Implementacja drzewa decyzyjnego dla atrybutów ciągłych.

Kinga Świderek, Jakub Kowalczyk

1 Treść zadania

Implementacja drzewa decyzyjnego dla atrybutów ciągłych. Porównanie metod ustalenia liczby i miejsc podziałów przestrzeni ciągłej w testach.

W ramach projektu zbadamy jakości 3 metod dyskretyzacji dla zadanej liczby przedziałów (metoda równoczęstościowa, równoodległościowa i k-średnich) oraz porównamy je z metodami bazującymi na ocenie najlepszego miejsca podziału dla wartości ciągłych danego atrybutu podczas procesu uczenia się drzew decyzyjnych (metoda gini impurity i przyrostu informacji).

Testy zostaną przeprowadzone dla dwóch zbiorów danych: klasyfikacji binarnej i wieloklasowej.

2 Implementacja

Zaimplementowane drzewo jest drzewem binarnym, zdolnym do klasyfikacji binarnej jak i wieloklasowej. Kod został napisany w języku Python. Nasza implementacja składa się z 3 głównych klas:

1. *TreeNode*: reprezentuje pojedynczy węzeł drzewa. Jeśli jest liściem, przechowuje klasę (pole *target*). W przeciwnym wypadku ma pole *test*, które przechowuje funkcję anonimową, zwracającą wartość boolowską, wykorzystywaną do podziału.

2. *DecisionTree*: aby stworzyć drzewo, należy podać jako argumenty dane treningowe (w formacie *pandas DataFrame*), metodę testowania (niżej opisana klasa *TestMethod*) oraz, jako argument opcjonalny, maksymalną wysokość drzewa. W konstruktorze wołana jest funkcja *fit*, która, na podstawie otrzymanych danych, buduje drzewo w sposób rekurencyjny. Na drzewie można następnie zawołać metodę *predict*, aby otrzymać predykcje klas dla zadanego zbioru danych (również w formacie *pandas DataFrame*).

3. *TestMethod*: jest klasą bazową, po której dziedziczą wszystkie zaimplementowane metody tworzenia podziału w drzewie decyzyjnym - *InformationGain*, *GiniImpurity*, *EqualFrequency*, *EqualWidth* oraz *KMeansTest*.

Metody *EqualFrequency*, *EqualWidth* oraz *KMeansTest* przyjmują jako parametr liczbę przedziałów.

Dodaliśmy ograniczenie maksymalnej wysokości drzewa, gdyż dla testów równoczęstościowych zdarzały się przypadki przekroczenia limitu rekursji podczas budowania drzewa. Nie każdy wylosowany podzbiór trenujący powodował ten błąd. Z racji losowej natury tego przypadku, nie udało nam się zbadać jego przyczyn.

3 Plan eksperymentów

3.1 Zbiory danych

Eksperymenty będziemy przeprowadzać na dwóch zbiorach danych.

1. Klasyfikacja binarna. Zbiór do przewidywania cukrzycy u kobiet - na podstawie 8 atrybutów z dziedziny rzeczywistej klasyfikuje na osoby zdrowe (0) i chore (1). Ilość przykładów dla klasy 0: 500; ilość przykładów dla klasy 1: 268. Link do Kaggle.

2. Klasyfikacja wieloklasowa. Zbiór *iris*, klasyfikujący dane z 4 atrybutów ciągłych dotyczących rozmiaru kwiatu na 3 gatunki irysów. Ilość przykładów dla klasy *setosa*: 50; ilość przykładów dla klasy *versicolor*: 50; ilość przykładów dla klasy *virginica*: 50. Link do Kaggle.

Zbiory danych zostaną losowo podzielone na zbiory treningowe i testowe w proporcji 80:20.

3.2 Opis

Dla każdego zbioru danych algorytm zostanie uruchomiony 25 razy w celu uzyskania zagregowanych wyników dla różnych danych treningowych i testowych. Wśród nich znajdzie się średnia, odchylenia standardowe, najlepszy i najgorszy wynik.

Dla każdej z 3 klasycznych metod dyskretyzacji zbadamy różne możliwości ustalenia liczby przedziałów i spróbujemy znaleźć tę optymalną. Dla każdej badanej wartości liczby przedziałów (zbiór liczb całkowitych od 2 do 30), na podstawie tych samych zbiorów treningowych i testowych, będziemy budować drzewo i badać jakość jego predykcji.

