

Biblioteka wspomagająca uczenie maszynowe obwodów logicznych

Prezentacja postępów

Wstęp

Praca ma na celu implementację biblioteki, która udostępnia interfejs użytkownika oraz algortmy pozwalające wygenerować zbiór formuł logicznych(CNF), na podstawie zbioru treningowego. Formuły wygenerowane w taki sposób będą miały na celu weryfikować przynależność do danej klasy decyzyjnej poprzez głosowanie, w którym o przynależności do danej kategorii będzie decydować największa ilość spełnionych formuł z danej klasy decyzyjnej.

Użyte technologie:

- Python – jest on wykorzystywany do pracy z danymi oraz do stworzenia interfejsu(PyQt)
- C++ - w tym języku został napisany algorytm generujący formuły logiczne, algorytm działa na danych binarnych przygotowanych przez moduły napisane w pythonie
- Docker – w celu uproszczenia uruchamiania algorytmu generującego formuły logiczne

Praca w skrócie

- Interfejs użytkownika do zarządzania konwersją danych na binarne
- Interfejs użytkownika do zarządzania parametrami algorytmu uczenia formuł logicznych
- Moduł odpowiedzialny za generowanie formuł logicznych na podstawie binarnego zbioru treningowego
- Interfejs użytkownika pozwalający przeanalizować skuteczność wygenerowanych formuł

Interfejs użytkownika do zarządzania konwersją danych na binarne

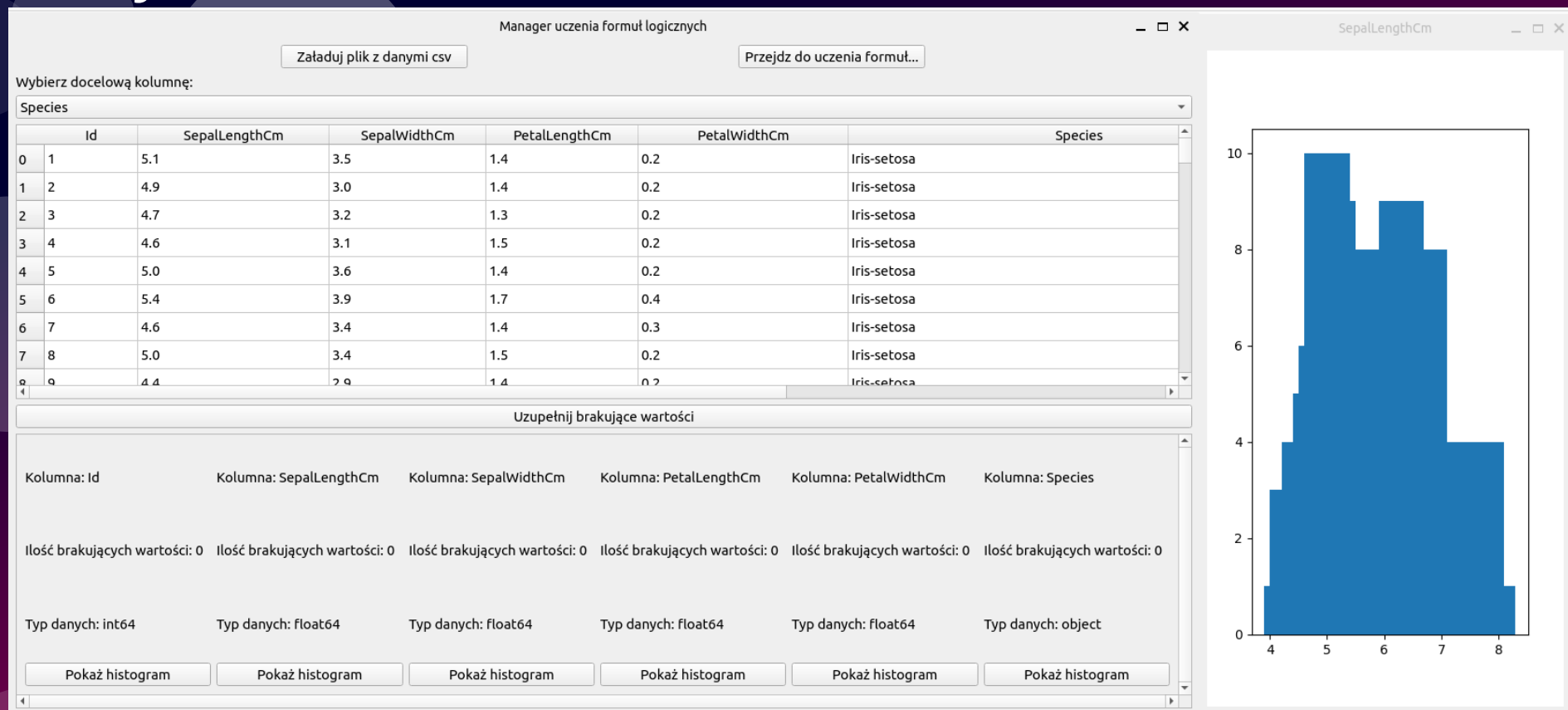
Zadania interfejsu:

- Umożliwienie przeglądania danych i wyrobienia sobie o nich obrazu
- Umożliwienie zarządzania usuwaniem oraz wypełnianiem brakujących danych
- Umożliwienie zarządzania konwersją liczb zmiennoprzecinkowych na liczby całkowite
- Umożliwienie zarządzania konwersją danych typu kategorycznego na zmienne całkowite
- Umożliwienie zarządzania zakresem przedziałów wartości, które otrzymają ten sam kod binarny
- Przygotowanie plików z danymi binarnymi dla algorytmu uczenia maszynowego

Interfejs użytkownika do zarządzania konwersją danych na binarne cz.2

- Interfejs udostępnia podgląd na tabelę z danymi oraz pokazuje takie informacje jak ilość brakujących wartości w danej kolumnie, dla wartości numerycznych (średnia, wartość maksymalna, mediana) oraz histogram pokazujący rozkład wartości

Interfejs użytkownika do zarządzania konwersją danych na binarne cz.3



Interfejs użytkownika do zarządzania konwersją danych na binarne cz.4

- Interfejs umożliwia usuwanie brakujących wartości ze zbioru. Po kliknięciu przycisku „Uzupełnij brakujące wartości” następuje oczyszczanie zbioru danych z wartości null.
- Zasady uzupełniania:
 - Brakuje mniej niż 10% wartości(uzupełniamy wartość średnią), obiekty dostają wartość Not assigned
 - Brakuje mniej niż 40% wartości – usuwamy wiersze z nullami
 - Więcej niż 40% wartości – usuwamy całą kolumnę
 - Wartości te docelowo będą konfigurowalne(użytkownik będzie mógł zmodyfikować progi procentowe)

Interfejs użytkownika do zarządzania konwersją danych na binarne cz.5

- Interfejs pozwala jednym kliknięciem przekonwertować wartości typu float na wartości całkowite. Domyślnie wartość jest mnożona razy 100, wartości po przecinku są ucinane. Docelowo będzie to konfigurowalne

Id		SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
7	8	5.0	3.4	1.5	0.2	Iris-setosa
8	9	4.4	2.9	1.4	0.2	Iris-setosa

Przekonwertuj na wartości numeryczne

Interfejs użytkownika do zarządzania konwersją danych na binarne cz.6

- Kolejnym etapem konwersji danych jest zakodowanie obiektów na wartości typu integer. Mamy to do czynienia z parametrem, który definiuje maksymalną ilość atrybutów kategorycznych. Docelowo będzie on modyfikowalny.

Id		SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	510	350	140	20	Iris-setosa
1	2	490	300	140	20	Iris-setosa
2	3	470	320	130	20	Iris-setosa
3	4	460	310	150	20	Iris-setosa
4	5	500	360	140	20	Iris-setosa
5	6	540	390	170	40	Iris-setosa
6	7	460	340	140	30	Iris-setosa
7	8	500	340	150	20	Iris-setosa
8	9	440	290	140	20	Iris-setosa

Zakoduj obiekty jako integer

Interfejs użytkownika do zarządzania konwersją danych na binarne cz.7

- Na ostatnim etapie konwersji interfejs umożliwia zarządzanie przedziałami kodowania danych na binarne

The screenshot shows a window titled "Zmień ilość wartości:" (Change number of values:). It contains a table for defining ranges for 6 values (0 to 5). Each value has two input fields for the range. Below the table is a list of columns to select for applying these settings, and a "Zastosuj" (Apply) button at the bottom.

