewa decyzyjnego zadanymi(niekompletnymi) zbiorami i zweryfikowaniu dokładności klasyfikacji z wykorzystaniem predykcji probabilistycznej

Tabela 6: Badany zbiór danych - Rak piersi: Zmierzona referencyjna jakość klasyfikatora - w przypadku bez brakujących danych = 0.90 Ilość iteracji: 5

Procent brakujących danych	**Liczba brak. atrybutów**	**Dokładność klasyfikacji probabilistycznej**
20%	**2**	**0.92**
20%	4	0.91
20%	6	0.91
40%	**2**	**0.88**
40%	4	0.88
40%	6	0.84
60%	**2**	**0.93**
60%	4	0.88
60%	6	0.83
80%	**2**	**0.88**
80%	4	0.89
80%	6	0.79

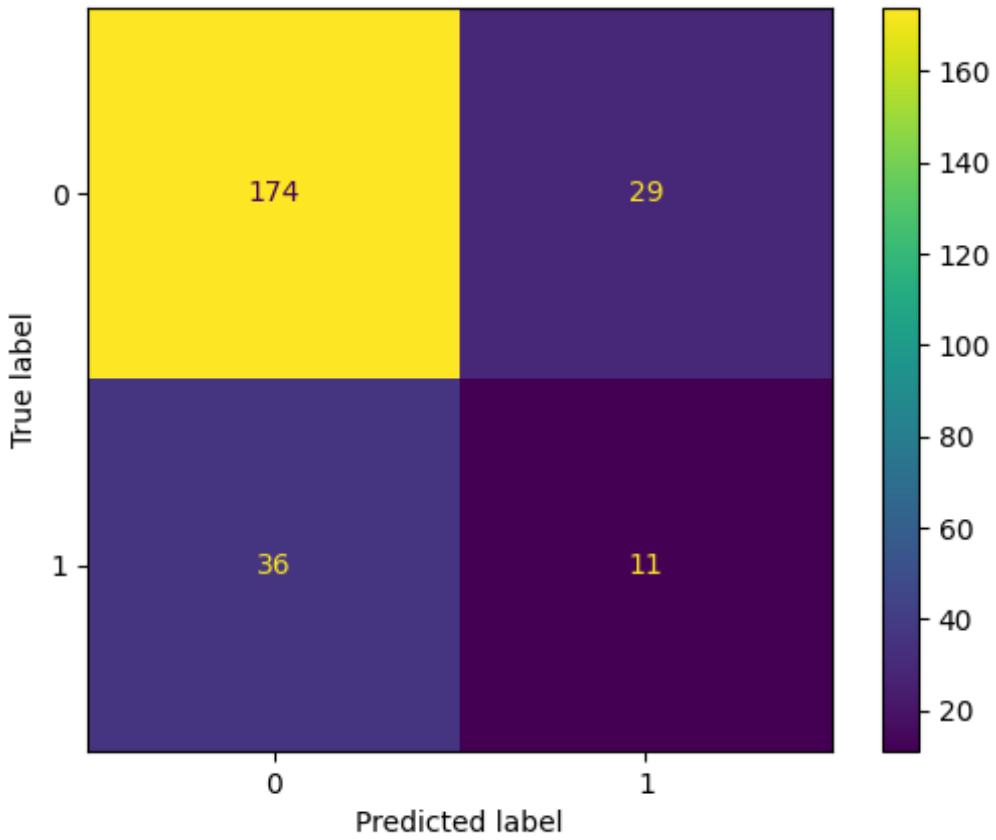


Zrzut ekranu 8: Macierz pomyłek dla klasyfikacji z wykorzystaniem predykcji probabilistycznej - zbiór danych - rak piersi

Komentarz: W przypadku klasyfikacji danych z powyższego zbioru widać, że nie zależnie od procentu brakujących danych metoda predykcji probabilistycznej pozwala dość skutecznie klasyfikować (dokładność na poziomie 0.79-0.92)