Metody gini impurity i przyrostu informacji wymagają podziału na 2 grupy wartości i poprzez ocenianie automatycznie znajdują najlepsze miejsce podziału; skupimy się zatem na porównaniu efektywności takiego rozwiązania względem ręcznego zadawania wymaganej liczby grup pozostałym trzem metodom.

Dodatkowo, porównamy zaimplementowany przez nas model drzewa decyzyjnego z implementacją dostarczaną przez bibliotekę *sci-kit learn*.

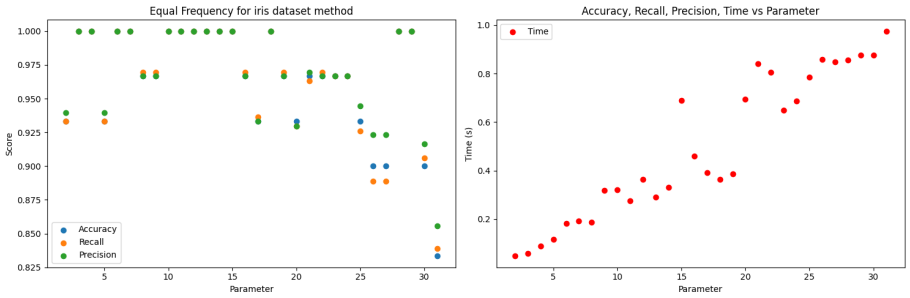
Poza samą jakością modelu, porównany zostanie także sam czas trwania obliczeń. Zwrócimy także szczególną uwagę na stabilność algorytmu dla różnych uruchomień tzn. w jakim stopniu zachowuje się on spójnie i przewidywalnie.

4 Eksperymenty

4.1 Znalezienie liczby przedziałów dla metod dyskretyzacji

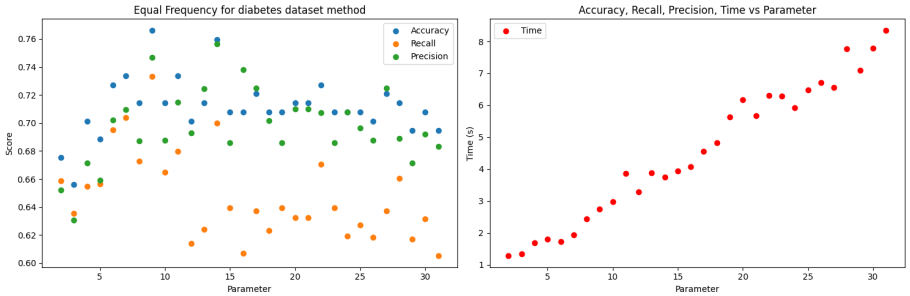
Poniższe wykresy przedstawiają zależności jakości mierzonej przez dokładność, odzysk oraz precyzję od liczby przedziałów. Dodatkowo, mierzony jest czas obliczeń. Wartość maksymalnej wysokości drzewa została ustawiona na 10.

4.1.1 Metoda równoczęstościowa



Rysunek 1: Metoda równoczęstościowa dla zbioru *iris*

Dla zbioru *iris*, liczba przedziałów powyżej 15 znacząco pogarsza wyniki. Za najlepszy parametr uznamy 12, jako środek najdłuższego ciągłego przedziału o najwyższej miarze jakości.

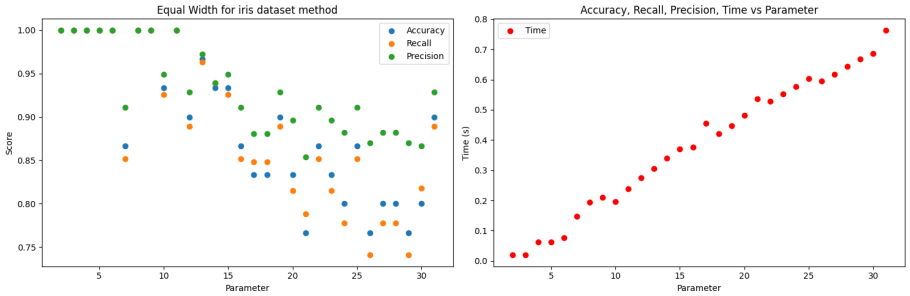


Rysunek 2: Metoda równoczęstościowa dla zbioru *diabetes*

Zbiór *diabetes* jest trudniejszy do nauczenia dla modelu drzewa decyzyjnego. W tym wypadku wyniki są znacznie bardziej rozproszone. Widać jednak znaczny spadek odzysku przy liczbie podziałów ustawionej powyżej 10. Widać

