

UMA Projekt – Ewolucja drzewa

Dokumentacja końcowa

Skład zespołu:

Jakub Proboszcz 318713

Paweł Kochański 318673

1. Temat projektu

„Ewolucja drzewa

Tworzenie drzewa decyzyjnego przy pomocy algorytmu ewolucyjnego. Zwykle klasyfikatory budowane są w oparciu o metodę zachłanną - w kolejnym kroku wybieramy lokalnie najlepszy podział. Takie podejście jest bardzo szybkie jednak nie zawsze prowadzi do utworzenia optymalnej struktury drzewa.”

2. Ustalenia z dokumentacji wstępnej i konsultacji

Doprecyzowanie tematu

Zaimplementowanie algorytmu tworzącego na podstawie danych wejściowych drzewo decyzyjne z użyciem metod typowych dla algorytmu ewolucyjnego. Algorytm ewolucyjny umożliwia eksplorację różnych konfiguracji, dążąc do znalezienia bardziej globalnie optymalnego drzewa decyzyjnego, w odróżnieniu od zachłannych metod stosowanych w zwykłych klasyfikatorach. Zbadanie efektywności powstałych drzew i porównanie ich z istniejącą implementacją algorytmu ID3.

Wybrany wariant drzewa decyzyjnego

W używanym przez nas wariantcie drzewa decyzyjnego każdemu węzłowi niebędącemu liściem przypisany jest **atrybut** $a(x)$ i **wartość graniczna** A . Każdy taki węzeł ma 2 dzieci - lewe jest wybierane, jeżeli (dla obecnie rozważanego przykładu \bar{x}) spełniony jest warunek $a(\bar{x}) < A$, a prawe w przeciwnym przypadku ($a(\bar{x}) \geq A$).

Każdemu liściowi przyporządkowana jest **klasa**, przydzielana przykładom które do tego liścia trafiają.

Z takich drzew składa się populacja w zaimplementowanym przez nas algorytmie ewolucyjnym.

Inicjalizacja populacji

Każdy węzeł każdego drzewa w populacji startowej jest generowany następująco:

Jeżeli osiągnięto maksymalną głębokość drzewa, lub z prawdopodobieństwem `leaf_probability(depth)`, węzeł będzie liściem. Jeżeli węzeł jest liściem, przypisywana jest mu klasa odpowiadająca klasie większościowej (w przypadku wieloklasowym - najliczniejszej) z części zbioru danych, która trafia do tego liścia.

Jeżeli w tym podzbiorze jest kilka równolicznych najliczniejszych klas, wybierana jest losowa z nich.

Jeżeli węzeł nie jest liściem, losowany jest atrybut, według którego następuje podział. Następnie, granica podziału jest losowana z dziedziny tego atrybutu. Dzieci tego węzła są generowane rekurencyjnie; zbiór danych otrzymany przez ten węzeł jest dzielony według wylosowanego atrybutu i granicy podziału, i podzbiory są przekazywane dzieciom.

Reprodukcja

Wykonaliśmy eksperymenty z użyciem 4 wariantów reprodukcji: proporcjonalnej, rangowej, progowej i turniejowej. W naszym algorytmie ewolucji drzewa działają one tak samo, jak w zwykłym algorytmie ewolucyjnym.

Mutacja

Mutacja polega na wylosowaniu jednego spośród węzłów drzewa. Z prawdopodobieństwem `leaf_inner_swap_probability` jest on zamieniany z liścia na węzeł wewnętrzny lub odwrotnie.

Przy zmianie z liścia na węzeł wewnętrzny, atrybut, granica podziału i dzieci są generowane tak, jak w przypadku inicjalizacji, z tym, że dzieci zawsze będą liśćmi. Zamiana nie ma skutku, jeżeli drzewo przekroczyłoby maksymalną głębokość. Przy zmianie z węzła wewnętrznego na liść, klasa nowego liścia jest ustalana na Jeżeli nie zachodzi taka zamiana,

Krzyżowanie

Krzyżowanie polega na wyborze dwóch węzłów w obu krzyżowanych drzewach oraz zamianie ich miejscami, razem z ich poddrzewami.

Jeżeli jedno z drzew potomnych ma głębokość większą niż parametr `max_depth`, to zamiast niego jest zwracany jeden z rodziców.

Sukcesja

Wykonywaliśmy eksperymenty z użyciem 2 wariantów sukcesji: generacyjnej i elitarnej. W naszym algorytmie ewolucji drzewa działają one tak samo, jak w zwykłym algorytmie ewolucyjnym.

Funkcja oceny

Funkcją oceny używaną przez nas jest dokładność (*accuracy*), czyli iloraz liczby przykładów ze zbioru treningowego, któremu dane drzewo przydzieliło właściwą klasę oraz liczby wszystkich przykładów w zbiorze treningowym. Funkcja ta jest **maksymalizowana**.

Wybrane zbiory danych

Wybraliśmy pięć zbiorów danych:

nazwa zbioru danych	liczba atrybutów	liczba klas	liczba przykładów
glass_identification	9	6	214
dry_bean_dataset	16	7	13611
breast_cancer_wisconsin_diagnostic	30	2	569
wine	13	3	178
high_diamond_ranked_10min	38	2	9879

Hiperłącza do nich oraz licznosci ich klas są wymienione poniżej.

Zbiór danych glass_identification [link]:

Przewidywanie rodzaju szkła na podstawie jego własności fizykochemicznych.

nazwa klasy	liczba wystąpień
1	70
2	76
3	17
5	13
6	9
7	29

Zbiór danych dry_bean_dataset [link]:

Przewidywanie gatunku suszonej fasoli na podstawie cech zdjęcia pojedynczej fasolki.

nazwa klasy	liczba wystąpień
BARBUNYA	1322
BOMBAY	522
CALI	1630
DERMASON	3546
HOROZ	1928
SEKER	2027
SIRA	2636

Zbiór danych breast_cancer_wisconsin_diagnostic [link]:

Diagnoza rodzaju raka piersi na podstawie właściwości komórek pobranych z piersi.

nazwa klasy	liczba wystąpień
B	357
M	212

Zbiór danych wine [link]:

Przewidywanie rodzaju wina na podstawie jego cech fizykochemicznych.

nazwa klasy	liczba wystąpień
1	59
2	71
3	48

Zbiór danych `high_diamond_ranked_10min` [link]:

Przyporządkowanie wyników (zwycięstwo/porażka) meczy rankingowych w grze „League of Legends” na podstawie statystyk meczu pobranych w jego dziesiątej minucie trwania.

nazwa klasy	liczba wystąpień
0	4949
1	4930

Metoda referencyjna

Wyniki uzyskane przez nasz algorytm porównujemy z wynikami istniejącej implementacji algorytmu ID3: ID3 - hiperłącze

3. Poprawność implementacji

W celu weryfikacji poprawności implementacji algorytmu, przygotowaliśmy testy jednostkowe do niektórych jego części składowych. Są one zawarte w plikach `test_tree.py`, `test_genetic_operations.py` oraz `test_succession.py`. Ponadto, sprawdziliśmy poprawność reprodukcji w pliku `manual_testing.ipynb` - rysujemy histogram przybliżający, jak zostają wybrane osobniki w każdej z reprodukcji.

Poprawność całości implementacji algorytmu pokazuje porównanie z wynikami algorytmu ID3, zawarte w poniższej sekcji.

4. Eksperymenty

Przeszukiwanie po hipersiatce hiperparametrów

Skrypt wykonujący poniższy eksperyment jest zawarty w pliku `experiments_evotree.py`. Dane z eksperymentów były agregowane i rysowane na wykresach skryptem `aggregate_results.py`. Dodatkowe eksperymenty z poniższej sekcji były wykonywane z pomocą skryptu `single_experiment.py` - hiperparametry były określane bezpośrednio w skrypcie.

Uruchomiliśmy nasz algorytm dla 128 różnych kombinacji hiperparametrów. Liczba ta została ustalona na podstawie przewidywanego czasu obliczeń - obliczenia zajęły kilka dni. Sprawdzane były następujące wartości każdego z hiperparametrów:

- `max_depth` - maksymalna głębokość drzewa w populacji - wartości 5 oraz 20;
- `reproduction` - sprawdzane były reprodukcje: proporcjonalna, rangowa z parametrami rozmiarem turnieju 2;
- `mutation_probability` - prawdopodobieństwo zajścia mutacji dla każdego drzewa w populacji - wartości 0,4 oraz 0,8;
- `leaf_inner_swap_probability` - prawdopodobieństwo zamiany liścia w węzeł wewnętrzny lub odwrotnie podczas mutacji - wartości 0 oraz 0,3;
- `crossover_probability` - prawdopodobieństwo zajścia krzyżowania dla każdej pary drzew - wartości 0 oraz 0,4;
- `succession` - sprawdzane były sukcesje: generacyjna oraz elitarna dla rozmiaru elity 2.

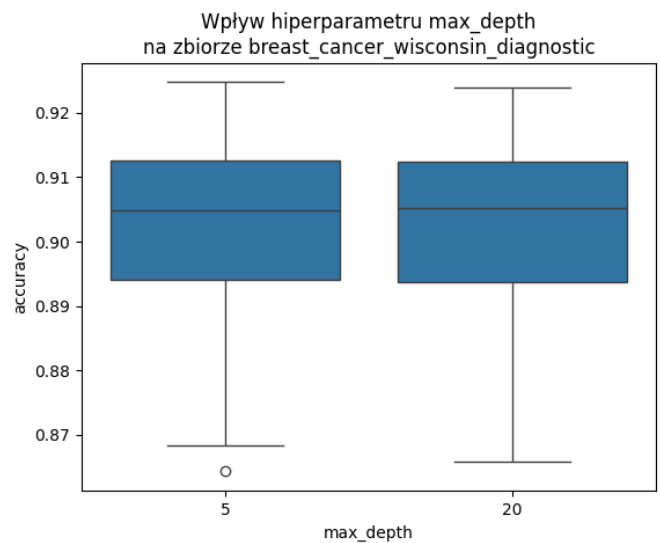
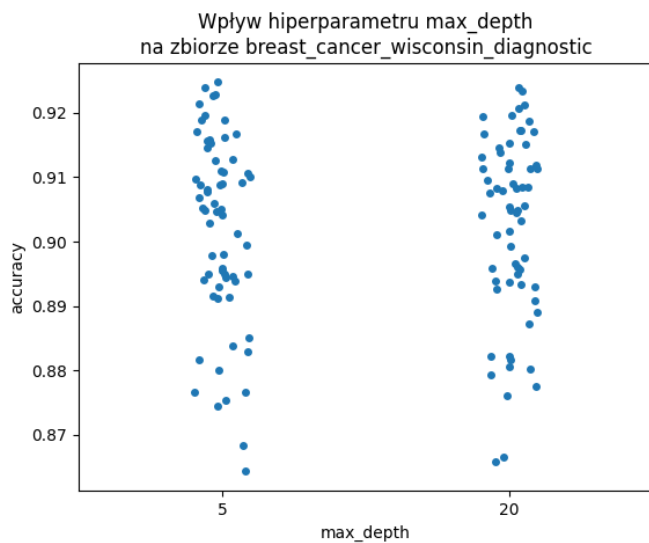
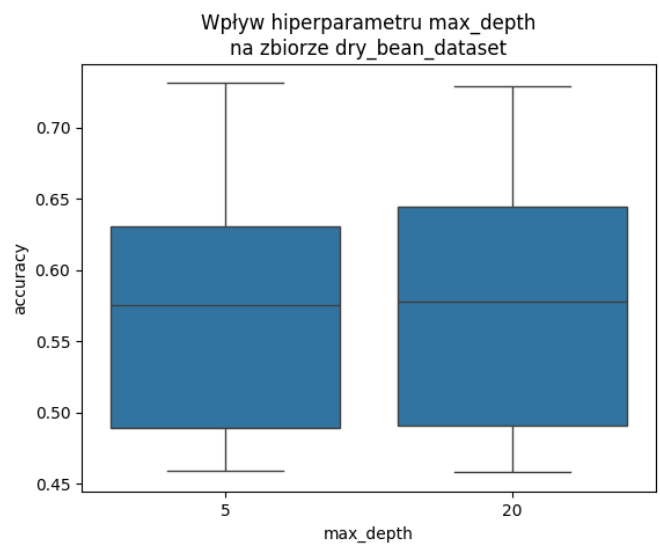
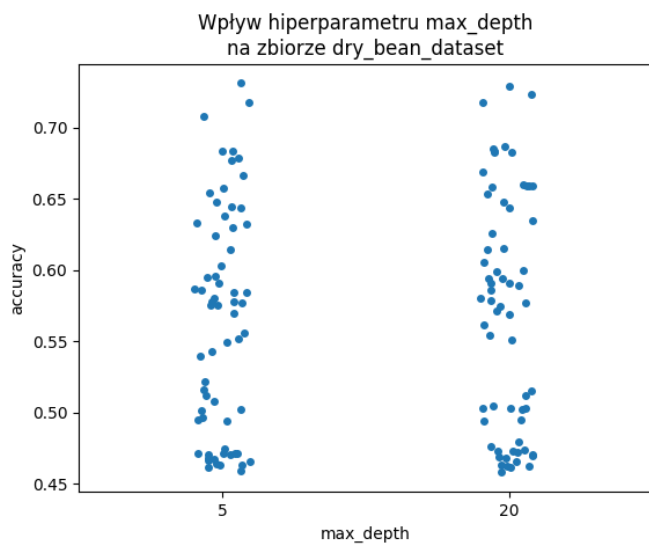
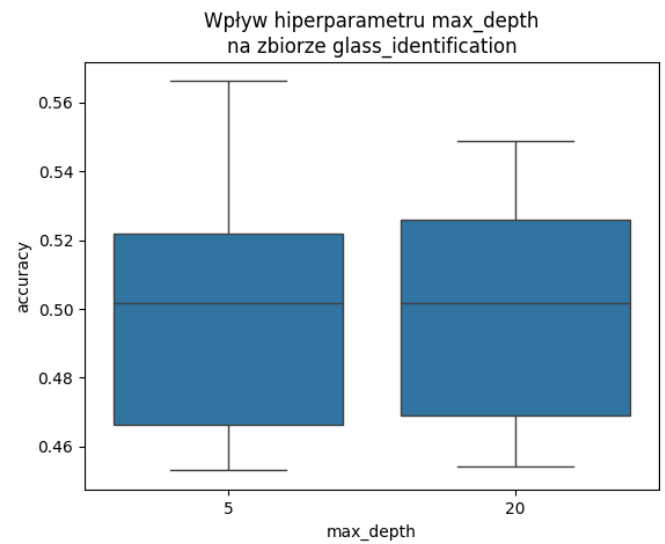
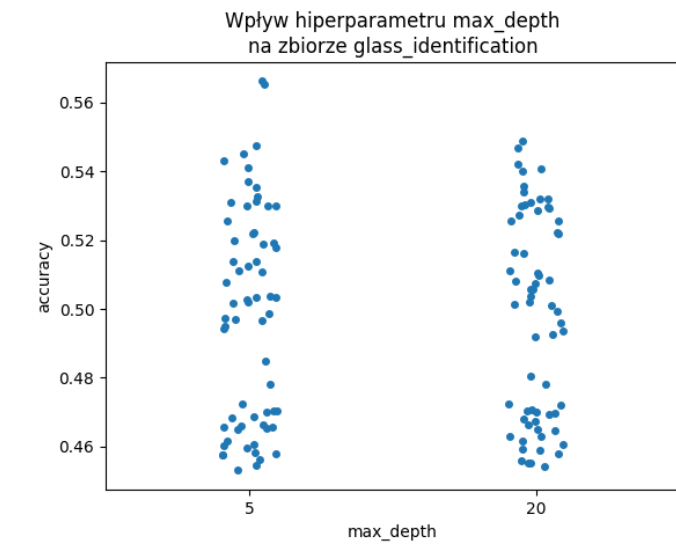
Prawdopodobieństwo, że węzeł będzie liściem podczas inicjalizacji ustaliliśmy na $1 - \left(\frac{1}{2}\right)^g$, gdzie g to głębokość tego węzła (korzeń ma głębokość 0).