Tabela 7: Badany zbiór danych - Ryzyko kredytowe: Zmierzona referencyjna jakość klasyfikatora - w przypadku bez brakujących danych = 0.72 Ilość iteracji: 5

Procent brakujących danych	**Liczba brak. atrybutów**	**Dokładność klasyfikacji probabilistycznej**
20%	**2**	**0.70**
20%	4	0.67
20%	6	0.64
40%	**2**	**0.66**
40%	4	0.65
40%	6	0.63
60%	**2**	**0.54**
60%	4	0.65
60%	6	0.54
80%	**2**	**0.59**
80%	4	0.57
80%	6	0.50

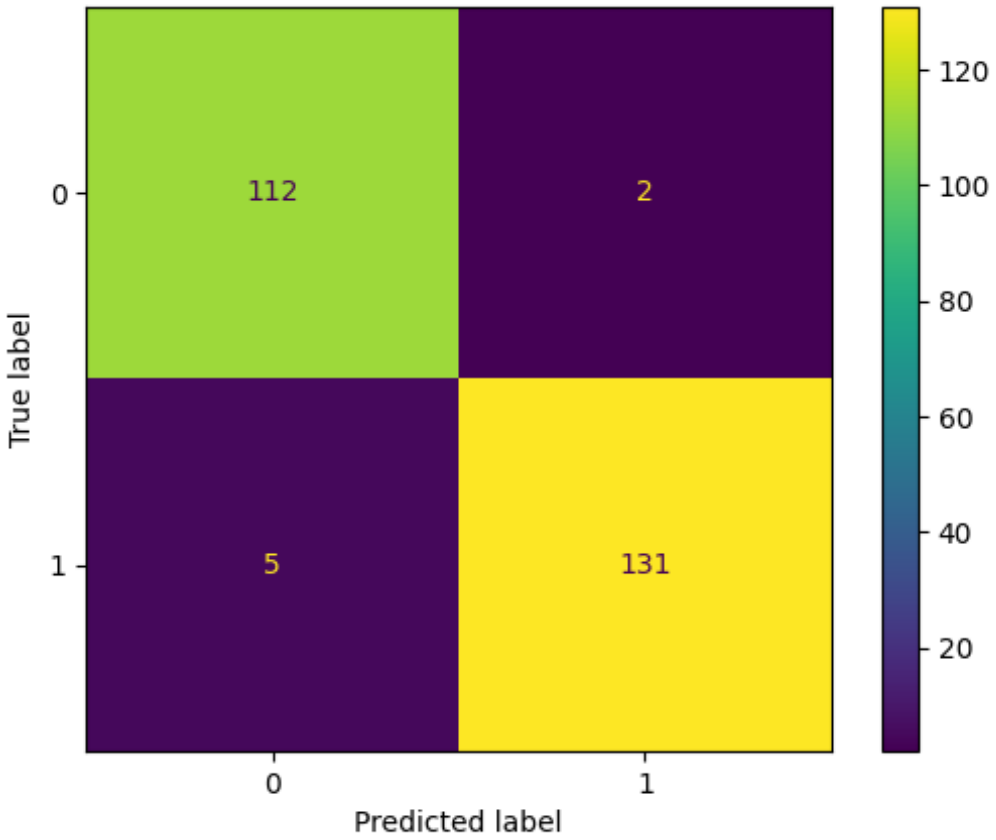


Zrzut ekranu 9: Macierz pomyłek dla klasyfikacji z wykorzystaniem predykcji probabilistycznej - zbiór danych - ryzyko kredytowe

Komentarz: Zbiór danych z ryzykiem kredytowym cechuje się dość dużą entropią, trudno jest sklasyfikować jednoznacznie wiele przykładów. Domyślna jakość klasyfikatora dla tego zbioru danych do 0.72, a w przypadku dodania do zbioru brakujących danych dokładność maleje sukcesywnie do wartośći 0.54 przy 60% brakujących danych.

