Podstawy Sztucznej Inteligencji
Projekt 2 – Sprawozdanie

Zespół:
Jarosław Zabuski - 300288
Jakub Strawa - 300266

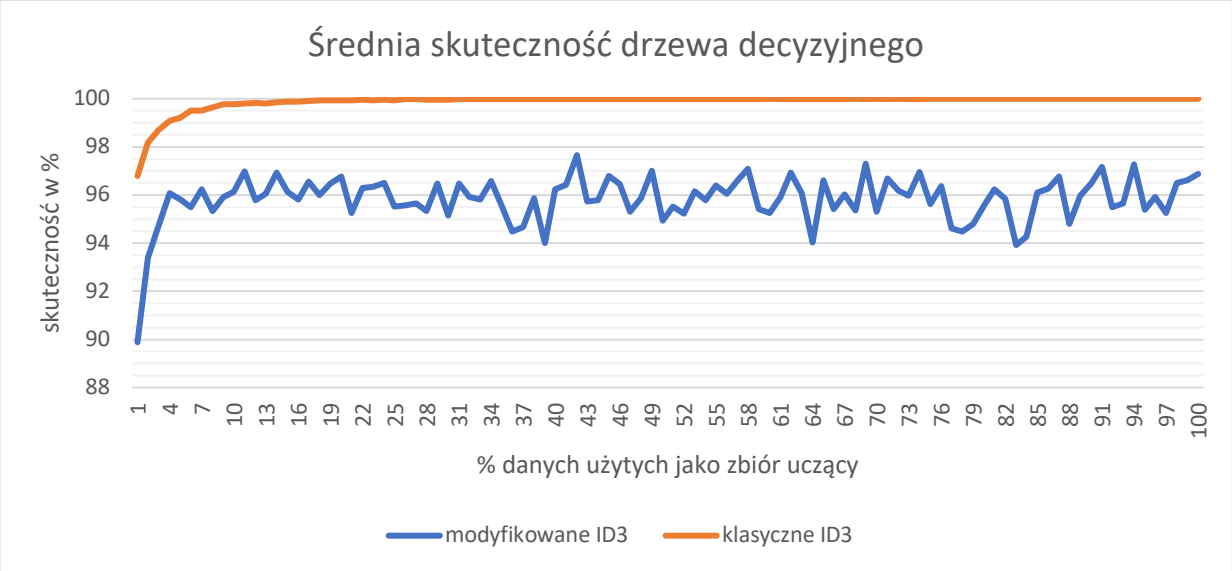
Temat:
Przewidywanie czy grzyb jest jadalny przy użyciu zmodyfikowanej implementacji algorytmu ID3. Do wyboru testu w drzewie stosujemy zasadę koła ruletki - większe prawdopodobieństwo wyboru mają lepsze testy. Porównanie wyników z wynikami wersji klasycznej algorytmu.

Doprecyzowanie tematu i założenia:
Założyliśmy, że w podstawowej wersji algorytmu ID3 dodajemy nowe gałęzie do drzewa decyzyjnego wybierając atrybut z największym zyskiem informacyjnym aż cały zbiór testowy nie zostanie w pełni opisany. Dla zmodyfikowanej wersji ID3 nowe gałęzie są dodawane do momentu aż żaden z pozostałych atrybutów nie niesie ze sobą żadnego zysku informacyjnego. Prawdopodobieństwo wyboru atrybutu jest równe zyskowi informacyjnemu podzielonego przez sumę zysków informacyjnych wszystkich atrybutów które są możliwe do wyboru na tym etapie.

Podział prac:
- Jarosław Zabuski: implementacja klasycznej wersji ID3, opisanie eksperymentów
- Jakub Strawa: implementacja zmodyfikowanej wersji ID3, przeprowadzenie eksperymentów

Wykorzystane narzędzia i biblioteki:
Program został napisany w języku Python. Korzystaliśmy wyłącznie z modułów biblioteki standardowej: math, random, sys, gc, time(w celach testowych) oraz klasy Counter z collections. Do przeprowadzenia testów i eksperymentów wykorzystaliśmy plik z danymi ze strony: https://archive.ics.uci.edu/ml/datasets/mushroom używany w formie rozszerzonej (z pełnymi nazwami atrybutów oraz podpisami kolumn) jako plik tekstowy mushroom.txt. Klasyczna implementacja ID3 została przez nas wprowadzona na podstawie artykułu „An Application of Decision Tree Based on ID3" autorstwa Wang Xiaohu, Wang Lele, oraz Li Nianfeng. https://www.sciencedirect.com/science/article/pii/S1875389212006098

Przeprowadzone eksperymenty:
1. Zmiana skuteczności wraz ze wzrostem zbioru uczącego.
    a. Założenia: Testujemy jaki wpływ na skuteczność działania zmodyfikowanego algorytmu ID3 ma rozmiar danych uczących. Zakładamy, że najprawdopodobniej dość szybko (okolice 5%) obie metody osiągną swoje maksymalne skuteczności i będą się na tym poziomie utrzymywały.