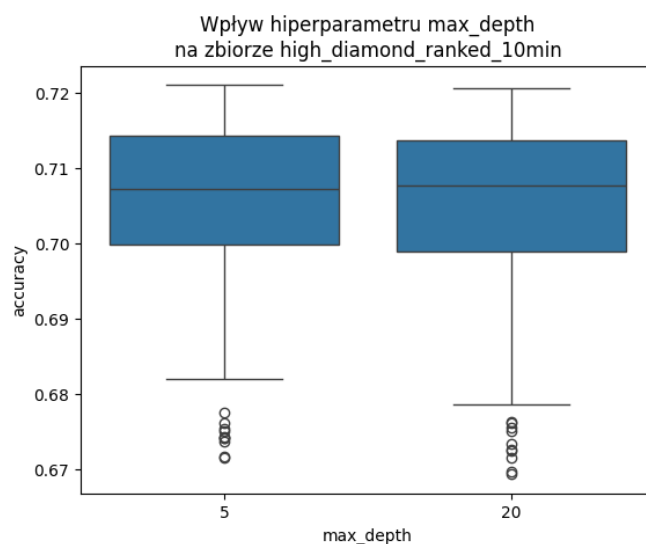
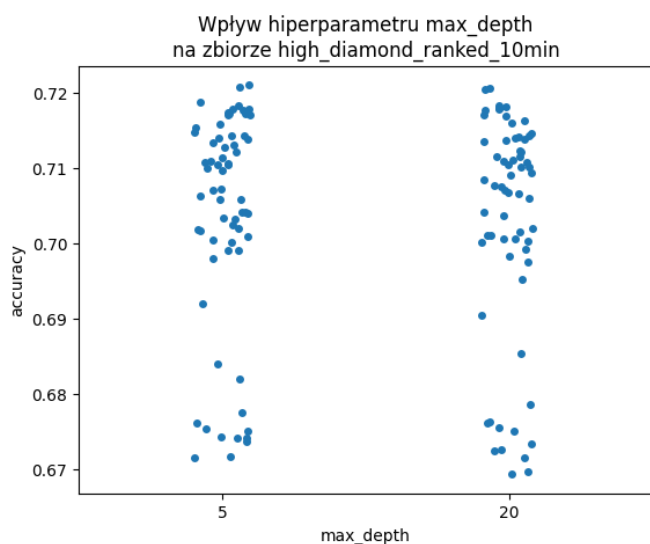
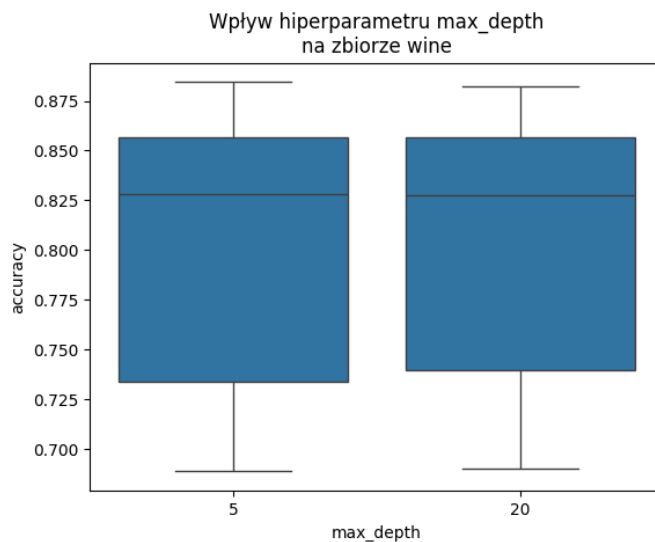
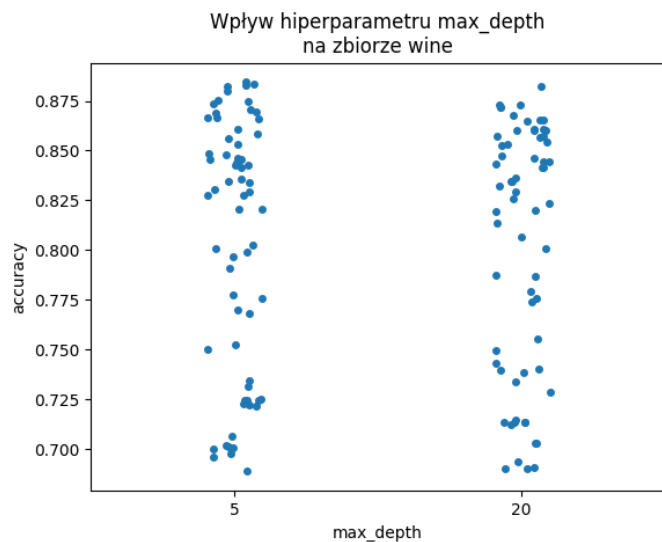
Przyjęliśmy rozmiar populacji równy 20 - czas działania algorytmu nie pozwolił na optymalizację tego hiperparametru.

Przyjęliśmy maksymalną liczbę iteracji równą 500; obliczenia są również zatrzymywane gdy ocena najlepszego drzewa nie poprawi się w ciągu 50 kolejnych iteracji. Warunek ten jest sprawdzany co 50-tą iterację.

Eksperyment został wykonany dla wszystkich 5 zbiorów danych. Z ich wszystkich został wydzielony zbiór testowy wielkości 0,2 całego zbioru. Dla każdej kombinacji hiperparametrów wykonano 5-krotnej walidacji krzyżowej na zbiorze bez zbioru testowego, dla każdego zbioru danych. Na poniższych wykresach są przedstawione uśrednione dokładności na zbiorach walidacyjnych z tych algorytmu.

Wpływ hiperparametru max_depth na działanie algorytmu

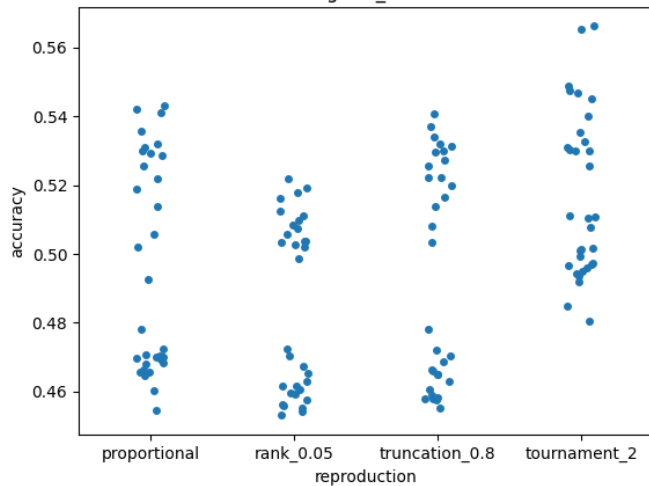




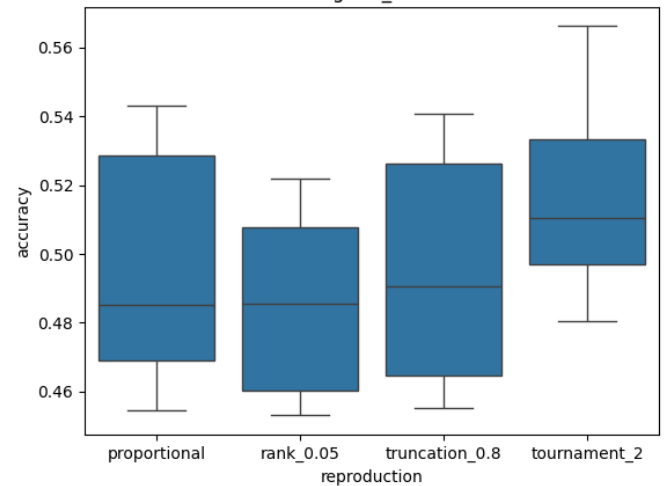
Jak widać na wykresach, wartość hiperparametru `max_depth` nie wpłynęła znacząco na wyniki algorytmu niższe wartości `max_depth` - skoro płytsze drzewa wystarczają, to nie ma sensu próbować konstruować głębszych, dla których operacje (w szczególności predykcja) są wolniejsze, i występuje większe ryzyko nadmiernego dopasowania.

Wpływ hiperparametru reproduction na działanie algorytmu

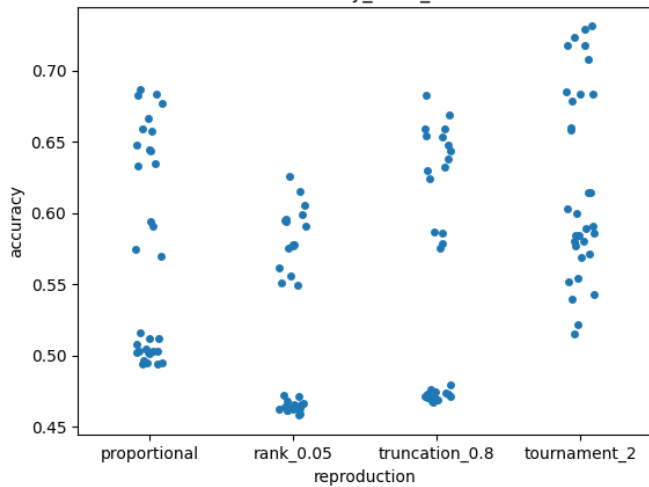
Wpływ hiperparametru reproduction
na zbiorze glass_identification



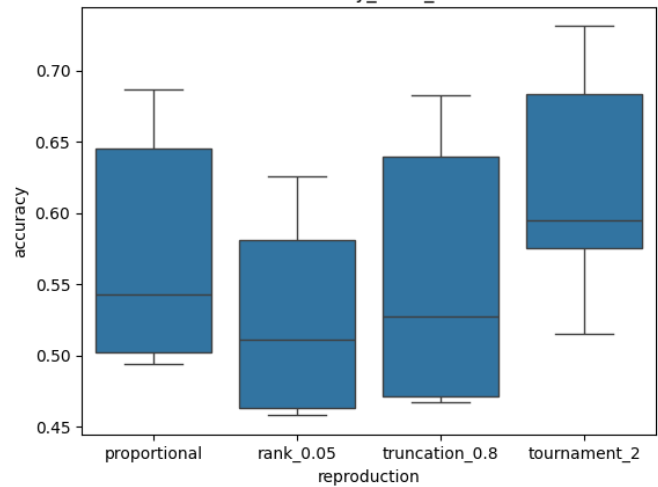
Wpływ hiperparametru reproduction
na zbiorze glass_identification