Zmień ilość wartości:	
5	
Wartość: 0	1 25
Wartość: 1	25 50
Wartość: 2	50 75
Wartość: 3	75 100
Wartość: 4	100 125
Wartość: 5	125 150

Wybierz kolumny, do zaaplikowania tych samych ustawień

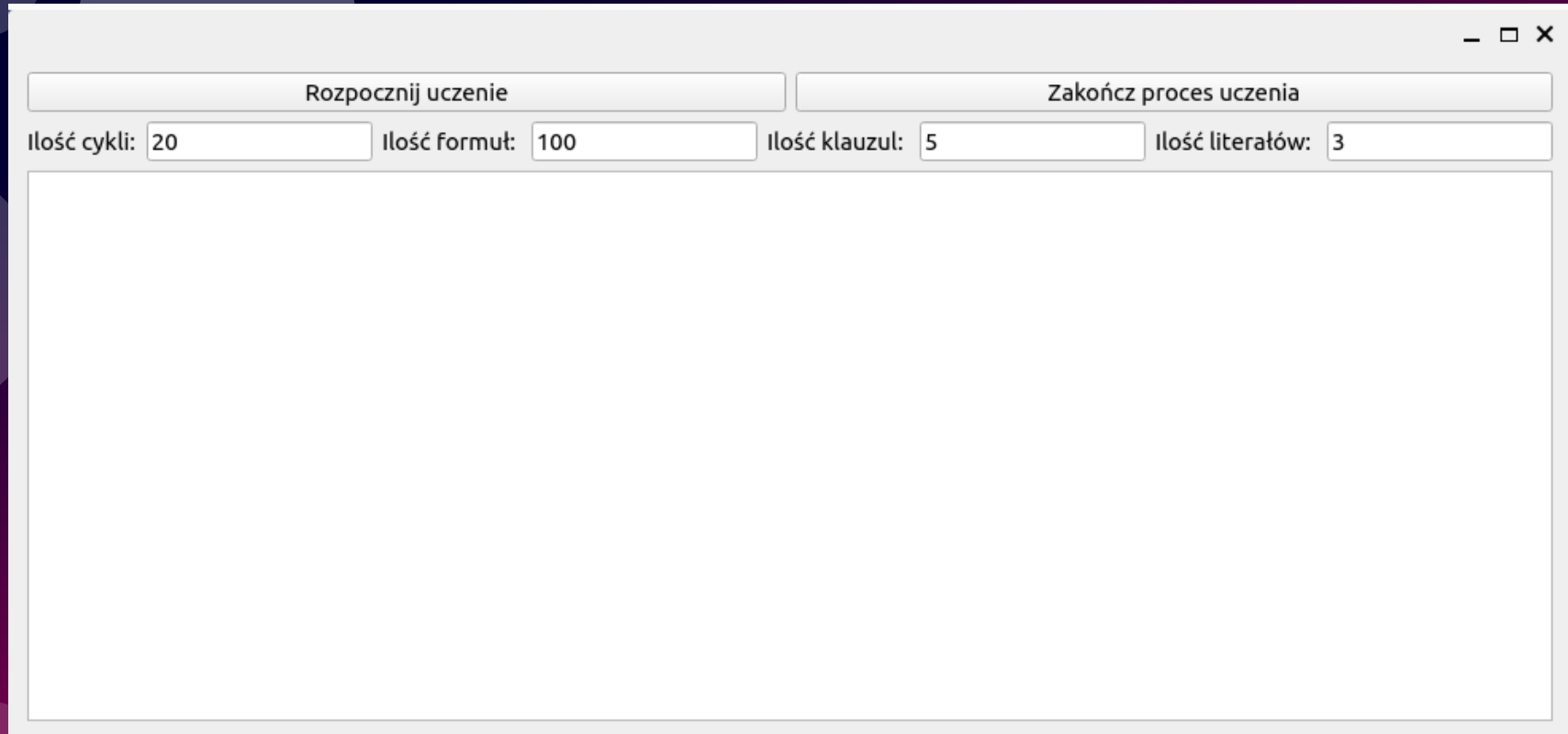
- Id
- SepalLengthCm
- SepalWidthCm
- PetalLengthCm
- PetalWidthCm
- Species