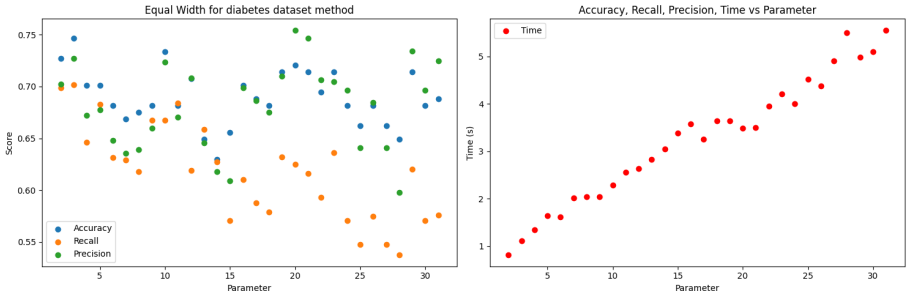
również tendencję wzrostową dla wszystkich miar jakości na przedziale $(2, 10)$. Za najlepszy parametr uznamy 9, jako ostatnia wartość, dla której miary jakości rosną.

4.1.2 Metoda równoodległościowa



Rysunek 3: Metoda równoodległościowa dla zbioru *iris*

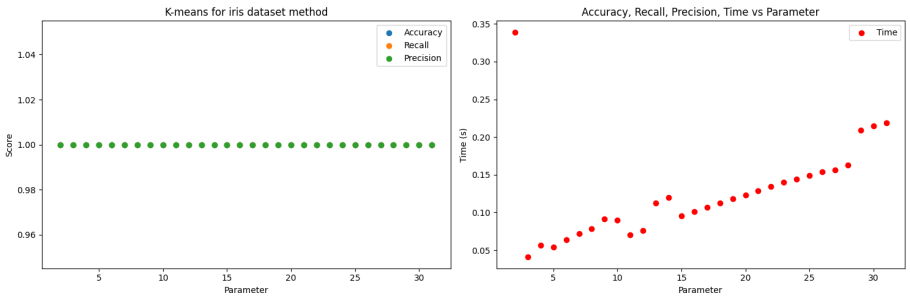
Dla zbioru *iris* wszystkie miary jakości spadają, gdy liczba przedziałów ustalona jest powyżej 6. Ponownie wybierzemy środek najdłuższego ciągłego przedziału najlepszych miar jakości - parametrem będzie ilość przedziałów równa 3.



Rysunek 4: Metoda równoodległościowa dla zbioru *diabetes*

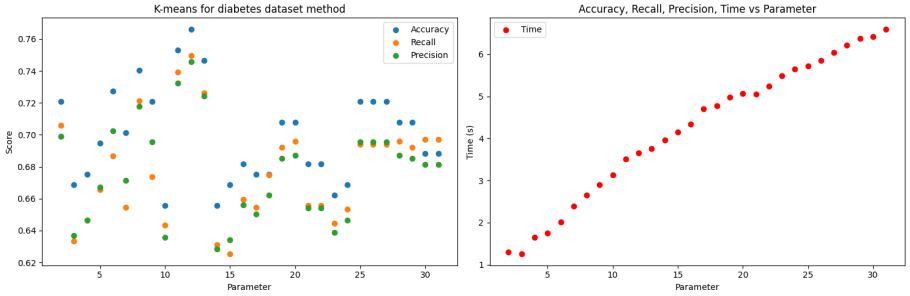
Również w przypadku zbioru *diabetes* lepsze miary osiągamy przy mniejszej liczbie podziałów. Najlepszym wynikiem jest liczba podziałów równa 3.

4.1.3 Metoda k-średnich



Rysunek 5: Metoda k-średnich dla zbioru *iris*

W przypadku metody k-średnich zbiór *iris* jest przewidywany za każdym razem tak samo dobrze - predykcje są tworzone bez żadnych pomyłek. Wybierzemy ilość przedziałów równą 3, z racji najszybszego czasu obliczeń.