Tabela 8: Badany zbiór danych - Ryż: Zmierzona referencyjna jakość klasyfikatora - w przypadku bez brakujących danych = 0.99 Ilość iteracji: 5

Procent brakujących danych	**Liczba brak. atrybutów**	**Dokładność klasyfikacji probabilistycznej**
20%	**2**	**0.97**
20%	4	0.96
20%	6	0.89
40%	**2**	**0.98**
40%	4	0.98
40%	6	0.98
60%	**2**	**0.98**
60%	4	0.99
60%	6	0.76
80%	**2**	**0.98**
80%	4	0.97
80%	6	0.85



Zrzut ekranu 10: Macierz pomyłek dla klasyfikacji z wykorzystaniem predykcji probabilistycznej - zbiór danych - ryż - 50% brakujących danych

Komentarz: Klasyfikator domyślnie jest w przypadku tego zbioru danych bardzo dokładny - 0.99. W przypadku brakujących danych, sięgających nawet 80%, metoda probabilistyczna pozwala utrzymać dokładność klasyfikacji na bardzo zadowalającym poziomie. Najgorszy uzyskany wynik to 0.76 dla 60% brakujących danych w przypadku gdzie 6 atrybutów zawierało przykłady z brakującymi danymi.

6.1.4 Metoda 5.1.4 - Przykłady ułamkowe

Zaimplementowaliśmy przykłady ułamkowe do budowania drzewa. Testowanie wykonujemy używając innych wspomnianych metod - zastępowanie przykładami ułamkowymi wykonujemy dla zbioru trenującego, natomiast zbiór testujący przygotowujemy inaczej. Poniżej dwie tabele z uśrednionymi wynikami eksperymentów dla rozmiarów n=200 i n=1000.

Tabela 9: Testy dla 200 wierszy z każdego pliku przy maksymalnej głębokości drzewa = 8. Porównanie z modelem referencyjnym

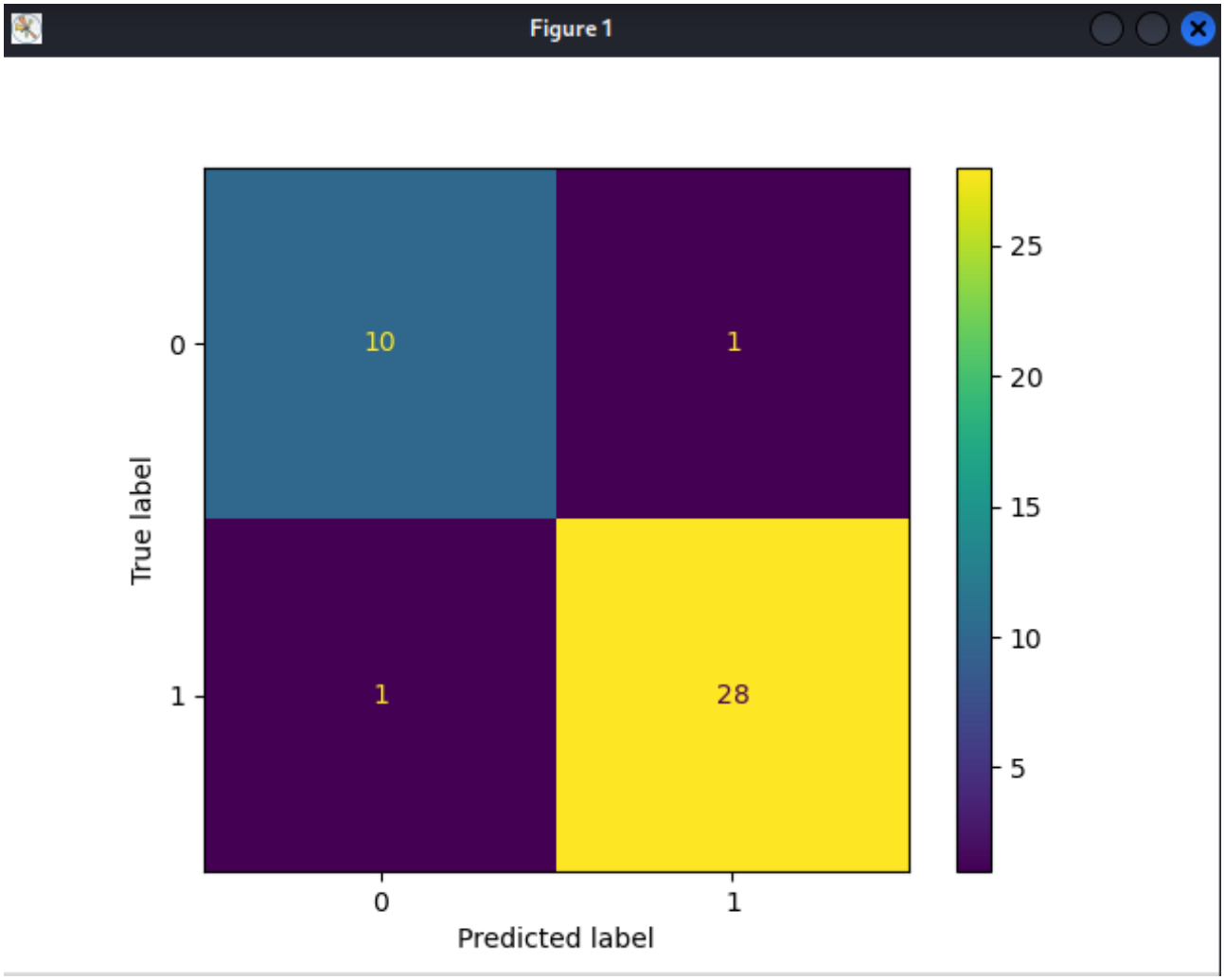
dane	budowanie drzewa		pomijanie		średnia		mediana		moda	
	nasz	referencyjny	nasz	ref.	nasz	ref.	nasz	ref.	nasz	ref.
rak piersi	13s	7ms	1.0	1.0	0.925	0.9	0.925	0.9	0.925	0.9
ryzyko kredytowe	26s	7ms	0.957	0.957	0.717	0.667	0.717	0.667	0.717	0.667
ryż	4s	2ms	1.0	1.0	0.967	0.983	0.967	0.983	0.967	0.983