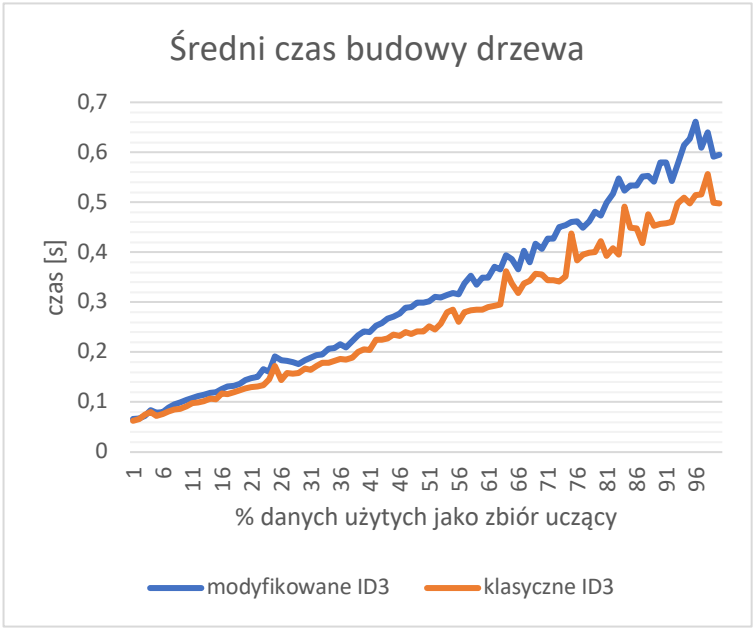


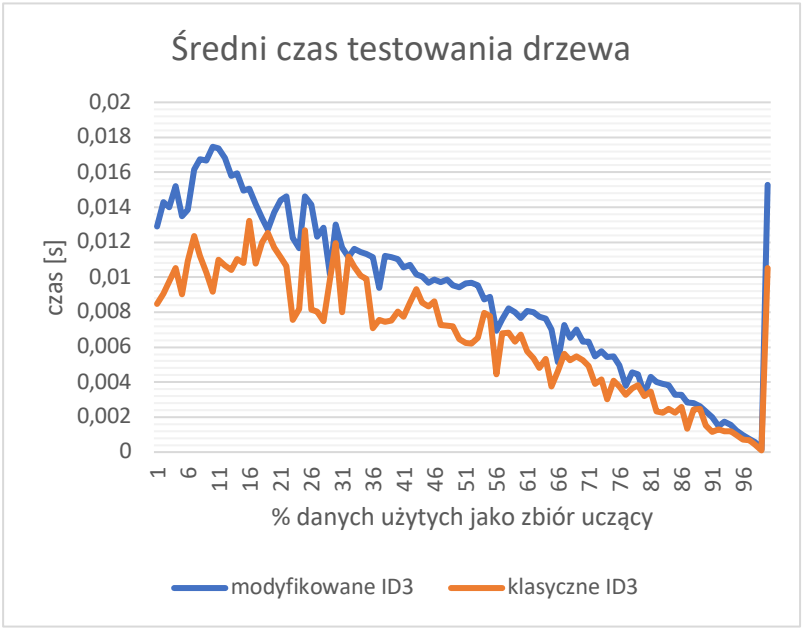
b. Eksperyment: Do każdego eksperymentu nasz program dla każdego procenta z zakresu [1,100] wykonał 100 uruchomień obu algorytmów. Na wykresie znajdują się średnie zgodności wyprodukowanych przez obie implementacje drzew decyzyjnych. Owe średnie zgodności są dla nas główny współczynnikiem badany w tym eksperymencie.

c. Wniosek: Z danych na wykresie wynika, że zmodyfikowana implementacja ID3 jest średnio o 5 p.p. mniej dokładna (oscyluje na poziomie 95%) od klasycznej implementacji w stosunku do klasycznej, a jest to powiązane z wprowadzeniem elementu losowości do zmodyfikowanego ID3. Klasyczna implementacja osiąga lepsze wyniki, bo zawsze sugeruje się ona metodą największego przyrostu informacyjnego.