Rysunek 6: Metoda k-średnich dla zbioru *diabetes*

Zbiór *diabetes* daje ponownie bardziej rozproszone wyniki. W przedziale $(25, 30)$ są one najbardziej stabilne, jednak najlepsze wyniki otrzymujemy, gdy wartość parametru jest ustalona na 12.

Dla każdej z metod czas budowania drzewa rośnie liniowo wraz ze wzrostem parametrów - warto również to wziąć pod uwagę przy ich doborze.

4.2 Zbiór *iris* - klasyfikacja wieloklasowa

4.2.1 Metoda równoczęstościowa (*EqualFrequency*)

Tabela 1: Wyniki dla metody równoczęstościowej (liczba przedziałów: 12)

Miara	Średnia	Odchylenie standardowe	Maksimum	Minimum
Błąd	0,07	0,06	0,20	0
Dokładność	0,93	0,06	1,00	0,80
Czułość (makro)	0,93	0,06	1,00	0,77
Czułość (mikro)	0,93	0,06	1,00	0,80
Precyzja (makro)	0,93	0,06	1,00	0,77
Precyzja (mikro)	0,93	0,06	1,00	0,80
Miara F (makro)	0,92	0,06	1,00	0,76
Miara F (mikro)	0,93	0,06	1,00	0,80
Czas (sekundy)	0,32	0,05	0,38	0,23

Tabela 2: Macierz pomyłek

	Klasa 1	Klasa 2	Klasa 3
Klasa 1	9,65	0,4	0,05
Klasa 2	0,4	8,85	0,25
Klasa 3	0	1,1	9,3

4.2.2 Metoda równoodległościowa (*EqualWidth*)

Tabela 3: Wyniki dla metody równoodległościowej (liczba przedziałów: 3)

Miara	Średnia	Odchylenie standardowe	Maksimum	Minimum
Błąd	0,05	0,05	0,13	0,00
Dokładność	0,95	0,05	1,00	0,87
Czułość (makro)	0,95	0,05	1,00	0,85
Czułość (mikro)	0,95	0,05	1,00	0,87
Precyzja (makro)	0,95	0,04	1,00	0,85
Precyzja (mikro)	0,95	0,05	1,00	0,87
Miara F (makro)	0,95	0,05	1,00	0,84
Miara F (mikro)	0,95	0,05	1,00	0,87
Czas (sekundy)	0,03	0,01	0,06	0,02

Tabela 4: Macierz pomyłek

	Klasa 1	Klasa 2	Klasa 3
Klasa 1	10,25	0,00	0,00
Klasa 2	0,00	9,35	0,95
Klasa 3	0,00	0,50	8,95

4.2.3 Metoda k-średnich (*KMeansTest*)

Tabela 5: Wyniki dla metody k-średnich (liczba przedziałów: 3)

Miara	Średnia	Odchylenie standardowe	Maksimum	Minimum
Błąd	0,01	0,02	0,07	0,00
Dokładność	0,99	0,02	1,00	0,93
Czułość (makro)	0,99	0,02	1,00	0,93
Czułość (mikro)	0,99	0,02	1,00	0,93
Precyzja (makro)	0,99	0,02	1,00	0,93
Precyzja (mikro)	0,99	0,02	1,00	0,93
Miara F (makro)	0,99	0,02	1,00	0,93
Miara F (mikro)	0,99	0,02	1,00	0,93
Czas (sekundy)	0,05	0,07	0,36	0,02

Tabela 6: Macierz pomyłek

	Klasa 1	Klasa 2	Klasa 3
Klasa 1	10,20	0,05	0,00
Klasa 2	0,10	10,20	0,05
Klasa 3	0,00	0,20	9,20

4.2.4 Metoda Gini impurity (*GiniImpurity*)

Tabela 7: Wyniki dla metody Gini impurity

Miara	Średnia	Odchylenie standardowe	Maksimum	Minimum
Błąd	0,02	0,02	0,07	0,00
Dokładność	0,98	0,02	1,00	0,93
Czułość (makro)	0,98	0,02	1,00	0,94
Czułość (mikro)	0,98	0,02	1,00	0,93
Precyzja (makro)	0,98	0,02	1,00	0,92
Precyzja (mikro)	0,98	0,02	1,00	0,93
Miara F (makro)	0,98	0,02	1,00	0,93
Miara F (mikro)	0,98	0,02	1,00	0,93
Czas (sekundy)	0,21	0	0,22	0,20