Tabela 10: Testy dla 1000 wierszy z każdego pliku przy maksymalnej głębokości drzewa = 8. Porównanie z modelem referencyjnym

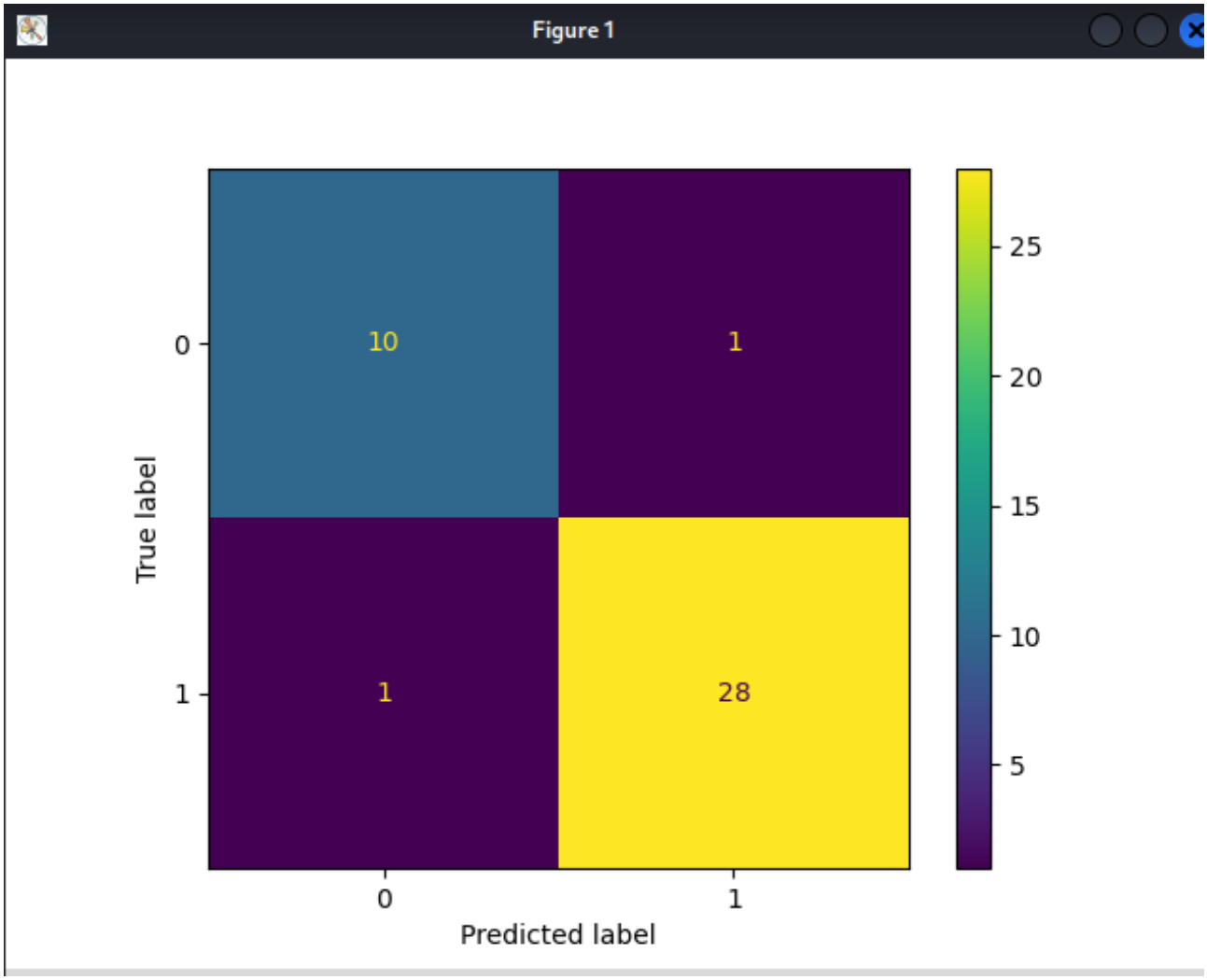
dane	budowanie drzewa		pomijanie		średnia		mediana		moda	
	nasz	referencyjny	nasz	ref.	nasz	ref.	nasz	ref.	nasz	ref.
rak piersi	122s	10ms	1.0	1.0	0.941	0.933	0.941	0.933	0.941	0.933
ryzyko kredytowe	642s	43ms	0.837	0.854	0.797	0.767	0.797	0.767	0.797	0.767
ryż	89s	8ms	1.0	1.0	0.998	1.0	0.998	1.0	0.998	1.0

Widać, że czas budowania drzewa według naszego programu jest niewspółmiernie duży (czas predykcji jest w przybliżeniu taki sam). Jednym z powodów dlaczego tak się dzieje (jedynym, który widzimy), jest to, że zastępowanie brakujących wartości przez przykłady ułamkowe znacząco zwiększa rozmiar zbioru (podczas badań np. z początkowych 80 powstawało 600 przykładów), a ponieważ funkcje od wyznaczania podziału i liczenia entropii przeszukują wielokrotnie całą tablicę, to musimy długo czekać na zbudowanie drzewa.
Dokładność naszego modelu jest zbliżona do modelu referencyjnego.
Predykcja dla średniej, mediany i mody (przy tym samym drzewie) daje przeważnie takie same wyniki, choć nie zawsze.
Ze względu duży czas wykonania przy oryginalnych zbiorach nie podjęliśmy się przebadania usuwania więcej wartości (więcej braków to więcej przykładów ułamkowych) - wstępne próby nie wykonywały się przez co najmniej kilka minut. Ślady tego procesu można znaleźć w kodzie. Uznaliśmy, że usuwanie kolejnych atrybutów dla małych zbiorów nie będzie wiarygodnym testem.

