Politechnika Wrocławska

Wydział Elektroniki

Zastosowania informatyki w medycynie

Komputerowe wspomaganie diagnozowania białaczek u dzieci z wykorzystaniem drzewa decyzyjnego

Autorzy:
Wojciech Czarnecki 235714
Piotr Stachnio 241268

Prowadzący:
Dr inż. Mariusz Topolski

Termin zajęć: Piątek N 13:15

Spis treści

1 Opis problemu medycznego jako zadania klasyfikacji oraz wyznaczenie rankingu cech pod względem ich przydatności

1.1 Opis projektu

Celem realizowanego przez nas projektu jest nabycie umiejętności zastosowania algorytmu klasyfikacji nadzorowanej (w naszym przypadku algorytmu drzewa decyzyjnego) w zadaniu diagnozowania białaczek u dzieci. Ten proces wymaga odpowiedniego wyselekcjonowania cech, dzięki którym będzie można rozpoznać chorobę u pacjenta. Dzięki danym rzeczywistym będzie można ocenić w przyszłości skuteczność wykorzystanego w tym projekcie algorytmu i sprawdzić, w jaki sposób jakość klasyfikacji zależy od liczby atrybutów wykorzystanych do skonstruowania modelu.

Wyszczególniliśmy następujące etapy realizacji projektu:

1. Zapoznanie się z algorytmem drzew decyzyjnych.
2. Zapoznanie się z danymi rzeczywistymi - analiza danych wejściowych, określenie liczby i znaczenia klas oraz dokonanie charakterystyki i znaczenia cech.
3. Opracowanie sposobu wyznaczania rankingu cech z wykorzystaniem aplikacji `Tibco Statistica 13`.
4. Zaplanowanie badań eksperymentalnych i implementacja algorytmu klasyfikacji.
5. Przeprowadzenie badań eksperymentalnych.
6. Analiza wyników badań i wyciągnięcie wniosków.
7. Przygotowanie dokumentacji projektu.

1.2 Charakterystyka analizowanego problemu medycznego

Do badań wykorzystamy dane przekazane przez prowadzącego, które zawierają następujące informacje:

- Istnieje 410 pacjentów ze zdiagnozowaną chorobą białaczki,
- Pacjenci zostali przypisani do jednej z 20 klas chorobowych oznaczających typy białaczki,
- Każdy z pacjentów został zbadany przez lekarza i opisany za pomocą 20 cech,
- Każda cecha jest typu binarnego lub dyskretnego co oznacza, że przyjmuje wartości z określonego zbioru wartości.

Większość cech ma charakter binarny, czyli posiada tylko dwie wartości (w tym przypadku 1 lub 2), np. temperatura, stopień krwawienia czy liczba płytek krwi. Jest to najprostsza odmiana atrybutu kategorycznego. W zbiorze cech znaleźć można też kilka cech kategorycznych, które przyjmować mogą kilka wartości z grupy możliwych opcji, np. anemia, miejsce krwawienia lub informacja o głównych komórkach w szpiku.

W przedstawionej na następnej stronie tabeli 1 znajdują się te właśnie cechy oraz ich charakter, na podstawie których można ocenić to, czy pacjent jest lub nie chory na białaczkę.