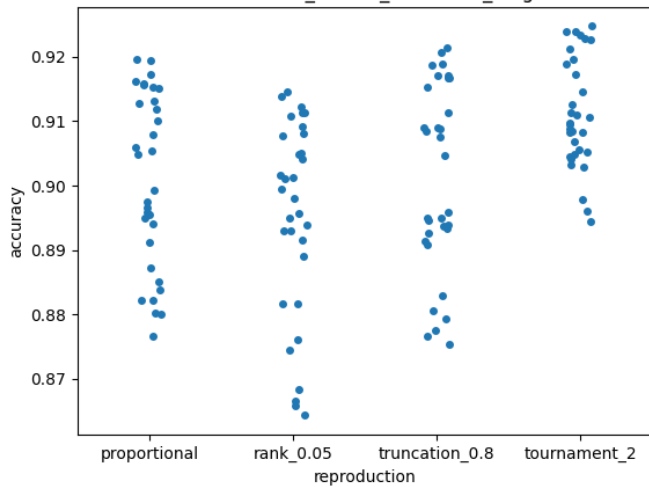
Wpływ hiperparametru reproduction
na zbiorze dry_bean_dataset



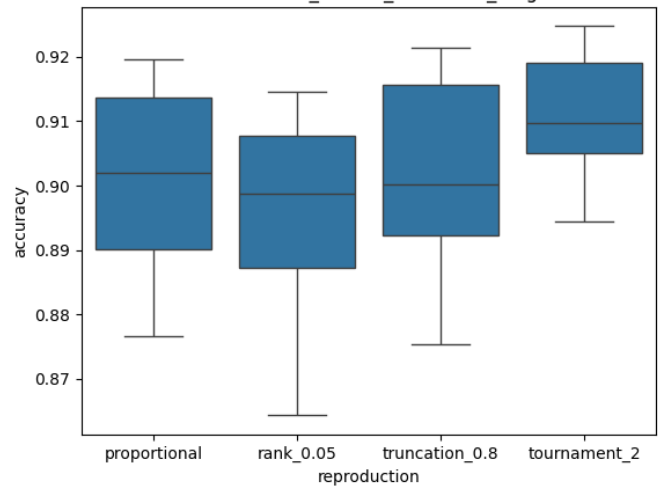
Wpływ hiperparametru reproduction
na zbiorze dry_bean_dataset



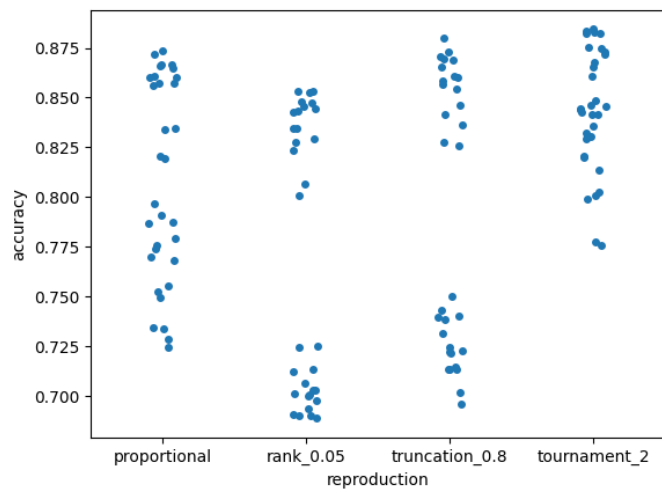
Wpływ hiperparametru reproduction
na zbiorze breast_cancer_wisconsin_diagnostic



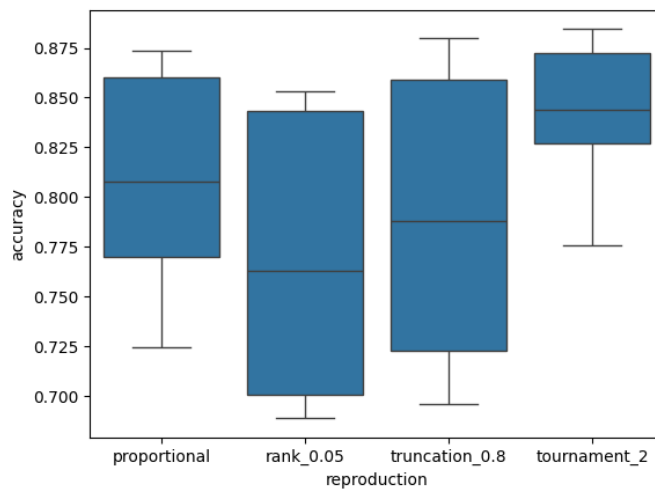
Wpływ hiperparametru reproduction
na zbiorze breast_cancer_wisconsin_diagnostic



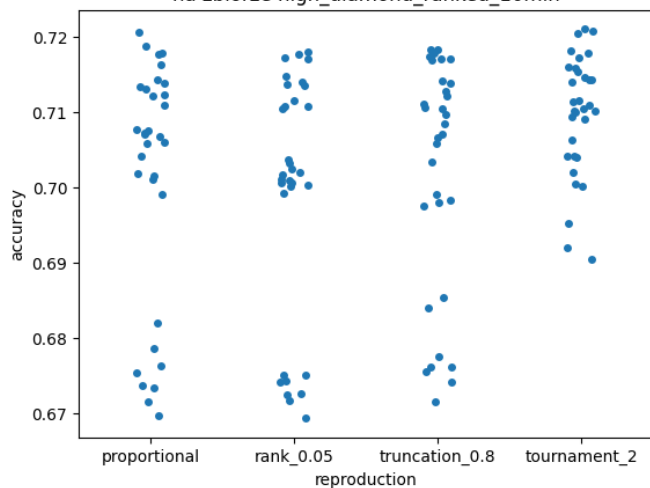
Wpływ hiperparametru reproduction na zbiorze wine



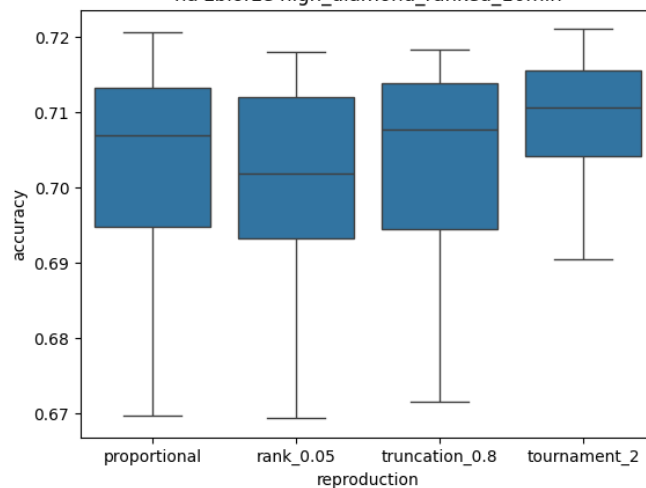
Wpływ hiperparametru reproduction na zbiorze wine



Wpływ hiperparametru reproduction na zbiorze high_diamond_ranked_10min

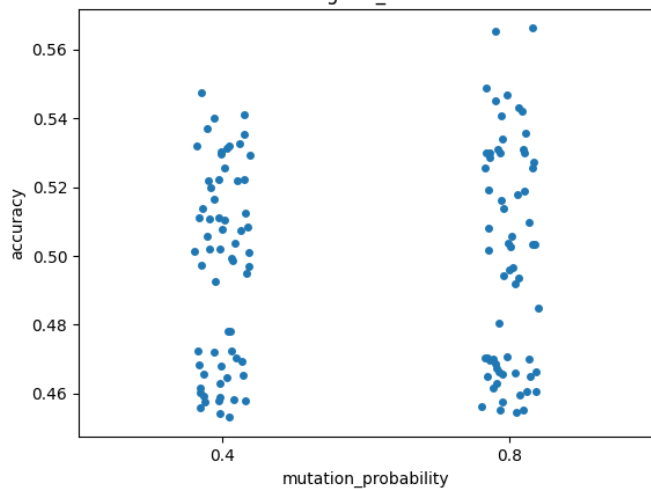


Wpływ hiperparametru reproduction na zbiorze high_diamond_ranked_10min

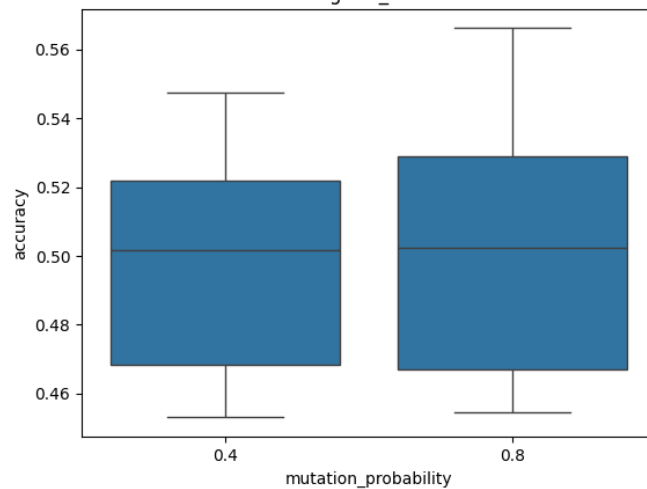


Wpływ hiperparametru mutation_probability na działanie algorytmu

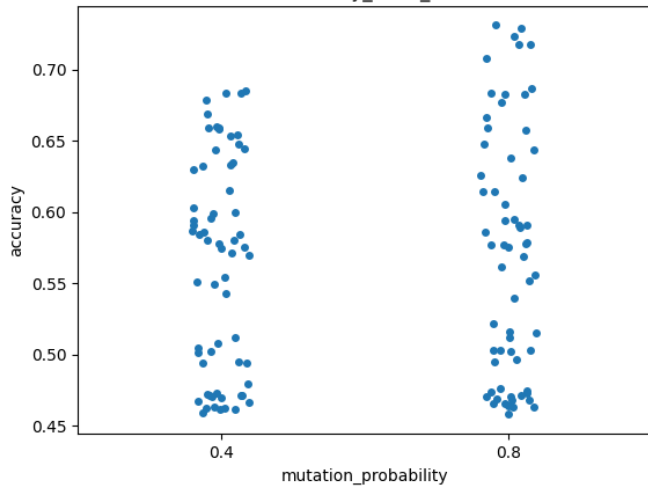
Wpływ hiperparametru mutation_probability na zbiorze glass_identification



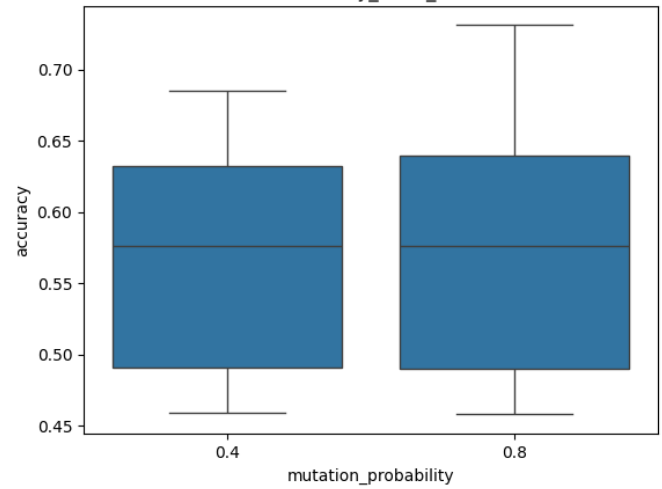
Wpływ hiperparametru mutation_probability na zbiorze glass_identification



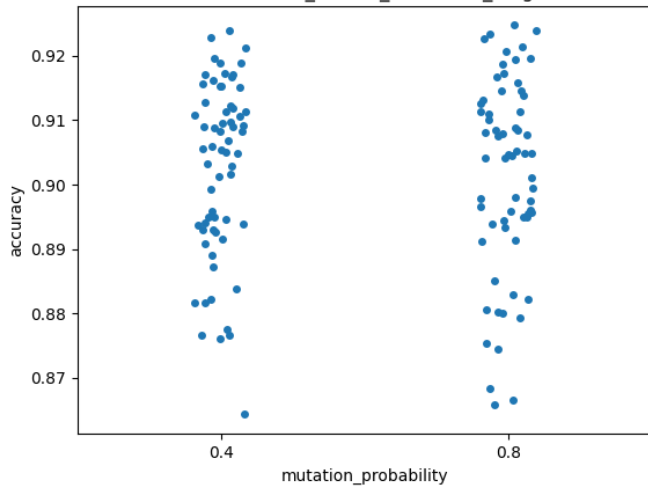
Wpływ hiperparametru mutation_probability
na zbiorze dry_bean_dataset



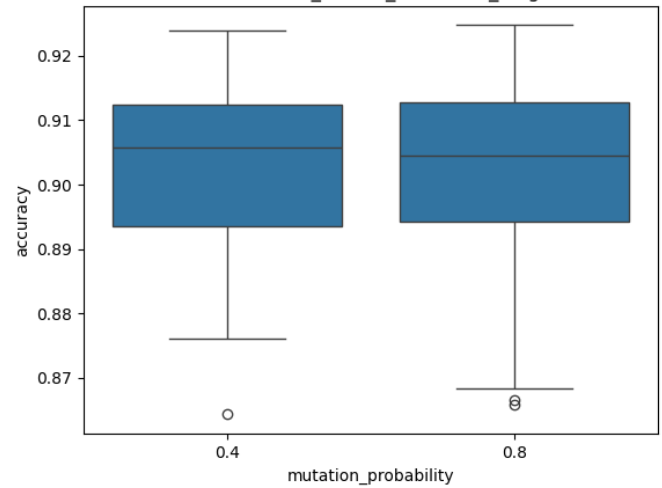
Wpływ hiperparametru mutation_probability
na zbiorze dry_bean_dataset