Poniżej macierze pomyłek dla pojedynczego testu (naszego i referencyjnego drzewa) z każdego zbioru przy 200 przykładach (dla ryzyka kredytowego sprawdziliśmy 400, licząc, że wynik będzie lepszy, niestety nie):

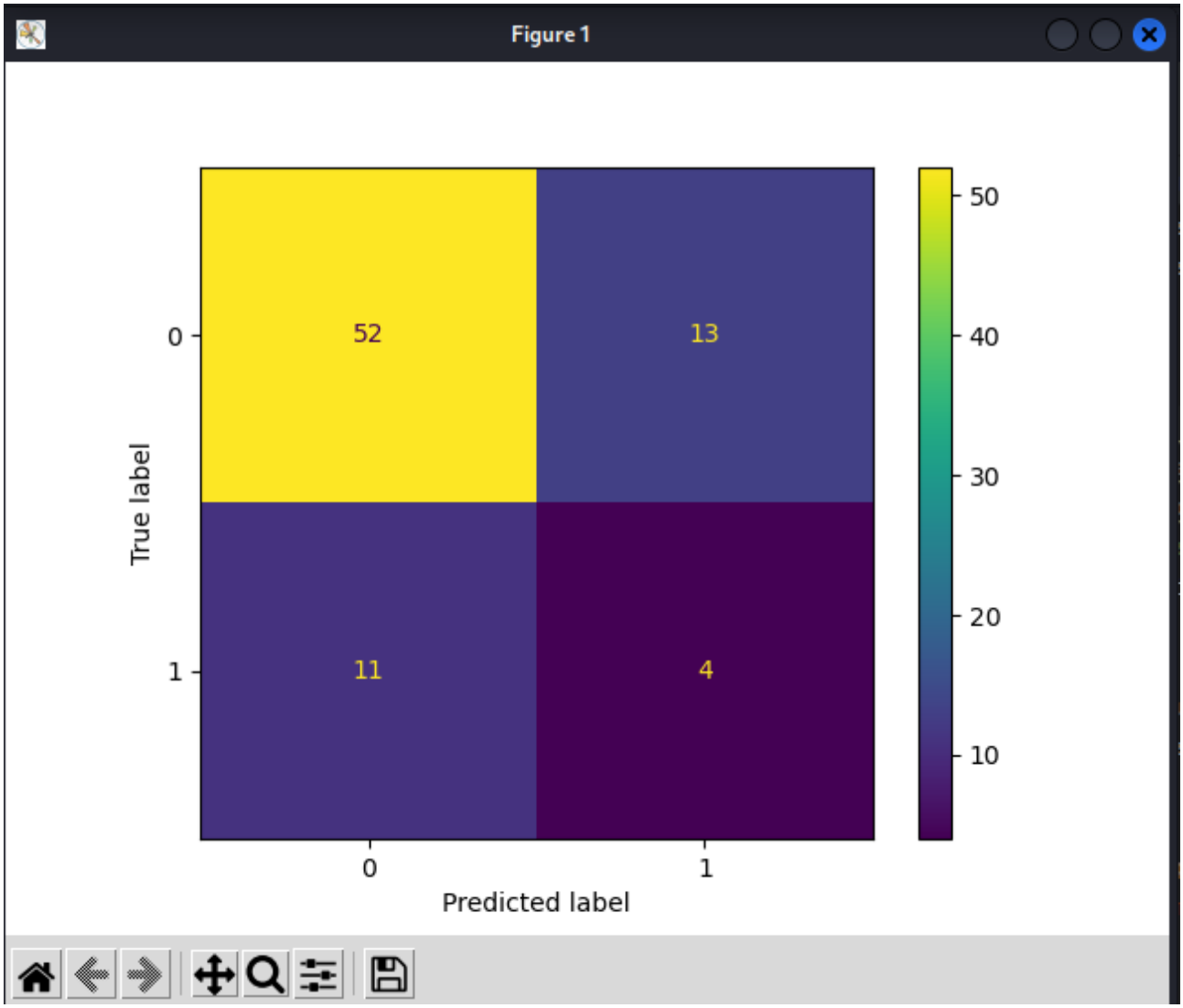


Zrzut ekranu 11: Macierz pomyłek dla klasyfikacji z wykorzystaniem przykładów ułamkowych - zbiór danych - rak piersi, drzewo nasze