Tabela 8: Macierz pomyłek

	Klasa 1	Klasa 2	Klasa 3
Klasa 1	10,25	0,00	0,00
Klasa 2	0,30	9,00	0,00
Klasa 3	0,00	0,30	10,15

4.2.5 Metoda przyrostu informacji (*InformationGain*)

Tabela 9: Wyniki dla metody przyrostu informacji

Miara	Średnia	Odchylenie standardowe	Maksimum	Minimum
Błąd	0,01	0,02	0,03	0,00
Dokładność	0,99	0,02	1,00	0,97
Czułość (makro)	0,99	0,02	1,00	0,95
Czułość (mikro)	0,99	0,02	1,00	0,97
Precyzja (makro)	0,99	0,01	1,00	0,96
Precyzja (mikro)	0,99	0,02	1,00	0,97
Miara F (makro)	0,99	0,02	1,00	0,96
Miara F (mikro)	0,99	0,02	1,00	0,97
Czas (sekundy)	0,25	0	0,25	0,23

Tabela 10: Macierz pomyłek

	Klasa 1	Klasa 2	Klasa 3
Klasa 1	10,1	0,00	0,00
Klasa 2	0,2	9,2	0,00
Klasa 3	0,00	0,1	10,4

4.2.6 Porównanie z drzewem decyzyjnym z biblioteki *sci-kit learn*

Drzewa decyzyjne w bibliotece *sci-kit learn* nie implementują wszystkich przez nas badanych metod tworzenia podziałów. Naszym punktem porównania będzie zatem drzewo o metodzie tworzenia przedziałów - Gini impurity, z ustaloną maksymalną wysokością drzewa na 10.

Tabela 11: Wyniki dla drzewa sci-kit learn

Miara	Średnia	Odchylenie standardowe	Maksimum	Minimum
Błąd	0,01	0,01	0,03	0,00
Dokładność	0,99	0,01	1,00	0,97
Czułość (makro)	0,99	0,01	1,00	0,97
Czułość (mikro)	0,99	0,01	1,00	0,97
Precyzja (makro)	0,99	0,01	1,00	0,96
Precyzja (mikro)	0,99	0,01	1,00	0,97
Miara F (makro)	0,99	0,01	1,00	0,97
Miara F (mikro)	0,99	0,01	1,00	0,97
Czas (sekundy)	0	0	0	0

Tabela 12: Macierz pomyłek

	Klasa 1	Klasa 2	Klasa 3
Klasa 1	10,1	0,05	0,00
Klasa 2	0,1	9,7	0,00
Klasa 3	0,00	0,05	10,00

4.2.7 Wnioski

Wszystkie zaimplementowane metody poszukiwania miejsc podziału okazały się bardzo skuteczne w klasyfikowaniu zbioru *iris*. Dodatkowo, odchylenie standardowe dla każdej miary jakości i dla każdej metody podziału było na poziomie poniżej 0,1, co oznacza, że zaimplementowane przez nas drzewo decyzyjne radzi sobie tak samo dobrze na różnych, losowych podzbiorach tego samego zbioru danych. Najgorzej sprawdziły się metody równoczęstościowe i równoodległościowe ze średnimi wartościami dokładności kolejno: 0,93 oraz 0,95. Zgadza się to z naszymi wstępnymi założeniami, gdyż koncepcyjnie metody te są najprostszymi z zaimplementowanych. Jakość predykcji w porównaniu z implementacją biblioteczną *sci-kit learn* jest identyczna dla metod przyrostu informacji oraz k-średnich. Od zaimplementowanej przez nas metody Gini impurity różni się średnio o 0,1 dla każdej miary jakości - pomimo tej samej metody podziału. Różnica jednak jest nieznaczna i może wynikać z losowości doboru podzbiorów treningowych i testowych. Czas wykonywania programu jest najlepszy dla metody równoodległościowej (średnio 0,03s) i k-średnich (średnio 0,05s). Najwolniejsza okazała się metoda równoczęstościowa - z racji największej liczby przedziałów. Średnio zbudowanie drzewa z wykorzystaniem tej metody zajęło 0,32s. Jest to znacznie gorszy wynik od drzewa *sci-kit learn*, którego zbudowanie zajęło mniej niż 0,01s.