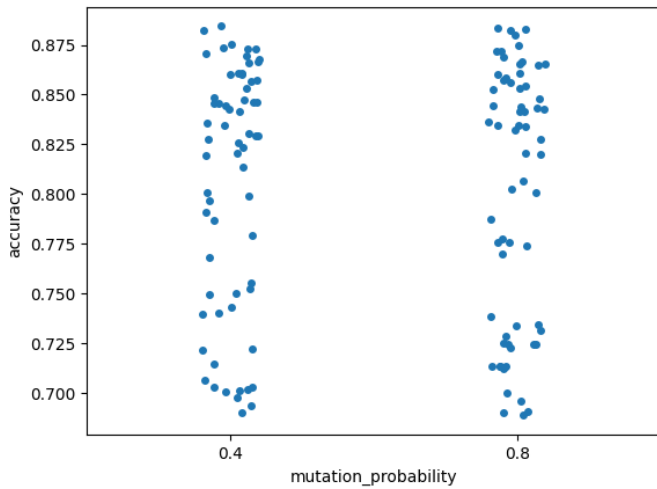
Wpływ hiperparametru mutation_probability
na zbiorze breast_cancer_wisconsin_diagnostic



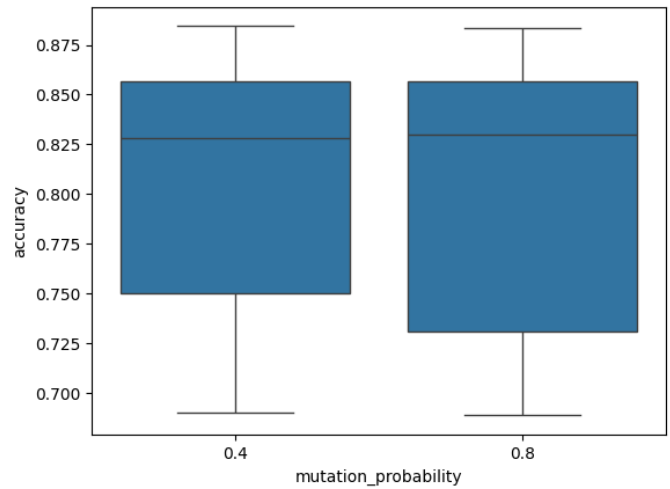
Wpływ hiperparametru mutation_probability
na zbiorze breast_cancer_wisconsin_diagnostic

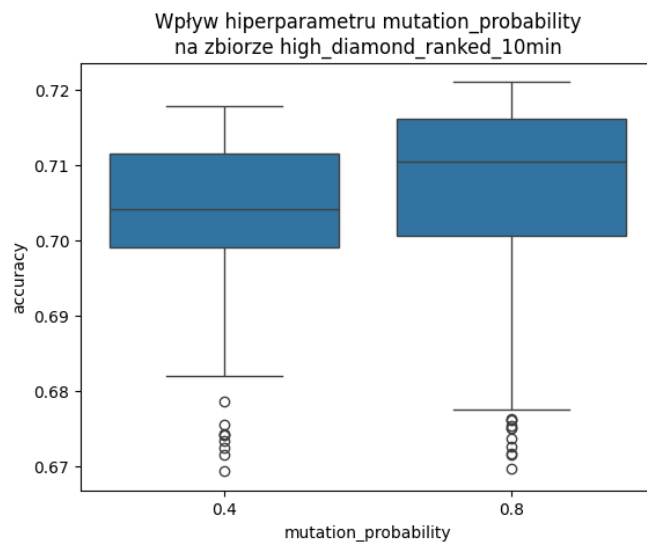
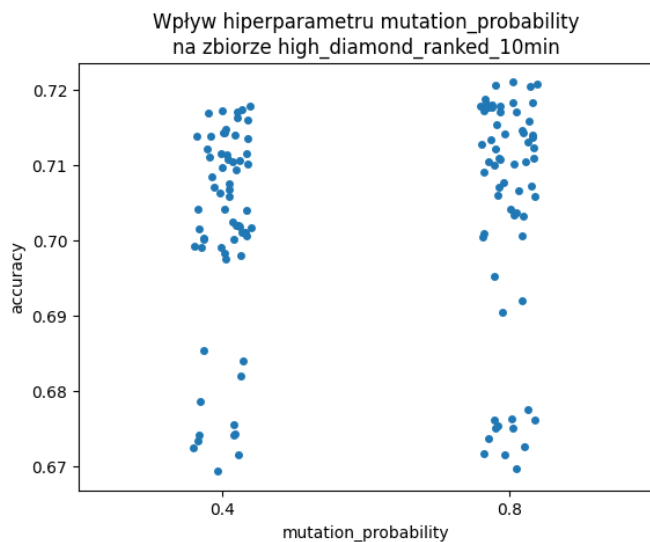


Wpływ hiperparametru mutation_probability
na zbiorze wine



Wpływ hiperparametru mutation_probability
na zbiorze wine

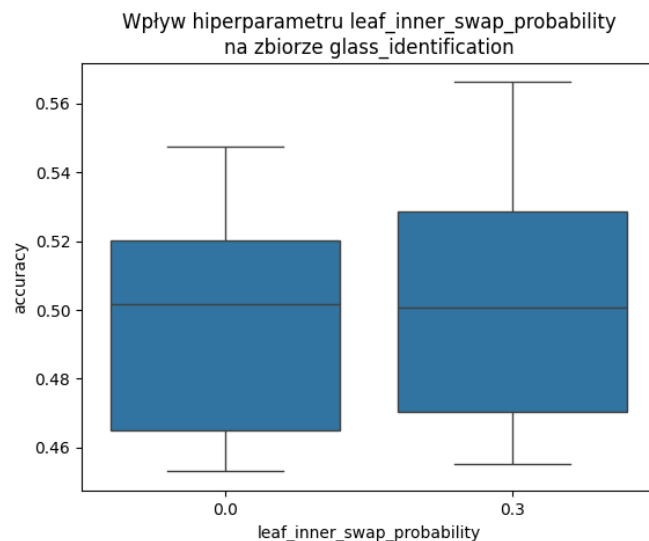
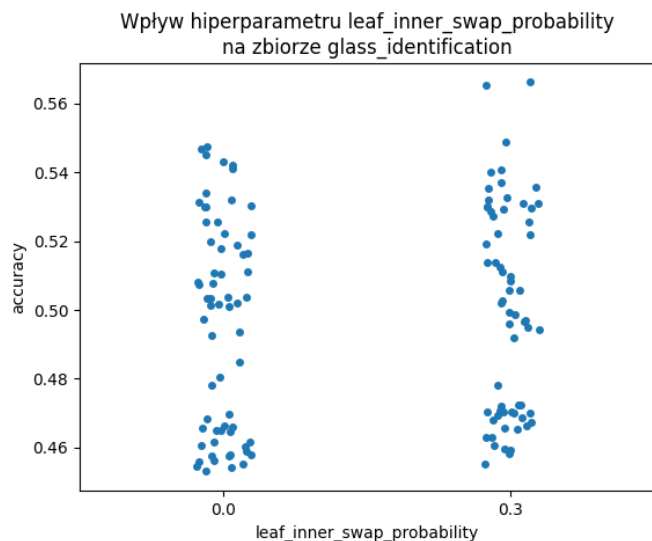




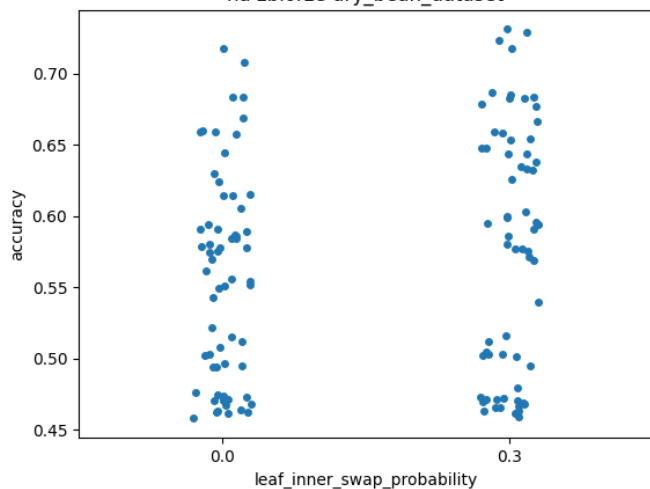
5-krotnej walidacji krzyżowej (używany był zbiór bez zbioru testowego) najlepszej kombinacji hiperparametrów dla tego zbioru danych.

p.mutacji	średnia	odchylenie standardowe	dokładność minimalna	dokładność maksymalna
0,0	0,50	0,10	0,29	0,68
0,2	0,60	0,07	0,50	0,76
0,4	0,64	0,10	0,49	0,85
0,6	0,59	0,10	0,41	0,85
0,8	0,60	0,10	0,38	0,79
1,0	0,60	0,11	0,44	0,82

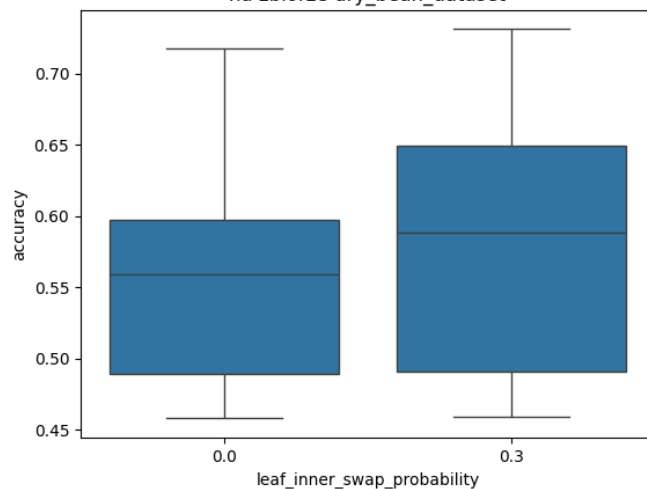
Wpływ hiperparametru leaf_inner_swap_probability na działanie algorytmu



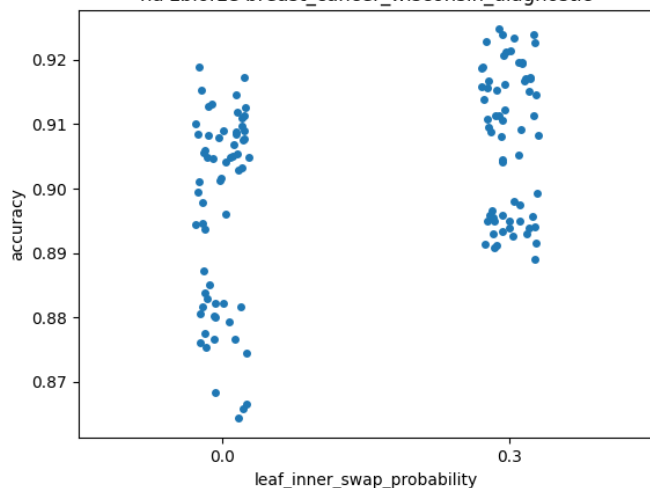
Wpływ hiperparametru leaf_inner_swap_probability
na zbiorze dry_bean_dataset



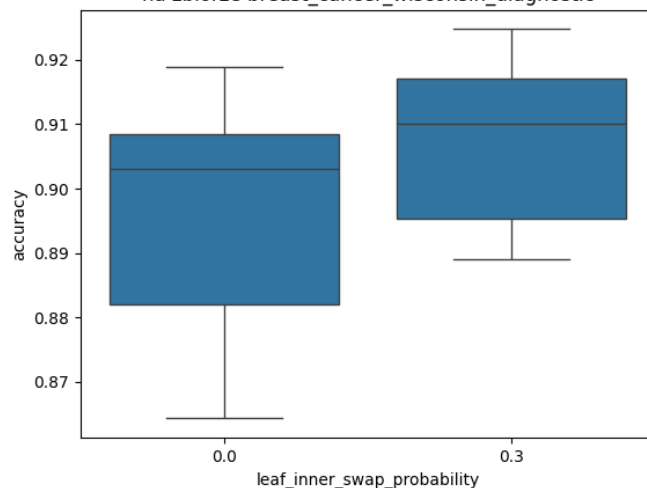
Wpływ hiperparametru leaf_inner_swap_probability
na zbiorze dry_bean_dataset



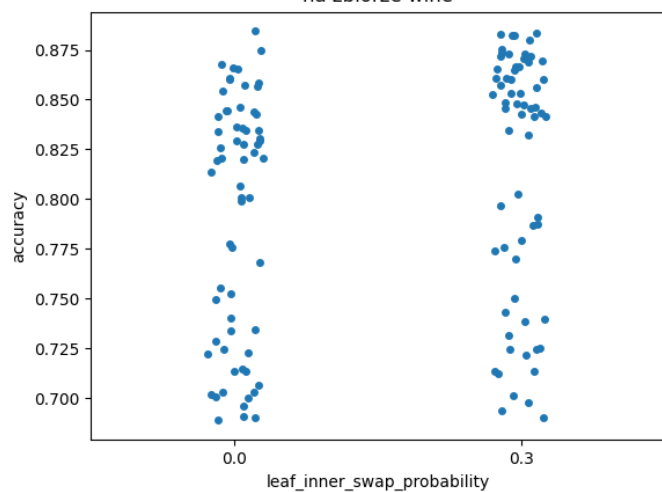
Wpływ hiperparametru leaf_inner_swap_probability
na zbiorze breast_cancer_wisconsin_diagnostic