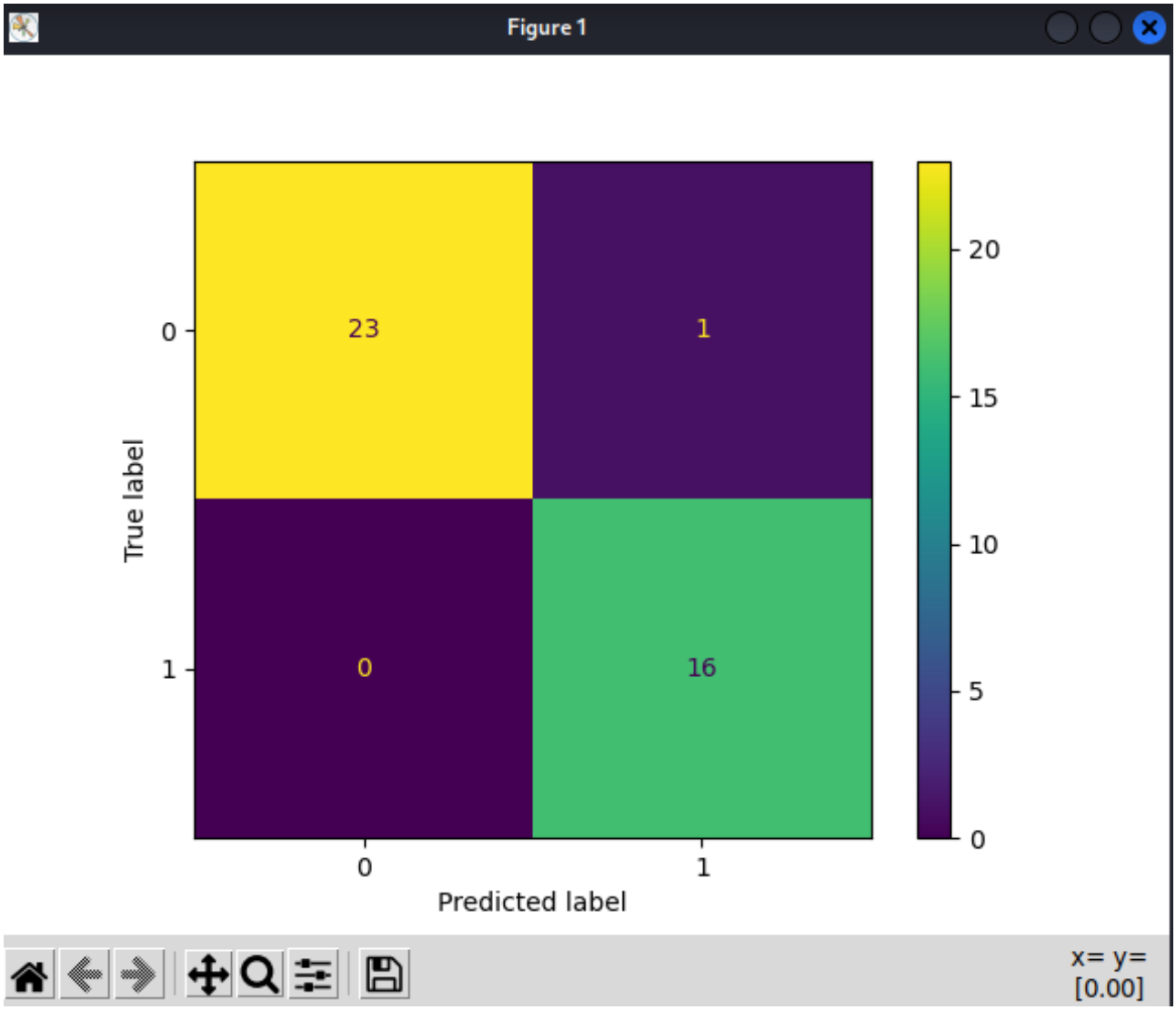
Zrzut ekranu 12: Macierz pomyłek dla klasyfikacji z wykorzystaniem przykładów ułamkowych - zbiór danych - rak piersi, drzewo referencyjne