4.3 Zbiór *diabetes* - klasyfikacja binarna

4.3.1 Metoda równoczęstościowa (*EqualFrequency*)

Tabela 13: Wyniki dla metody równoczęstościowej (liczba przedziałów: 9)

Miara	Średnia	Odchylenie standardowe	Maksimum	Minimum
Błąd	0,30	0,02	0,34	0,26
Dokładność	0,70	0,02	0,74	0,66
Czułość	0,45	0,10	0,69	0,26
Precyzja	0,59	0,07	0,70	0,44
Miara F	0,50	0,07	0,65	0,37
Czas (sekundy)	2,66	0,26	3,17	2,17

Tabela 14: Macierz pomyłek

	klasa 0	klasa 1
klasa 0	84,6	16,5
klasa 1	29,35	23,55

4.3.2 Metoda równoodległościowa (*EqualWidth*)

Tabela 15: Wyniki dla metody równoodległościowej (liczba przedziałów: 3)

Miara	Średnia	Odchylenie standardowe	Maksimum	Minimum
Błąd	0,30	0,03	0,36	0,23
Dokładność	0,70	0,03	0,77	0,64
Czułość	0,52	0,06	0,62	0,41
Precyzja	0,56	0,07	0,75	0,44
Miara F	0,54	0,05	0,64	0,45
Czas (sekundy)	1,12	0,05	1,22	1,01

Tabela 16: Macierz pomyłek

	klasa 0	klasa 1
klasa 0	80,6	21,3
klasa 1	24,95	27,15

4.3.3 Metoda k-średnich (*KMeansTest*)

Tabela 17: Wyniki dla metody K-średnich (liczba przedziałów: 12)

Miara	Średnia	Odchylenie standardowe	Maksimum	Minimum
Błąd	0,30	0,04	0,38	0,24
Dokładność	0,70	0,04	0,76	0,62
Czułość	0,53	0,09	0,65	0,35
Precyzja	0,57	0,05	0,66	0,43
Miara F	0,55	0,06	0,63	0,41
Czas (sekundy)	3,44	0,23	3,82	3,00

Tabela 18: Macierz pomyłek

	klasa 0	klasa 1
klasa 0	78,85	21,3
klasa 1	25,5	28,35

4.3.4 Metoda Gini impurity (*GiniImpurity*)

Tabela 19: Wyniki dla metody Gini Impurity

Miara	Średnia	Odchylenie standardowe	Maksimum	Minimum
Błąd	0,30	0,03	0,34	0,25
Dokładność	0,70	0,03	0,75	0,66
Czułość	0,58	0,06	0,72	0,50
Precyzja	0,55	0,05	0,64	0,48
Miara F	0,56	0,04	0,66	0,52
Czas (sekundy)	12,28	0,53	13,47	11,31

Tabela 20: Macierz pomyłek

	klasa 0	klasa 1
klasa 0	77,6	24,55
klasa 1	22,1	29,75

4.3.5 Metoda przyrostu informacji (*InformationGain*)

Tabela 21: Wyniki dla metody przyrostu informacji

Miara	Średnia	Odchylenie standardowe	Maksimum	Minimum
Błąd	0,29	0,02	0,32	0,21
Dokładność	0,71	0,02	0,79	0,68
Czułość	0,57	0,05	0,68	0,47
Precyzja	0,60	0,06	0,73	0,49
Miara F	0,58	0,03	0,65	0,49
Czas (sekundy)	10,70	0,43	11,73	10,10

Tabela 22: Macierz pomyłek

	klasa 0	klasa 1
klasa 0	78,95	21,00
klasa 1	23,30	30,75

4.3.6 Porównanie z drzewem decyzyjnym z biblioteki *sci-kit learn*

Tabela 23: Wyniki dla drzewa sci-kit learn

Miara	Średnia	Odchylenie standardowe	Maksimum	Minimum
Błąd	0,30	0,04	0,37	0,22
Dokładność	0,70	0,04	0,78	0,63
Czułość	0,58	0,07	0,70	0,43
Precyzja	0,56	0,08	0,73	0,44
Miara F	0,57	0,06	0,67	0,45
Czas (sekundy)	0	0	0	0