2. Średni czas wykonywania.
   a. Założenia: Testujemy w jaki sposób wprowadzenie semi-losowego wyboru testów do metody ID3 wpływa na czas jej wykonania. Zakładamy, że średni czas działania będzie dłuższy dla zmodyfikowanej wersji.

   b. Eksperyment: Badamy średni czas poświęcony przez dany algorytm na budowę drzewa i średni czas poświęcony na przeprowadzenie na nim testów. Wykresy te znajdują się poniżej.

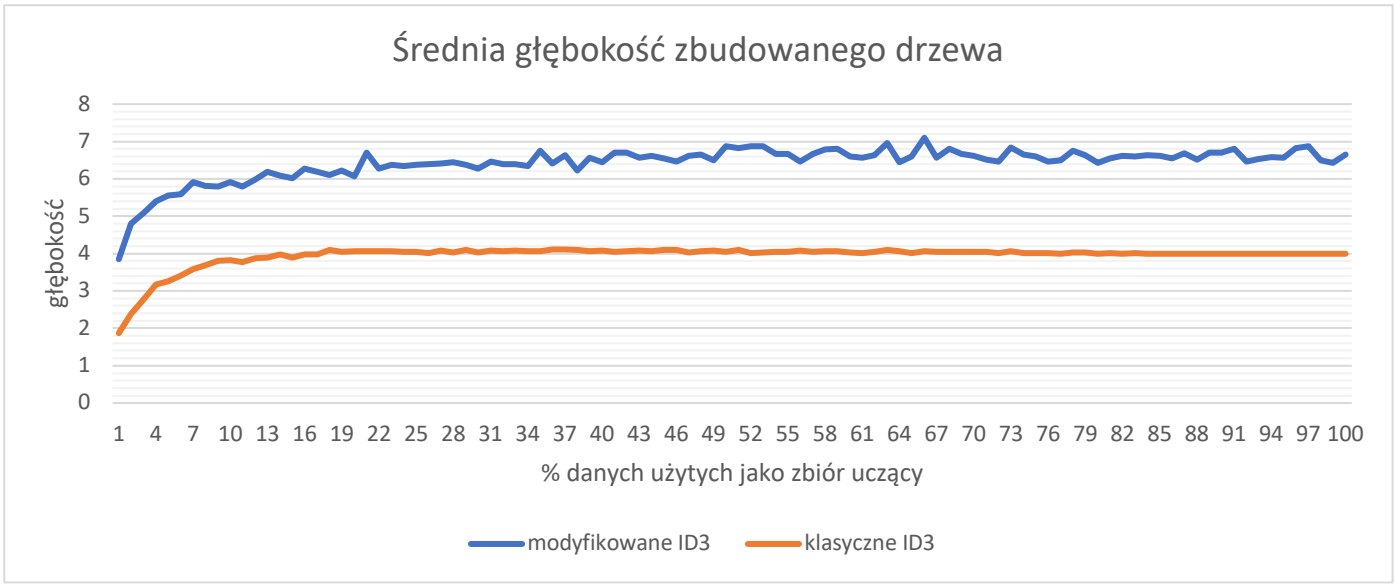




c. Wniosek: Zmodyfikowany ID3 w każdym przypadku okazywał się być bardziej czasochłonny aniżeli zwykły ID3. Związane jest to ze specyfiką wprowadzonej modyfikacji – zmodyfikowane ID3 buduje większe drzewa. Końcowy wystrzał dla wartości 100% związany jest z tym, że w taki przypadku algorytm uczy się i testuje na tym samym zbiorze wszystkich danych.

3. Średnia głębokość drzewa.
   a. Założenia: Testujemy w jaki sposób modyfikacja algorytmu ID3 wpływa na średnią głębokość drzew, generowanych za jego pomocą. Zakładamy, że dla zmodyfikowanej wersji średnia głębokość będzie rosła proporcjonalnie do wielkości zbioru uczącego, a dla klasycznej – szybko się ustabilizuje. Przez głębokość drzewa rozumiemy maksymalną ilość atrybutów które muszą zostać sprawdzone do uzyskania odpowiedzi w drzewie.