Zrzut ekranu 13: Macierz pomyłek dla klasyfikacji z wykorzystaniem przykładów ułamkowych - zbiór danych - ryzyko kredytowe, drzewo nasze



Zrzut ekranu 14: Macierz pomyłek dla klasyfikacji z wykorzystaniem przykładów ułamkowych - zbiór danych - rak piersi, drzewo referencyjne

Zrzut ekranu 15: Macierz pomyłek dla klasyfikacji z wykorzystaniem przykładów ułamkowych - zbiór danych - ryż, drzewo nasze



Zrzut ekranu 16: Macierz pomyłek dla klasyfikacji z wykorzystaniem przykładów ułamkowych - zbiór danych - ryż, drzewo referencyjne

Są to dość małe testy, ale widać, że dla raka piersi i ryżu klasyfikujemy dobrze, dla ryzyka kredytowego trochę gorzej.

6.2 Omówienie wyników

Przechodząc do analizy wyników z powyższych tabel, zwłaszcza tych przedstawiających metody najbardziej skuteczne należy stwierdzić, że programy z zaimplementowanymi drzewami i naniesionymi na nie mechanizmami uodparniającymi spełniają swoje zadanie i pozwalają z dość wysoką dokładnością sklasyfikować zadane zbiory danych. Oczywistym faktem, jest zależność dokładności od stopnia brakujących danych(zarówno % całości jak i w przypadku liczności atrybutów, w których brakowało danych). Przypuszczalnie nieznaczną przewagę posiada podejście przykładów ułamkowych, dzięki któremu drzewo było w stanie klasyfikować z największą dokładnością zbiory danych. Jednakże w praktyce nie polecamy tego rozwiązania, tworzenie drzewa taką metodą zajmowało wyraźnie dużo czasu - być może jest to kwestia zastosowania języka wysokopoziomowego python, który charakteryzuje się zdecydowanie długimi czasami kompilacji. Należy zaznaczyć, że podejście z klasyfikacją probabilistyczną również cechuje się dość wysokimi wynikami dokładności. Metoda sprawdza się w sytuacji braku dużej ilości danych (np. 80%). W takiej sytuacji nie zawsze możliwe jest znalezienie podziału zastępczego dającego porównywalny przyrost informacji.

7 Inne Obserwacje

Pobrane zbiory danych w plikach csv posiadały określony porządek (uporządkowanie względem klas) - z tego względu należało je wymieszać przed podjęciem jakiegokolwiek dzielenia na zbiory trenujące i testujące - w sytuacji gdy wszystkie przykłady z kilku tysięcy pierwszych wierszy miały tę samą klasę, skuteczność wynosiła 100%. Dla jednego zbioru danych algorytm nauczył się używać id jako atrybutu (powodowało to jednakże zerową skuteczność predykcji).
Przy testowaniu fragmentów zbiorów warto wymieszać je na początku.
Długi czas budowania modeli dość mocno utrudniał przeprowadzanie eksperymentów.

8 Podsumowanie

Poprzez zaimplementowane klasyfikatory danych, a szerzej pracę nad dużymi zbiorami danych i ich analizą mogliśmy dobrze zrozumieć zagadnienia stojące u podstaw problemu zadania klasyfikacji przy wykorzystaniu drzew decyzyjnych. Wykorzystanie modelów uczenia się, trenowanie modelów które służą później do zadań klasyfikacji jest procesem dającym zauważalne korzyści, raz wytrenowany model jest w stanie poprawnie oszacować kilka zbiorów testowych. Przypomnienie sobie innych standardowych zadań algorytmiki przy pracy na zbiorach było równie cenną lekcją. Realizacja projektu skutecznie przeprowadziła nas przez biblioteki: Numpy, Pandas oraz Sklearn, które oferują rozwiązania przydatne w pracy nad programami uczenia maszynowego