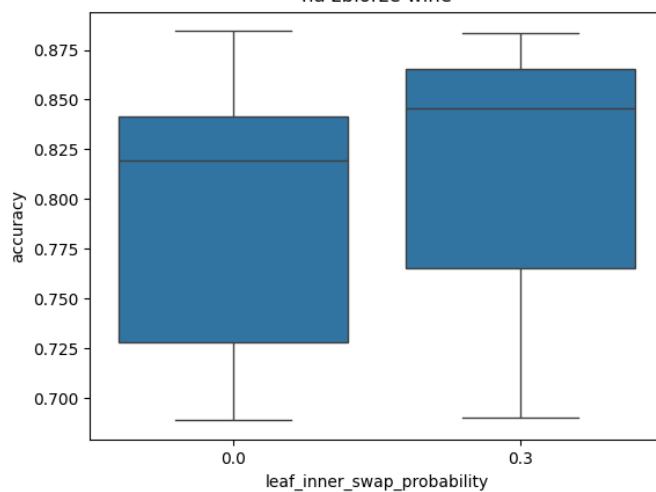
Wpływ hiperparametru leaf_inner_swap_probability
na zbiorze breast_cancer_wisconsin_diagnostic

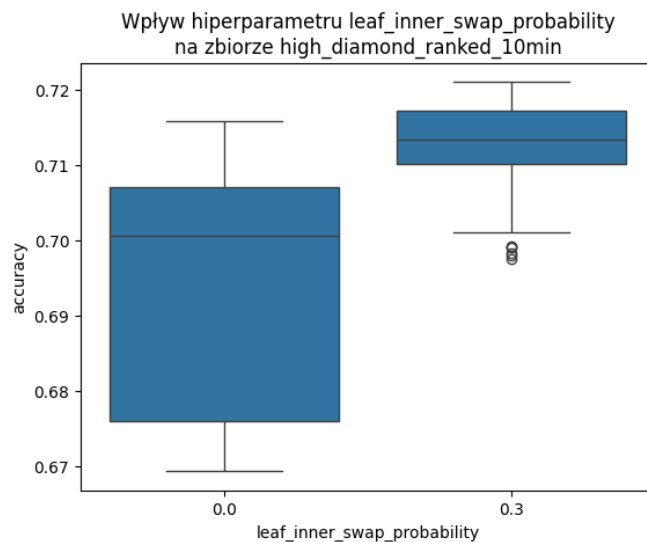
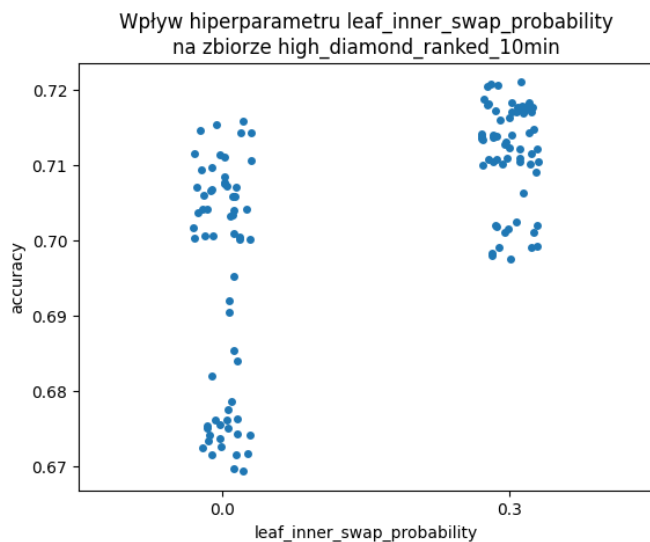


Wpływ hiperparametru leaf_inner_swap_probability
na zbiorze wine



Wpływ hiperparametru leaf_inner_swap_probability
na zbiorze wine





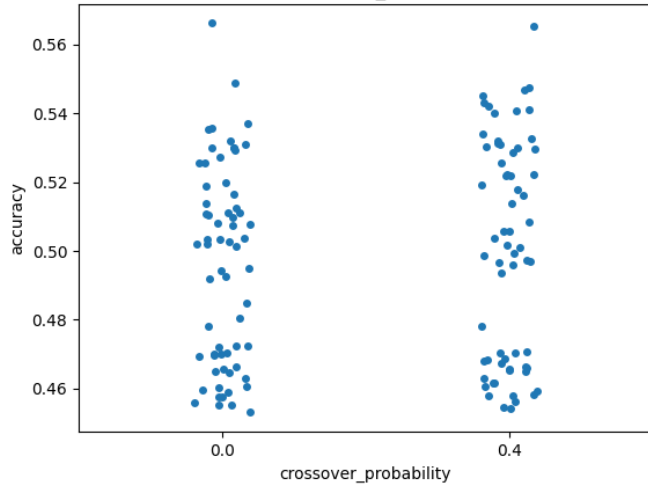
5-krotnej walidacji krzyżowej (używany był zbiór bez zbioru testowego) najlepszej kombinacji hiperparametrów dla tego zbioru danych.

p.zamiany	średnia	odchylenie standardowe	dokładność minimalna	dokładność maksymalna
0,00	0,92	0,05	0,82	1,00
0,15	0,90	0,06	0,76	1,00
0,30	0,93	0,03	0,86	1,00
0,45	0,91	0,05	0,82	1,00
0,60	0,91	0,05	0,79	1,00
0,75	0,90	0,06	0,75	1,00
0,90	0,90	0,07	0,69	1,00

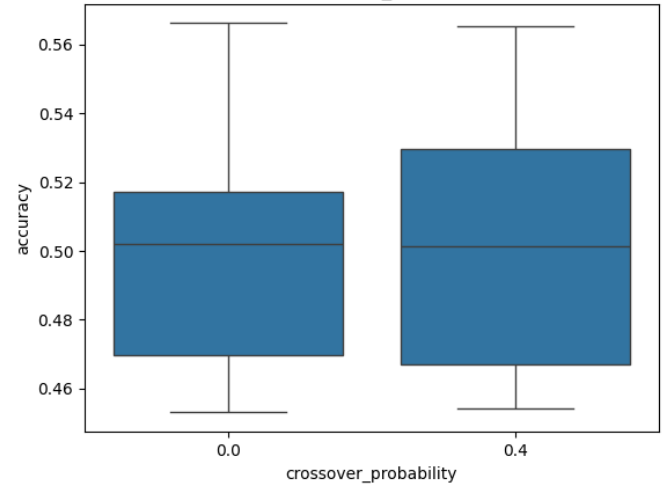
p.zamiany	średnia	odchylenie standardowe	dokładność minimalna	dokładność maksymalna
0,00	0,86	0,07	0,71	0,96
0,15	0,91	0,05	0,79	1,00
0,30	0,91	0,05	0,76	1,00
0,45	0,89	0,05	0,76	0,97
0,60	0,91	0,07	0,72	1,00
0,75	0,90	0,05	0,79	1,00
0,90	0,88	0,05	0,79	1,00

Wpływ hiperparametru crossover_probability na działanie algorytmu

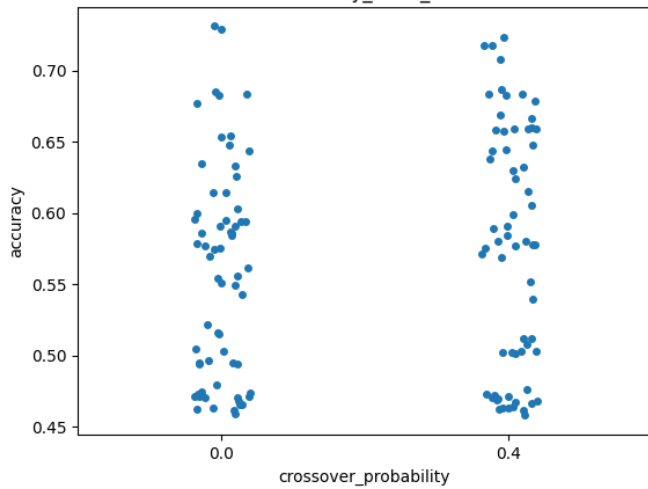
Wpływ hiperparametru crossover_probability na zbiorze glass_identification



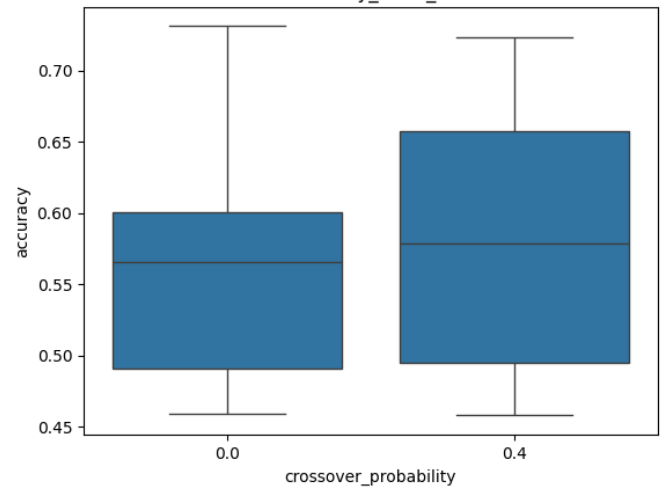
Wpływ hiperparametru crossover_probability na zbiorze glass_identification



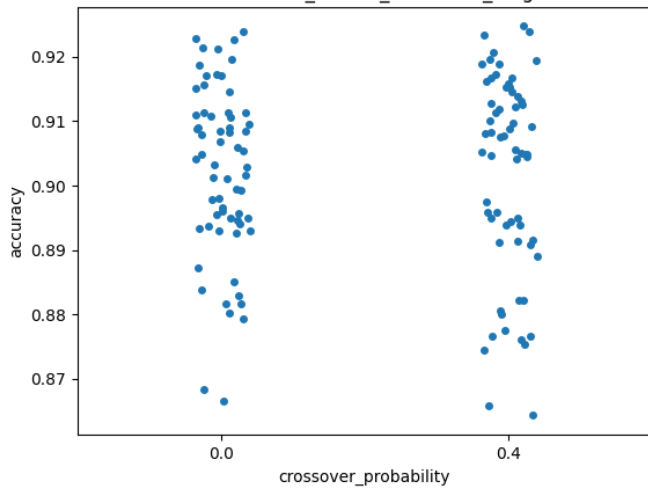
Wpływ hiperparametru crossover_probability na zbiorze dry_bean_dataset



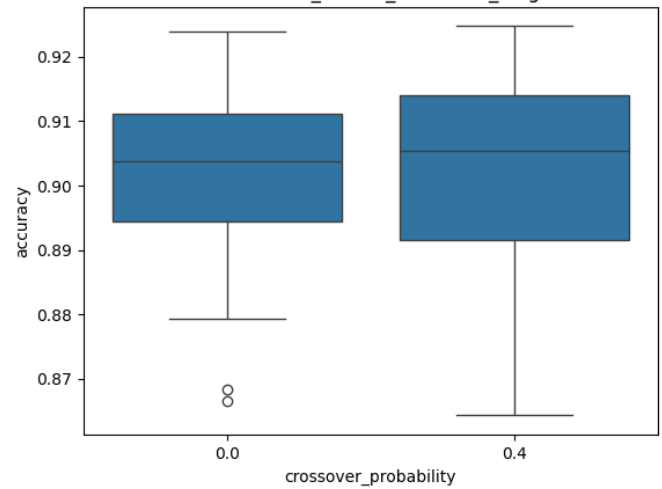
Wpływ hiperparametru crossover_probability na zbiorze dry_bean_dataset



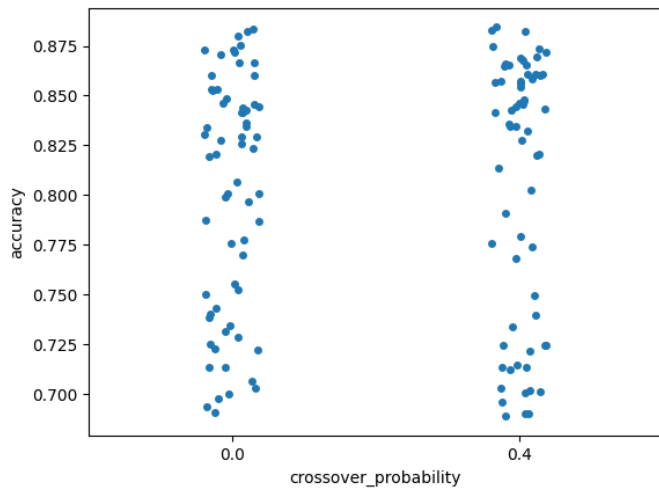
Wpływ hiperparametru crossover_probability na zbiorze breast_cancer_wisconsin_diagnostic