Zastosuj

Interfejs użytkownika do zarządzania konwersją danych na binarne cz.8

- Po wykonaniu wszystkich kroków konwersji dane binarne są zapisywane w postaci binarnej zrozumiałej dla algorytmu uczenia formuł logicznych
- Planowana jest, też funkcjonalność, która pozwoli zapamiętać wszystkie wyboru użytkownika do pliku i umożliwić ich wczytania, aby nie trzeba było powtarzać za każdym razem tych samych czynności

- Interfejs użytkownika do zarządzania parametrami algorytmu uczenia formuł logicznych



The image shows a graphical user interface (GUI) window for managing the parameters of a logic formula learning algorithm. The window has a standard title bar with minimize, maximize, and close buttons. Below the title bar, there are two buttons: "Rozpocznij uczenie" (Start learning) and "Zakończ proces uczenia" (End learning process). Below these buttons, there are four input fields for parameters: "Ilość cykli:" (Number of cycles) with a value of 20, "Ilość formuł:" (Number of formulas) with a value of 100, "Ilość klauzul:" (Number of clauses) with a value of 5, and "Ilość literałów:" (Number of literals) with a value of 3. The main area of the window is a large, empty white rectangle, likely intended for displaying the progress or results of the learning process.

Parameter	Value
Ilość cykli:	20
Ilość formuł:	100
Ilość klauzul:	5
Ilość literałów:	3

Moduł odpowiedzialny za generowanie formuł logicznych na podstawie binarnego zbioru treningowego

- Moduł ten działa obecnie w prosty sposób. Pobiera on 4 parametry, ilość formuł dla danej klasy decyzyjnej, ilość klauzul w formule, oraz ilość literałów. Następnie dla każdej klasy decyzyjnej w sposób losowy są generowane formuły. Losowanie stara unikać się powtórzeń tych samych wartości. Dla każdej klasy decyzyjnej generujemy formuły dwóch typów (takie, które mają na celu spełniać jak najwięcej formuł z danej klasy decyzyjnej oraz takie, które mają negować jak najwięcej formuł z innych klas decyzyjnych)

Moduł odpowiedzialny za generowanie formuł logicznych na podstawie binarnego zbioru treningowego cz.2

Mamy tu do czynienia z wieloma klasami decyzyjnymi, w związku z tym generowanie formuły negujących dane z innej klasy decyzyjnej przebiega w ten sposób, że dla każdej klasy decyzyjnej inna niż docelowa formuły, ilość generowanych negatywnych formuł jest równo rozdzielana pomiędzy resztę klas decyzyjnych.

Moduł odpowiedzialny za generowanie formuł logicznych na podstawie binarnego zbioru treningowego cz.3