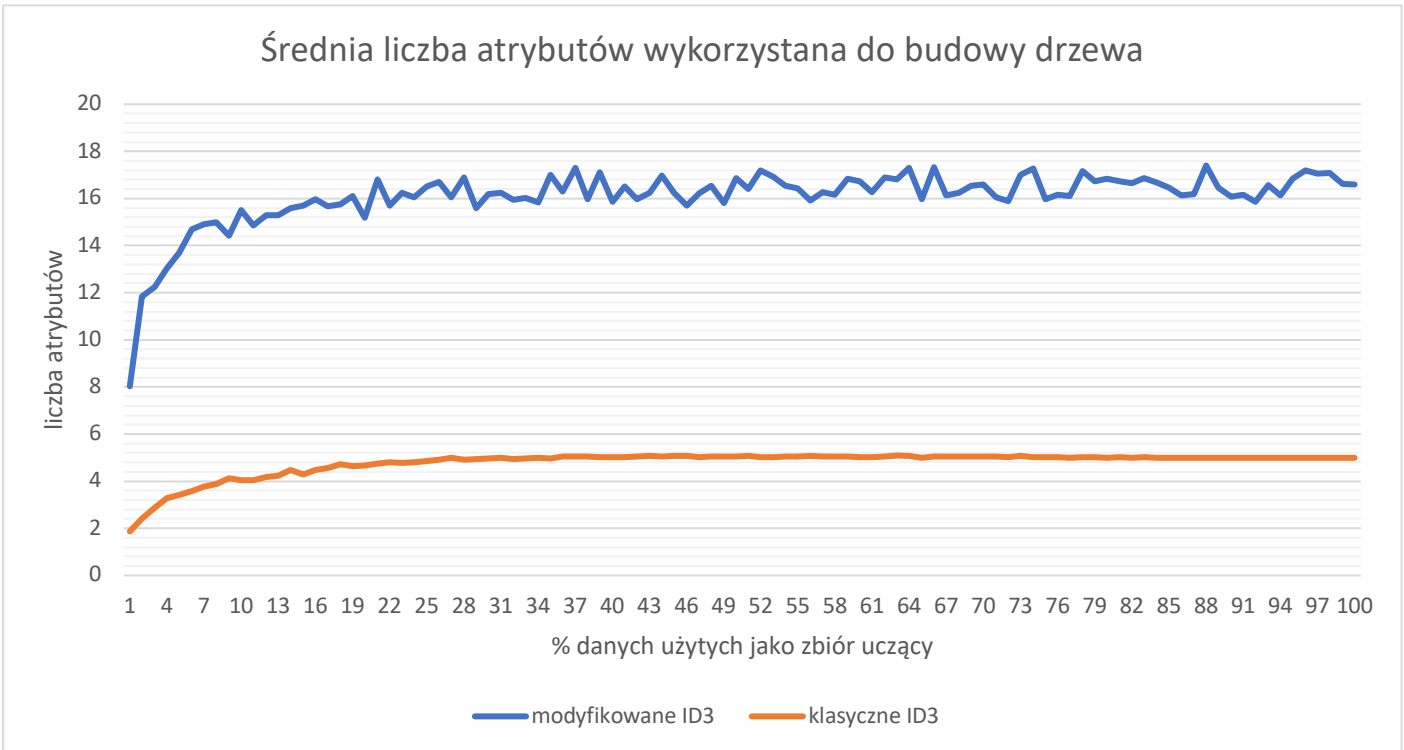
   b. Eksperyment: Badamy średnią głębokość generowanego drzewa decyzyjnego w zależności od ilości danych uczących, liczoną dla obu implementacji algorytmu ID3.



c. Wniosek: Podczas gdy średnia głębokość drzewa decyzyjnego, wygenerowanego przez klasyczny ID3, znalazła optymalny podział zbioru testowego na klasy przy użyciu czterech atrybutów, średnia głębokość drzewa decyzyjnego dla modyfikacji ID3 oscylowała pomiędzy sześcioma a siedmioma atrybutami. Oznacza to, że zmodyfikowany ID3 tworzy bardziej głębokie drzewo decyzyjne, próbując w ten sposób zniwelować wpływ losowości przy decydowaniu o optymalnych atrybutach dzielących i dojść do maksymalnego pokrycia danych testowych.