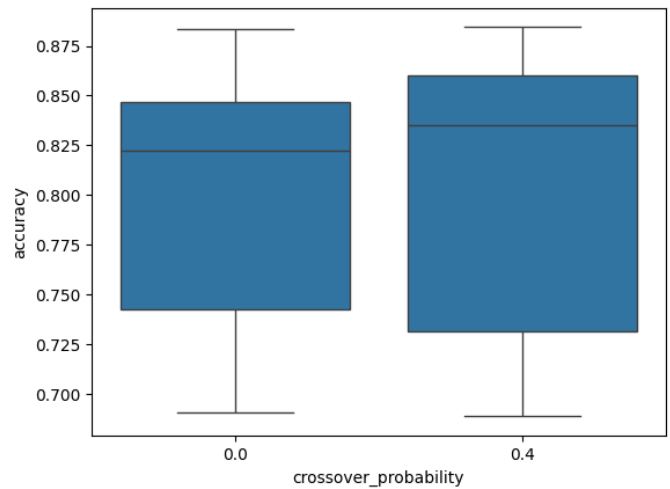
Wpływ hiperparametru crossover_probability na zbiorze breast_cancer_wisconsin_diagnostic



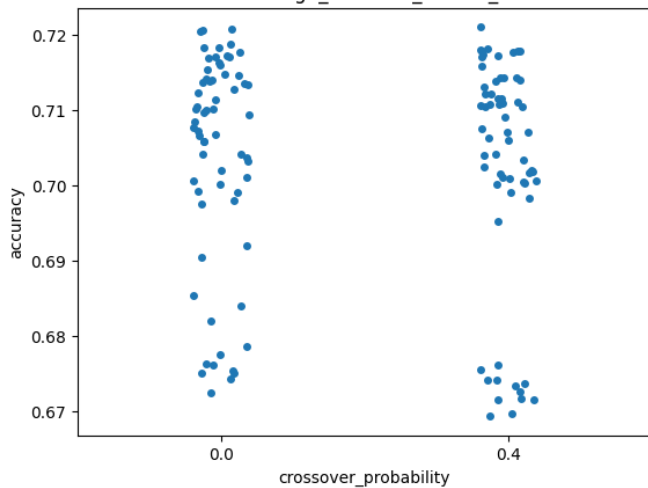
Wpływ hiperparametru crossover_probability
na zbiorze wine



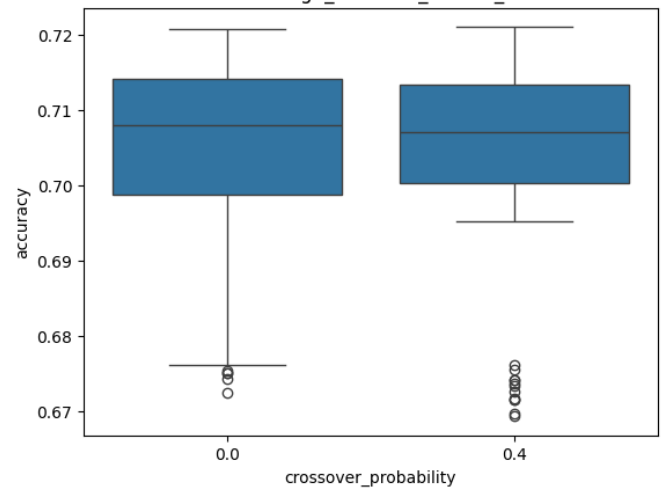
Wpływ hiperparametru crossover_probability
na zbiorze wine



Wpływ hiperparametru crossover_probability
na zbiorze high_diamond_ranked_10min

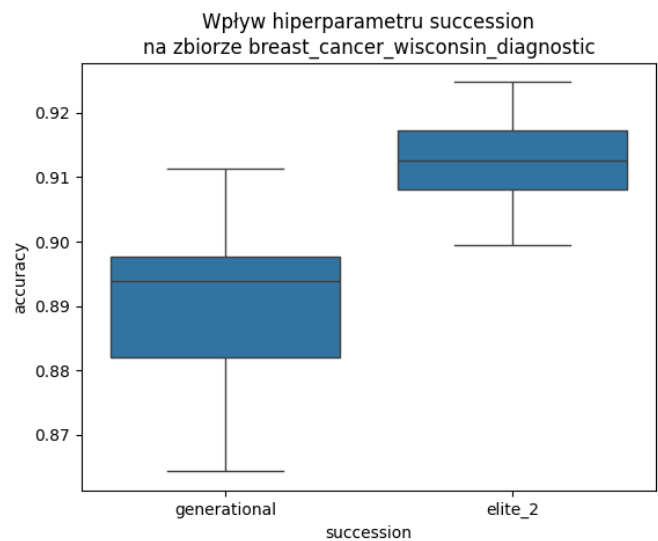
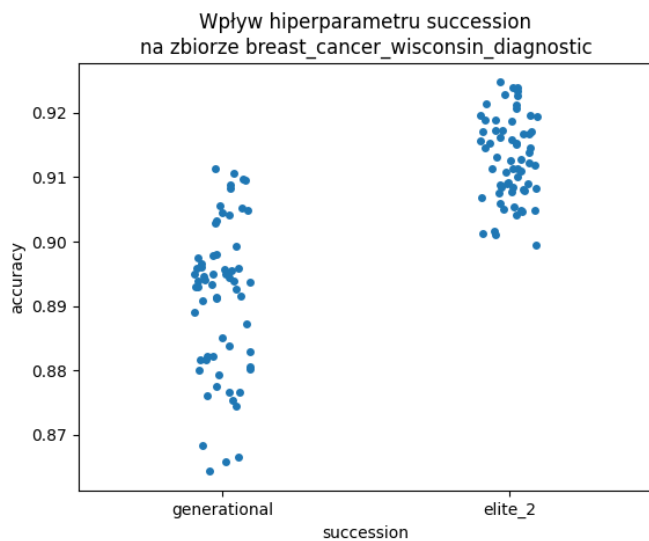
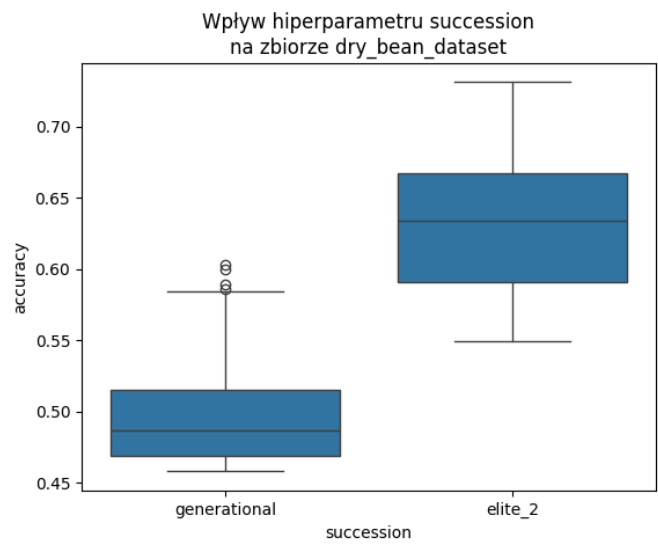
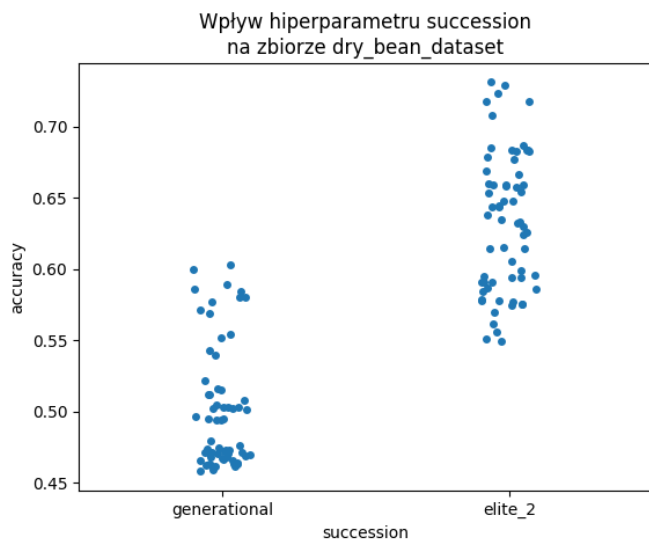
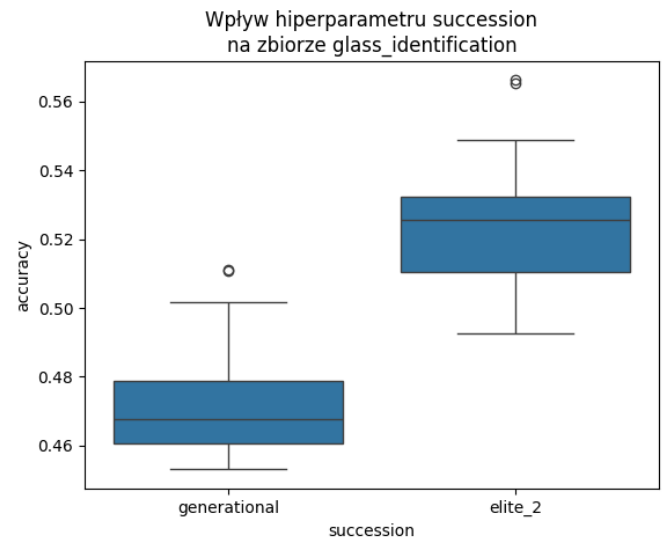
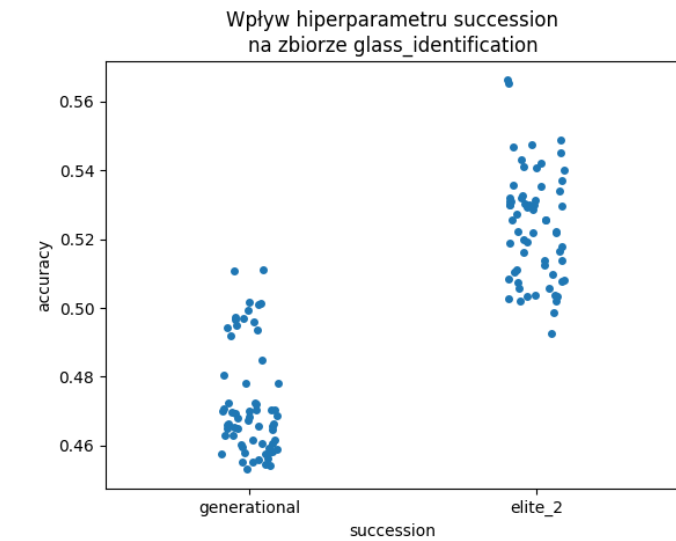


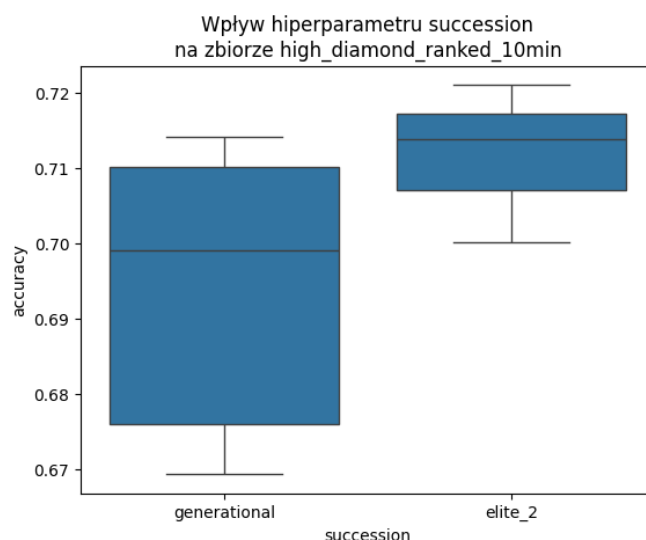
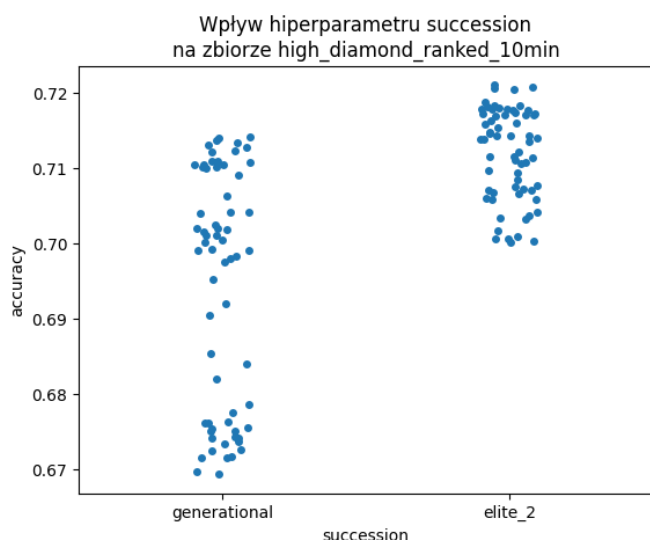
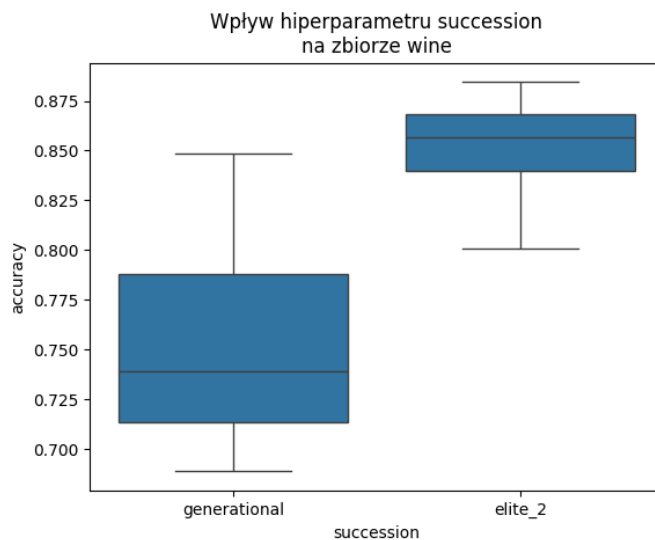
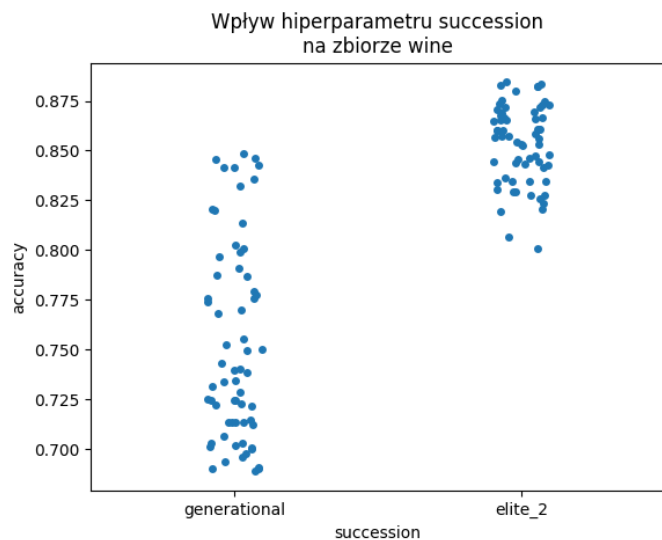
Wpływ hiperparametru crossover_probability
na zbiorze high_diamond_ranked_10min



p.zamiany	średnia	odchylenie standardowe	dokładność minimalna	dokładność maksymalna
0,0	0,57	0,09	0,41	0,82
0,2	0,60	0,10	0,43	0,82
0,4	0,64	0,10	0,49	0,85
0,6	0,64	0,11	0,49	0,88
0,8	0,62	0,09	0,49	0,79
1,0	0,59	0,10	0,46	0,82