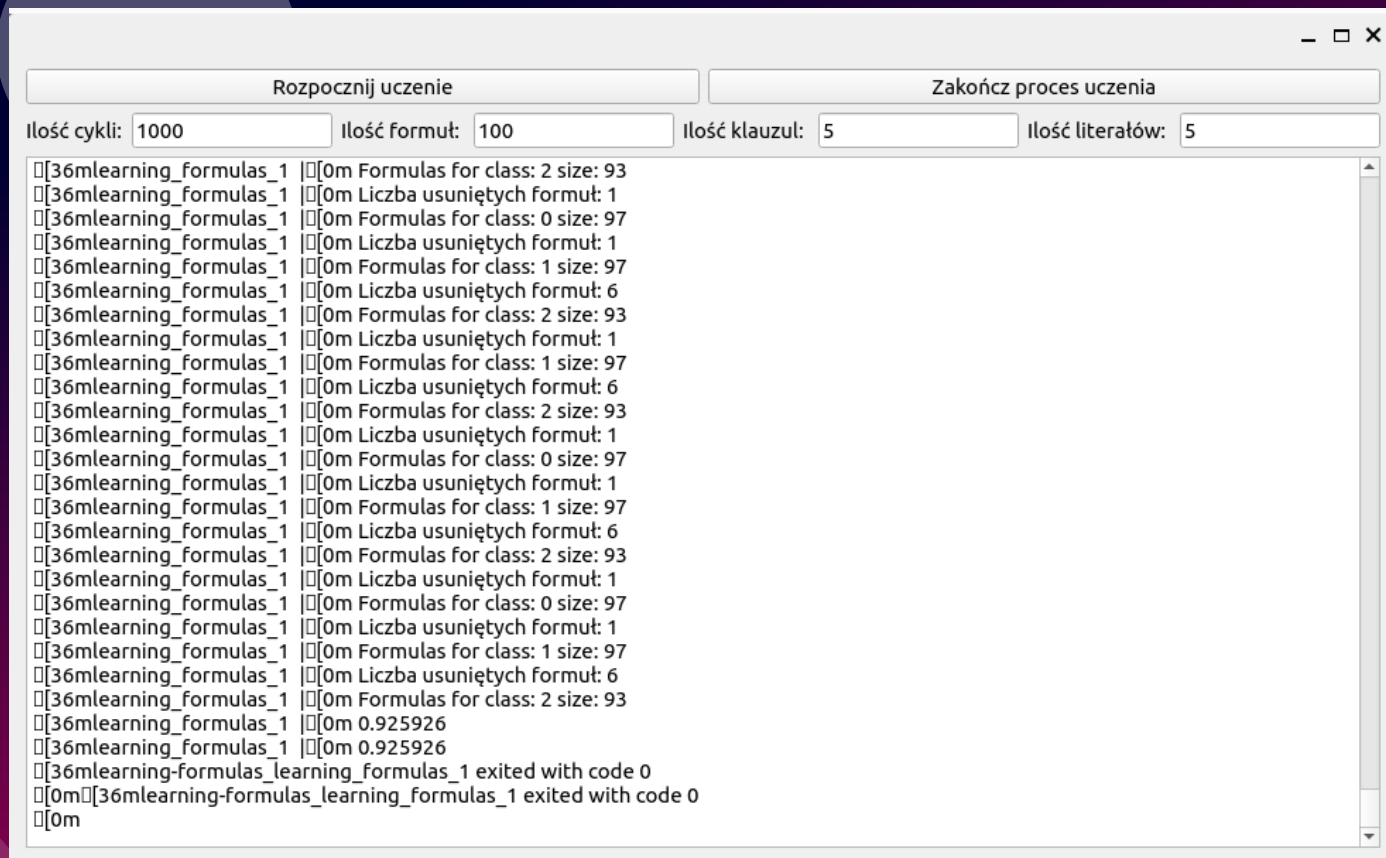
Algorytm ten jest rozwinięciem algorytmu zastosowanego w innej pracy magisterskiej. Wprowadzone ulepszenia względem oryginału:

- Nieograniczona ilość klas decyzyjnych
- Nowy parametr – wymagany % pozytywnych odpowiedzi dla danej klasy decyzyjnej
- Optymalizacja czasu wykonywania algorytmu poprzez zastosowanie języka C++ oraz użycie zbiorów, które pozwalają szybciej weryfikować jaki element był już wylosowany
- Optymalizacja przechowywania danych binarnych w pamięci poprzez przechowywanie tylko pozytywnych bitów oraz ich pozycji(sprawdza się gdy większość bitów ma wartość false)
- Planowane zastosowanie algorytmu ewolucyjnego

Dziękuję za uwagę!

- Michał Jakubowski

Przykład uruchomienia algorytmu na zbiorze Iris



Interfejs użytkownika pozwalający przeanalizować skuteczność wygenerowanych formuł

Interfejs nie został jeszcze zaimplementowany. Planowane funkcjonalności to.

- Możliwość wczytania formuł logicznych
- Możliwość sprawdzenia skuteczności formuł na wczytanym zbiorze danych(dane będą automatycznie kodowane do postaci binarnej według tych samych reguł, które były użyte do kodowania zbioru treningowego)
- Możliwość wykonania predykcji klasy decyzyjnej