Tabela 24: Macierz pomyłek

	klasa 0	klasa 1
klasa 0	77,6	23,8
klasa 1	22,3	30,3

4.3.7 Wnioski

Zbiór *diabetes* okazał się trudniejszy do przewidywania dla drzewa decyzyjnego od zbioru *iris*. Może to wynikać z dwukrotnie większej liczby atrybutów bądź większej dysproporcji w ilości reprezentantów klas - osób klasyfikowanych jako zdrowe (0) jest niemal dwukrotnie więcej od osób chorych (1). W tym przypadku wszystkie metody okazały się równie skuteczne, z dokładnością predykcji na poziomie 70%. Jest to również spójne z wynikami drzewa implementowanego

przez *sci-kit learn*. Czas budowania drzewa okazał się wielokrotnie dłuższy od czasu dla zbioru *iris*. Jest to spowodowane dwukrotnie większą ilością atrybutów oraz znacznie większą liczbą przykładów - dla zbioru *diabetes* jest ich 768, natomiast dla zbioru *iris* 150. Najgorszą, pod względem czasu, metodą okazała się Gini impurity (12,28s), a najlepszą metoda równoodległościowa (1,12s). Jest ona wciąż wielokrotnie wolniejsza od drzewa *sci-kit learn*, które również dla tego zbioru danych budowało się mniej niż 0,01s. Po zbadaniu predykcji na zbiorze treningowym, odnotowaliśmy dokładność 0,84 dla metody równoczęstościowej oraz 0,92 - 0,94 dla pozostałych metod. Wydaje się więc, że ma ona nieco mniejszą tendencję do przeuczania, lecz mogła też po prostu mieć mniejszą skuteczność dla całego zbioru. Gotowa implementacja z *sci-kit learn* również wykazuje taką samą tendencję.

5 Podsumowanie

Dobór metody podziału atrybutów ciągłych może zależeć od typu zadania. W przypadku zbioru *iris* widoczne były różnice, z najgorszą jakością metody równoczęstościowej. W przypadku zbioru *diabetes* wszystkie metody okazały się równie skuteczne. Z racji, że metody Gini impurity oraz przyrostu informacji nie wymagają strojenia parametrów, mogą one być najłatwiejsze w użyciu, a zarazem nie dają gorszych wyników; ponadto, w porównaniu z metodami równoczęstościowymi oraz równoodległościowymi uzyskane wyniki są lepsze. W naszych testach okazały się jednak najwolniejszymi metodami. Warto odnotować, że najwyższy możliwy maksymalny oraz minimalny wynik zaobserwowano dla metody przyrostu informacji. Wyróżniła się ona również najlepszą stabilnością (najmniejsze odchylenie standardowe) oraz najwiekszą miarą F (średnią harmoniczną precyzji i odzysku). W związku z tym, w przypadku naszych zbiorów danych wytypowalibyśmy ją na faworyta z całego zestawienia.

We wszystkich testach odchylenie standardowe dla każdej z miar okazało się bardzo niskie - na poziomie poniżej 0,1. Oznacza to, że napisany przez nas algorytm jest stabilny przy uruchomieniu na różnych zbiorach. Warto wspomnieć, że sam algorytm drzewa decyzyjnego jest deterministyczny dla tych samych danych, jednak jego względnie spójne zachowanie przy różnym podziale zbioru na testowy i treningowy jest pożądaną cechą.

Drzewa decyzyjne zawsze mają pewną tendencję do przeuczenia - jeśli dla zbioru *diabetes* spróbujemy dokonać predykcji na danych treningowych, otrzymamy dokładność na poziomie ok. 90%. Aby temu zapobiec można wykorzystać algorytm lasu losowego, który, przez dodanie szumu losowego, mniej dostosowuje się do danych treningowych.

Ze względu na to, że 3 opisane metody dyskretyzacji bazowały na wyborze najlepszego testu licząc przyrost informacji, należy zauważyć, że najlepiej sprawdziła się klasyczna metoda ustalania punktów podziału dla inf gain (branie średniej z dwóch punktów po uszeregowaniu niemalejąco). Większa liczba przedziałów niż 2 oraz użycie nieco bardziej skomplikowanych metod nie dało wcale lepszych rezultatów.

Po porównaniu algorytmu z gotową implementacją z *sci-kit learn* widzimy pole do poprawy pod kątem optymalizacji czasu obliczeń. Jesteśmy jednak zadowoleni, że udało nam się uzyskać niemal identyczną jakość predykcji oraz, że nasz algorytm nie odstaje pod kątem zdolności do generalizacji i stabilności dla różnych podziałów zbioru danych.