Wpływ hiperparametru succession na działanie algorytmu





Elitarna sukcesja zdecydowanie polepszyła wyniki algorytmu w stosunku do sukcesji generacyjnej. Wynika to z tego, że mutacja zmienia drzewa dość mocno - nawet mutacja bez zamiany liścia w węzeł wewnętrzny lub odwrotnie może bardzo popsuć jakość drzewa, np. zmieniając test w korzeniu. W związku z tym, zachowanie najlepszych osobników w populacji jest konieczne - możliwe, że dopiero po wielu iteracjach uzyskana zostanie poprawa, szybka strata dobrego osobnika uniemożliwi jakąkolwiek eksploatację jego okolic.

Najlepsze zestawy hiperparametrów

Wyniki w poniższej tabeli to dokładności na zbiorze walidacyjnym, zagregowane z najlepszych (o najlepszej średniej dokładności) zestawów hiperparametrów dla każdego ze zbiorów danych.

z.danych	max_depth	reprodukcja	p.mut.	p.zamiany	p.krzyż.	sukcesja	średnia	odch.stan.	najgorszy	najlepszy
cancer	5	proporcjonalna	0,4	0,3	0,4	elitarna	0,94	0,02	0,88	0,98
bean	20	rangowa	0,8	0,3	0,0	elitarna	0,89	0,01	0,88	0,90
glass	20	turniejowa	0,4	0,0	0,4	elitarna	0,64	0,10	0,49	0,85
wine	5	rangowa	0,4	0,3	0,4	elitarna	0,93	0,03	0,86	1,00
games	20	proporcjonalna	0,4	0,0	0,4	elitarna	0,73	0,01	0,71	0,75

Nadmierne dopasowanie

W celu ustalenia, czy wystąpiło nadmierne dopasowanie, wyliczyliśmy wartości dokładności dla algorytmu ewolucji drzewa dla zbiorów treningowego (tego, który wcześniej był używany do walidacji krzyżowej) i testowego.

Poniższe tabele przedstawiają wyniki (dokładności) naszego algorytmu ewolucji drzewa na zbiorach treningowym i testowym, zagregowane z uruchomień algorytmu.

Dla zbioru danych **glass_identification**:

zbiór danych	średnia dokładność	odchylenie standardowe	dokładność minimalna	dokładność maksymalna
treningowy	0,78	0,04	0,70	0,83
testowy	0,74	0,04	0,67	0,84

Dla zbioru danych **dry_bean_dataset**:

zbiór danych	średnia dokładność	odchylenie standardowe	dokładność minimalna	dokładność maksymalna
treningowy	0,89	0,01	0,88	0,90
testowy	0,89	0,01	0,88	0,90

Dla zbioru danych **breast_cancer_wisconsin_diagnostic**:

zbiór danych	średnia dokładność	odchylenie standardowe	dokładność minimalna	dokładność maksymalna
treningowy	0,95	0,01	0,93	0,97
testowy	0,94	0,02	0,89	0,96

Dla zbioru danych **wine**:

zbiór danych	średnia dokładność	odchylenie standardowe	dokładność minimalna	dokładność maksymalna
treningowy	0,98	0,01	0,95	1,00
testowy	0,93	0,05	0,83	1,00

Dla zbioru danych **high_diamond_ranked_10min**:

zbiór danych	średnia dokładność	odchylenie standardowe	dokładność minimalna	dokładność maksymalna
treningowy	0,73	0,00	0,72	0,74
testowy	0,72	0,01	0,71	0,74

Porównanie z algorytmem ID3

Skrypt wykonujący poniższy eksperyment jest zawarty w pliku `comparison_experiment.py`.

Wykonaliśmy analogiczne przeszukiwanie po hipersiatce hiperparametrów dla metody referencyjnej - algorytmu ID3. Sprawdzane były następujące wartości hiperparametrów:

- **max_depth** - maksymalna głębokość drzewa w populacji - wartości 5 oraz 20;
- **min_samples_split** - minimalna liczba przykładów w węźle niezbędna, aby rozważany był dalszy jego podział - wartości 1, 2 i 10;
- **prune** - czy wykonywane będzie przycinanie drzewa - wartości *True* oraz *False*;
- **gain_ratio** - czy używane będzie **gain ratio** zamiast **information gain** do wyboru podziału węzła - wartości *True* oraz *False*;
- **min_entropy_decrease** - minimalna wartość entropii, przy której wykonywany jest dalszy podział węzła - wartości 0 oraz 0,3;
- **repeating** - czy będą wykonywane wielokrotne podziały po tym samym atrybucie - wartości *True* oraz *False*.

Najlepsze (o najlepszej średniej dokładności) kombinacje hiperparametrów dla poszczególnych zbiorów danych są przedstawione w poniższej tabeli.

Wyniki w poniższej tabeli to dokładności na zbiorze walidacyjnym, zagregowane z

z.danych	max_depth	min_samp	prune	gain_ratio	min_e_dec	repeating	średnia	odch.stan.	najgorszy	najlepszy
cancer	5	1	<i>False</i>	<i>False</i>	0	<i>True</i>	0,95	0,03	0,91	0,98
bean	20	10	<i>True</i>	<i>False</i>	0,3	<i>False</i>	0,90	0,01	0,89	0,91
glass	20	1	<i>False</i>	<i>False</i>	0	<i>True</i>	0,64	0,03	0,60	0,68
wine	5	1	<i>False</i>	<i>False</i>	0	<i>False</i>	0,89	0,05	0,82	0,97
games	5	10	<i>False</i>	<i>True</i>	0	<i>True</i>	0,72	0,01	0,71	0,74

Dla najlepszych hiperparametrów przedstawionych w dwóch powyższych tabelach, uruchomiliśmy algorytmy ponownie. Jako zbiór treningowy został użyty cały zbiór używany wcześniej do walidacji krzyżowej. Ewaluacja została wykonana na wydzielonym uprzednio zbiorze testowym.

Poniższe tabele przedstawiają wyniki (dokładności) naszego algorytmu ewolucji drzewa i algorytmu ID3 na zbiorze testowym, zagregowane z uruchomień algorytmu.

Dla zbioru danych **glass_identification**:

algorytm	średnia dokładność	odchylenie standardowe	dokładność minimalna	dokładność maksymalna
ewolucja drzewa	0,74	0,04	0,67	0,84
ID3	0,74	0,00	0,74	0,74

Dla zbioru danych **dry_bean_dataset**:

algorytm	średnia dokładność	odchylenie standardowe	dokładność minimalna	dokładność maksymalna
ewolucja drzewa	0,89	0,01	0,88	0,90
ID3	0,91	0,00	0,90	0,91

Dla zbioru danych **breast_cancer_wisconsin_diagnostic**:

algorytm	średnia dokładność	odchylenie standardowe	dokładność minimalna	dokładność maksymalna
ewolucja drzewa	0,94	0,02	0,89	0,96
ID3	0,94	0,00	0,94	0,94

Dla zbioru danych **wine**:

algorytm	średnia dokładność	odchylenie standardowe	dokładność minimalna	dokładność maksymalna
ewolucja drzewa	0,93	0,05	0,83	1,00
ID3	0,92	0,00	0,92	0,92

Dla zbioru danych **high_diamond_ranked_10min**:

algorytm	średnia dokładność	odchylenie standardowe	dokładność minimalna	dokładność maksymalna
ewolucja drzewa	0,72	0,01	0,71	0,74
ID3	0,72	0,00	0,72	0,72

Algorytm ewolucji drzewa jest przeszukiwaniem w dużej mierze polegającym na losowości - widać to po zdecydowanie większych odchyleniach standardowych wyników.

Poniższe tabele przedstawiają wyniki (dokładności) algorytmu ID3 na zbiorach treningowym i testowym, zagregowane z

Dla zbioru danych **glass_identification**:

zbiór danych	średnia dokładność	odchylenie standardowe	dokładność minimalna	dokładność maksymalna
treningowy	1,00	0,00	1,00	1,00
testowy	0,74	0,00	0,74	0,74

Dla zbioru danych `dry_bean_dataset`:

zbiór danych	średnia dokładność	odchylenie standardowe	dokładność minimalna	dokładność maksymalna
treningowy	0,93	0,00	0,92	0,94
testowy	0,91	0,00	0,90	0,91

Dla zbioru danych `breast_cancer_wisconsin_diagnostic`:

zbiór danych	średnia dokładność	odchylenie standardowe	dokładność minimalna	dokładność maksymalna
treningowy	0,99	0,00	0,99	0,99
testowy	0,94	0,00	0,94	0,94

Dla zbioru danych `wine`:

zbiór danych	średnia dokładność	odchylenie standardowe	dokładność minimalna	dokładność maksymalna
treningowy	1,00	0,00	1,00	1,00
testowy	0,92	0,00	0,92	0,92

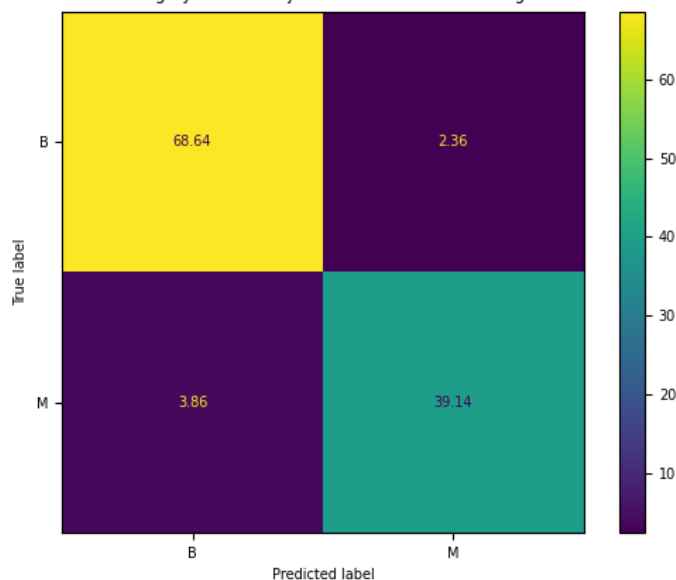
Dla zbioru danych `high_diamond_ranked_10min`:

zbiór danych	średnia dokładność	odchylenie standardowe	dokładność minimalna	dokładność maksymalna
treningowy	0,73	0,00	0,73	0,73
testowy	0,72	0,00	0,72	0,72

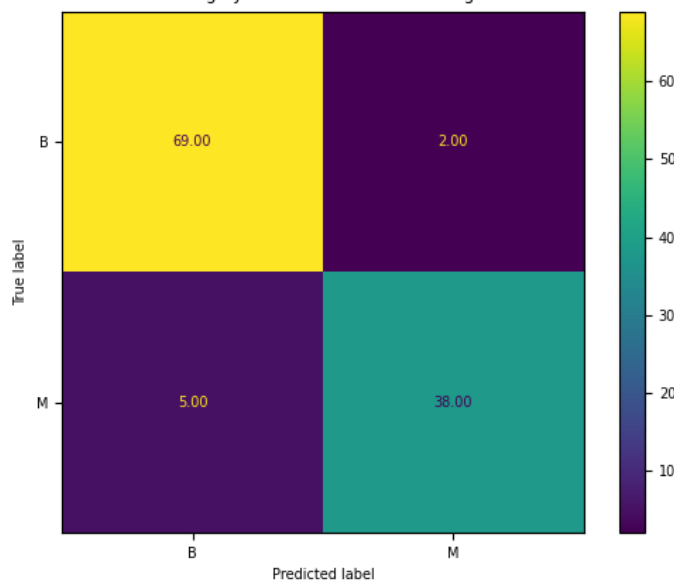
Algorytm ID3 osiągnął znacząco lepsze wyniki na zbiorze treningowym niż na testowym w przypadku mniej podatny na nadmierne dopasowanie niż algorytm ID3.