4. Średnia ilość wykorzystanych atrybutów.
   a. Założenia: Zależało nam na sprawdzeniu, ile atrybutów jest branych pod uwagę w obu rozpatrywanych przez nas implementacjach algorytmu ID3 przy generacji drzew decyzyjnych, i co można za pomocą tych danych powiedzieć o modyfikacji algorytmu. Zakładamy, że dla klasycznej wersji algorytmu wartość ta szybko się ustabilizuje, a dla zmodyfikowanej będzie rosnąć proporcjonalnie do wielkości zbioru uczącego.

   b. Eksperyment: Badamy średnią ilości atrybutów wykorzystanych przez obie implementacje, z zależnością co do procentu wykorzystania danych do uczenia się.



c. Wniosek: Od samego początku jest bardzo mocno widoczna różnica w ilości atrybutów wykorzystanych przez zmodyfikowany ID3. Gdy klasyczna implementacja potrzebuje 5 atrybutów by całkowicie pokryć i sklasyfikować zbiór danych, zmodyfikowana (z powodu semi-losowości przy wybieraniu atrybutów do oceny) potrzebuje ich ponad 15 na 22 atrybuty dostępne.

Wnioski:

- W każdym rozpatrywanym przez nas przypadku, zmodyfikowany algorytm ID3 sprawdzał się gorzej w zadaniu przewidywania czy grzyb jest jadalny, od klasycznego ID3.
- Głównym powodem niższej niż algorytm klasyczny ID3 skuteczności jest modyfikacja, polegająca na losowym doborze nagłówków służących jako separatory klas, co można zauważyć w każdym z przeprowadzonych przez nas eksperymentów. Chociaż czasem metoda zmodyfikowana dawała minimalnie lepsze lub identyczne wyniki, to były to jedynie pojedyncze przypadki z przetestowanych 10000 uruchomień zmodyfikowanej wersji ID3.
- Zasada koła ruletki jest gorszą metodą doboru testów w drzewie od stałej zasady wyboru atrybutu o największym przyroście informacyjnym, ponieważ do wystarczająco dobrej zasady o minimalizowaniu entropii wprowadza element losowości.
- W najlepszym wypadku, zmodyfikowany w ten sposób algorytm ID3 może działać tak samo dobrze jak klasyczny algorytm. W przypadku uwzględniania zagregowanych średnich czasów działania algorytmu czy parametrów produkowanego drzewa decyzyjnego, nie ma jednak szansy na to, by modyfikacja pozwoliła na poprawienie działania algorytmu ID3.
- Dzięki przeprowadzonym badaniom nauczyliśmy się, że w ten sposób sformułowana modyfikacja algorytmu ID3 nie nadałaby się do optymalnego konstruowania drzew decyzyjnych.

Instrukcja obsługi i sposób odtworzenia eksperymentów:
Program uruchamiany jest poleceniem:

python main.py iterations percentage if_print_tree

Gdzie za *iterations* wpisujemy liczbę iteracji algorytmów, za *percentage* wpisujemy liczbę określającą jaki procent zbioru danych będzie zbiorem przykładów uczących, a za *if_print_tree* wpisujemy 1 jeśli chcemy wyświetlić drzewo decyzyjne lub 0 jeśli nie chcemy (1 zalecana jest tylko dla pojedynczego uruchomienia algorytmów).
Przykład:

python main.py 100 5 0

Program wykona 100 iteracji dla obu algorytmów, gdzie zbiorem uczącym będzie 5% zbioru danych, drzewo decyzyjne nie zostanie wypisane.
Program nie jest związany stale z wykorzystywaną przez nas w eksperymentach bazą danych. Można wykorzystać dowolną bazę danych uczących w formacie csv, po zmianie jej nazwy na „mushroom.txt" i upewnieniu się, że pierwszy wiersz używanej bazy danych zawiera nagłówki kolumn.

Aby odtworzyć eksperyment:
Należy wykorzystać skrypt, wykorzystywany przez nas do badania poszczególnych współczynników obu implementacji algorytmu ID3: script1.sh.
Script1.sh jest skryptem bashowym, który po wywołaniu pozwala na uruchomienie działalności wytworzonego przez nas programu wprowadzającego i testującego obie implementacje ID3 wiele razy, celem sprawdzenia poprawności agregacji wyników poszczególnych instancji problemu. Zmienna max pozwala na ograniczenie ilości uruchamianych instancji programu testowego do zadanej liczby n, jeżeli przykładowo nie chcemy zagregować oddzielnych danych i je oddzielnie uśrednić dla lepszej dokładności, a tylko jednokrotnie uruchomić program testujący. Dla szybkiego przetestowania skryptu sugerujemy ustawienie zmiennej max na np. 5.