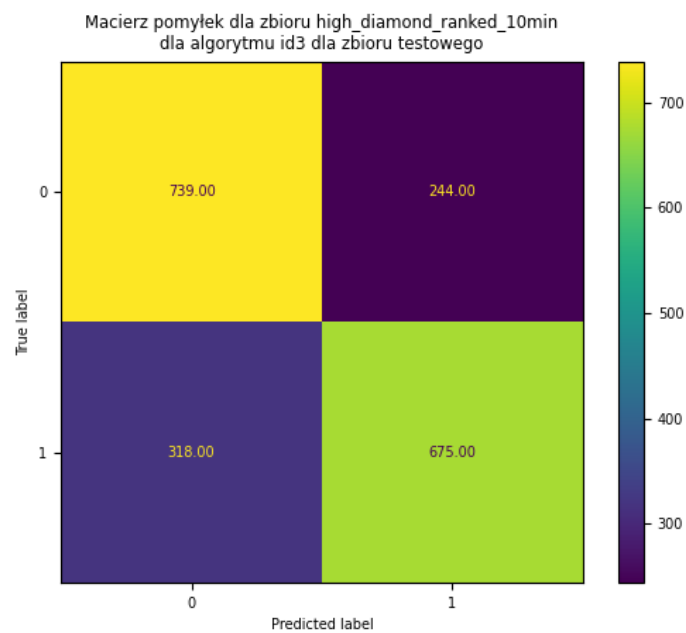
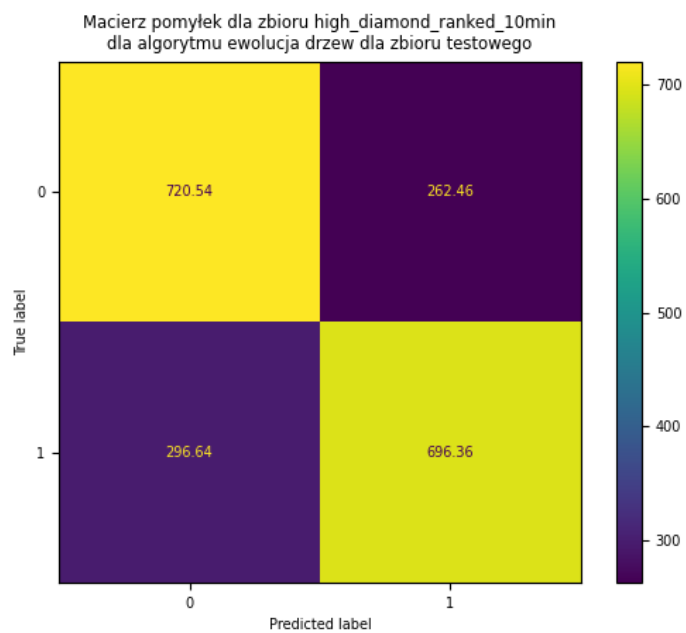
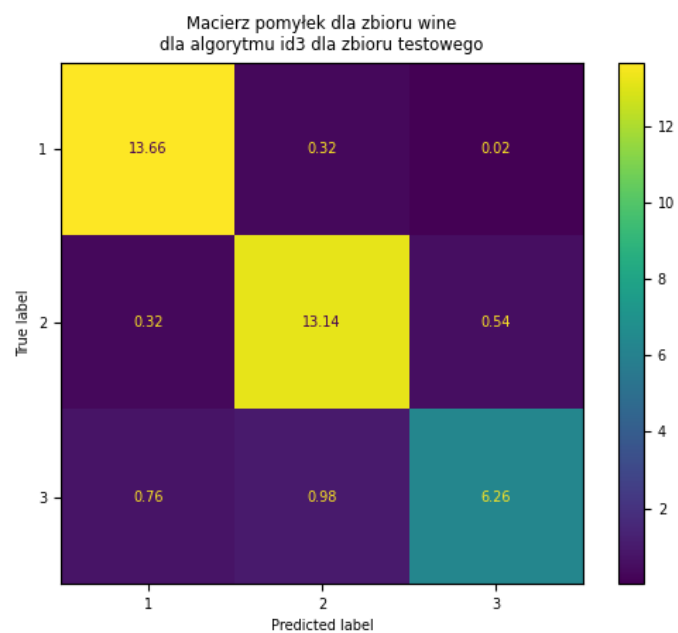
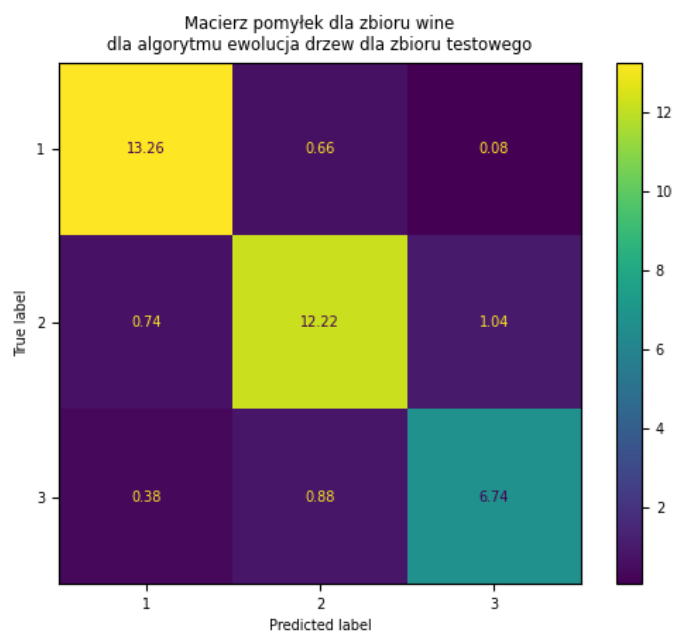
Poniżej przedstawione są macierze pomyłek (wyniki na zbiorze testowym, uśrednione dla dla poszczególnych zbiorów danych).

Macierz pomyłek dla zbioru `breast_cancer_wisconsin_diagnostic` dla algorytmu ewolucja drzew dla zbioru testowego

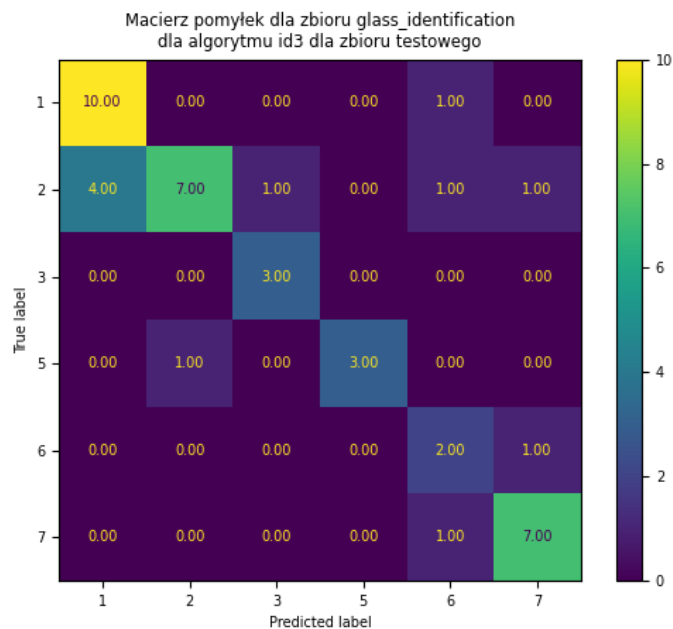
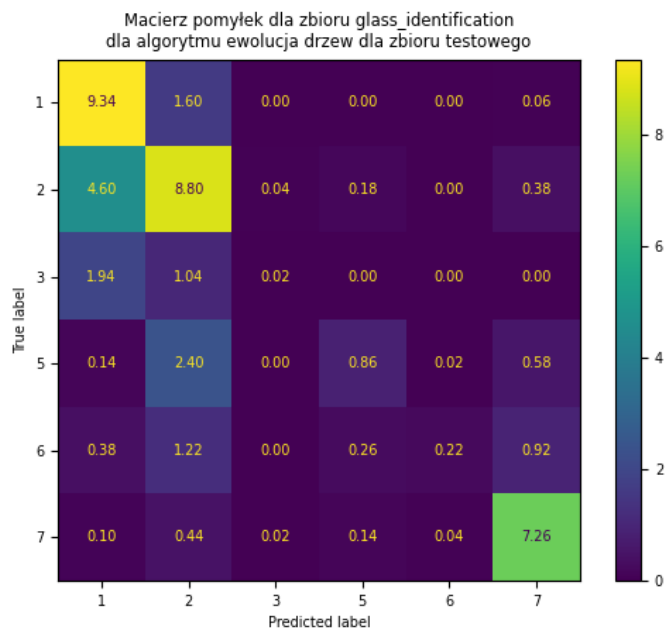
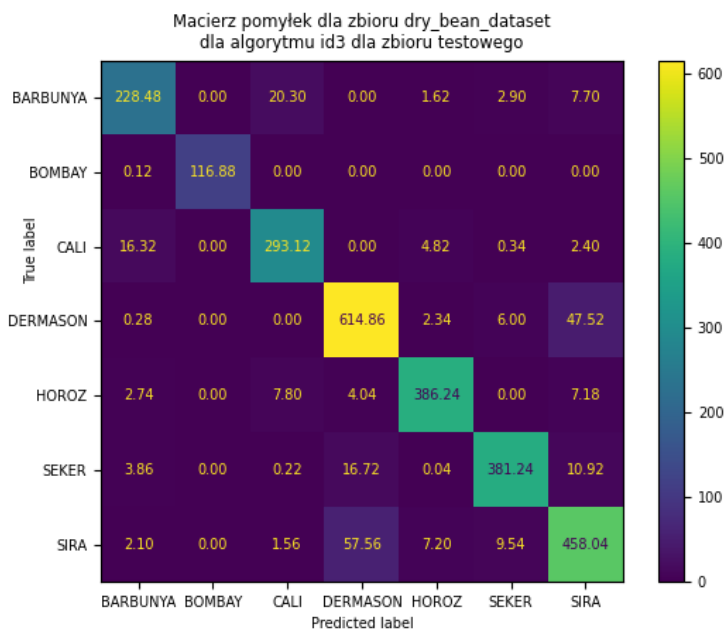
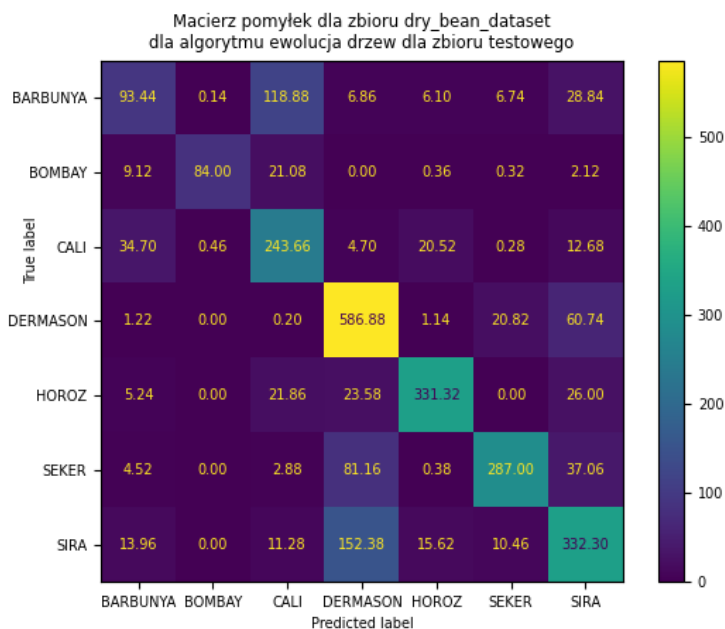


Macierz pomyłek dla zbioru `breast_cancer_wisconsin_diagnostic` dla algorytmu id3 dla zbioru testowego





Macierze pomyłek dla zbiorów danych z mniejszą liczbą klas nie wykazują żadnych anomalii, ani znaczących różnic między obydwoma algorytmami.



Dla zbiorów danych z większą liczbą klas, algorytm ID3 nie ignoruje żadnej z klas - nawet klasy mające bardzo małą liczbę przykładów są w większości poprawnie klasyfikowane. Algorytm ewolucji drzewa

Macierze pomyłek ponownie pokazują to, co wynika również z tabel -

5. Końcowe wnioski

Implementacja algorytmu ewolucji drzewa jest poprawna; osiągnął on wyniki porównywalne z metodą referencyjną.

Algorytm ewolucji drzewa nadaje się do zadań klasyfikacji,

Algorytm ewolucji drzewa nie jest szczególnie podatny na nadmierne dopasowanie.

W porównaniu z algorytmem ID3, algorytm ewolucji drzewa średnio osiąga porównywalne wyniki. Jednakże, czas jego wykonania jest znacznie dłuższy, zatem jest on mniej efektywny.

Podczas wykonywania projektu nauczyliśmy się więcej o zachowaniu algorytmów optymalizacyjnych dla różnych parametrów i zbiorów danych. Zyskaliśmy również doświadczenie w efektywnym przygotowywaniu trwających wiele dni obliczeń.