1.3 Zestaw badanych cech

Tabela 1: Zestaw badanych cech i ich wartości

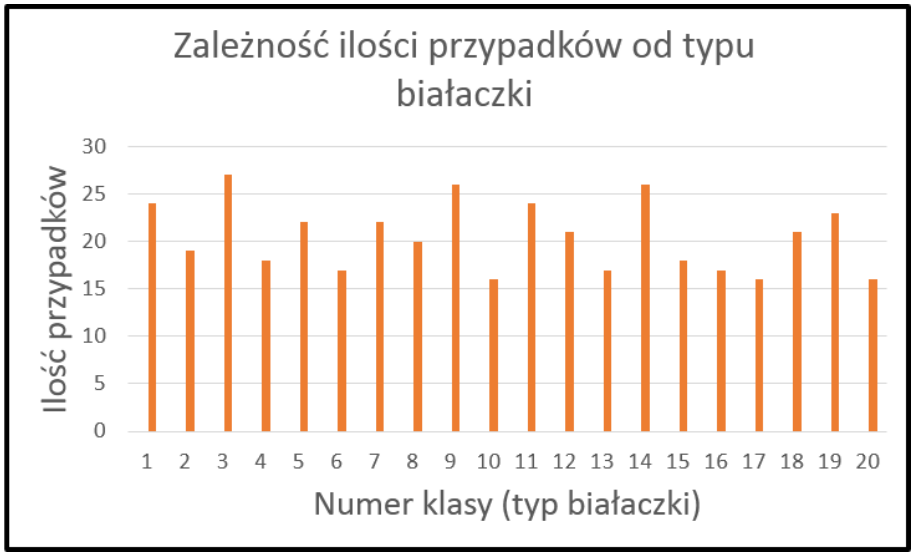
L.p.	Nazwa cechy	Przyjmowane wartości	Charakter cechy
1	Temperatura	1 - regularna 2 - nieregularna	binarna
2	Anemia	1 - średnia 2 - średnio-ciężka 3 - ciężka	dyskretna
3	Stopień krwawienia	1 - mały, 2 - duży	binarna
4	Miejsce krwawienia	1 - skóra 2 - jama ustna 3 - jama nosowa 4 - krwawienie do siatkówki oka 5 - drogi oddechowe 6 - przewód moczowy 7 - przewód trawienny 8 - mózg	dyskretna
5	Bóle kości	1 - tak, 2 - nie	binarna
6	Wrażliwość mostka	1 - tak, 2 - nie	binarna
7	Powiększenie węzłów chłonnych	1 - nieznaczne 2 - silne	binarna
8	Powiększenie wątroby i śledziony	1 - nieznaczne 2 - silne	binarna
9	Centralny układ nerwowy (ból głowy, wymioty, drgawki, senność, śpiączka)	1 - tak 2 - nie	binarna
10	Powiększenie jąder	1 - tak, 2 - nie	binarna
11	Uszkodzenie w sercu, płucach, nerce	1 - tak, 2 - nie	binarna
12	Gałka oczna (zaburzenia w widzeniu, krwawienie do siatkówki, wytrzeszcz oczu)	1 - tak 2 - nie	binarna
13	Poziom WBC (leukocytów)	1 - powiększony 2 - obniżony 3 - normalny	dyskretna
14	Obniżenie liczby RBC (erytrocytów)	1 - lekkie 2 - średnie 3 - duże	dyskretna
15	Liczba płytek krwi	1 - obniżone 2 - normalne	binarna
16	Niedojrzałe komórki (blastyczne)	1 - istnieją 2 - nie istnieją	binarna
17	Stan pobudzenia szpiku	1 - krańcowo czynny 2 - średnio czynny 3 - czynny	dyskretna
18	Główne komórki w szpiku	1 - prymitywne i niedojrzałe 2 - wcześniej niedojrzałe granulocyty 3 - dojrzałe	dyskretna
19	Poziom limfocytów	1 - duży 2 - niski 3 - nieregularny	dyskretna
20	Reakcja	1 - negatywna 2 - pozytywna	binarna

1.4 Zestaw badanych klas

W poniższej tabeli 2 znajduje się dwadzieścia klas oznaczających daną chorobę u pacjenta:

Tabela 2: Zestaw badanych klas i ilość chorych

Numer klasy	Nazwa klasy	Ilość chorych
1	Postać nie T I nie B (L1 - type)	24
2	Postać T (L2 - type)	19
3	Postać B (L3 - type)	27
4	Mieloblastyczna o niskim niezróżnicowaniu	18
5	Mieloblastyczna z dojrzewaniem	22
6	Promielocytowa	17
7	Mielomonoblastyczna	22
8	Monoblastyczna	20
9	Cytoeukemia	26
10	Podostra granulocytarna	16
11	Granulocytarna	24
12	Limfocytarna	21
13	Mielomonocytarna	17
14	Monocytarna	26
15	Chłoniak limfatyczny białaczka	18
16	Plazmocytowa	17
17	Wielokapilarnokomórkowa	16
18	Eozynofilowa	21
19	Bazofilowa	23
20	Białaczka komórek wielkojądrzastych	16



Rysunek 1: Zależność ilości przypadków białaczki od jej typu.

1.5 Stworzenie rankingu cech

Selekcja cech jest kluczowym problemem w modelowaniu obiektów, procesów i zjawisk, istotnym w rozpoznawaniu obrazów, uczeniu maszynowym, eksploracji danych i w sztucznej inteligencji. Celem selekcji cech jest redukcja wymiaru wektora wejściowego (obrazu) poprzez znalezienie podzbioru cech (zmiennych) opisujących obiekt w najlepszy sposób i zapewniających najwyższą jakość modelu (np. klasyfikatora, aproksymatora). Cechy nie przenoszące informacji, nieistotne lub nadmiarowe zostają wyeliminowane. Spodziewanym rezultatem selekcji cech jest redukcja „przekleństwa wymiarowości", poprawa dokładności i uproszczenie modelu, poprawa generalizacji, skrócenie czasu konstrukcji modelu oraz redukcja kosztu pozyskania danych. [1]

Do uzyskania rankingu cech wykorzystano algorytm chi-kwadrat (χ^2). Test istotności chi-kwadrat dokonuje weryfikacji zakładanej hipotezy, która zakłada, że pewne cechy zbioru A oraz zbioru B są niezależne statystycznie.

Test χ^2 może być zastosowany przy badaniach współzależności dwóch zmiennych. Aby przyjąć określone hipotezy pozwala on zweryfikować czy dana zbiorowość statystyczna ma właściwy typ rozkładu, czy cechy poddane analizie statystycznej posiadają wymaganą współzależność. Inaczej ujmując za pomocą testu χ^2 można ustalić prawdopodobieństwo wystąpienia zjawiska x, gdy zaistnieje zjawisko y i na odwrót. [2]

Wartość funkcji chi-kwadrat oblicza się według poniższego wzoru:

$$\chi^2 = \sum_{i=1}^{r} \frac{(n_i - np_i)^2}{np_i}, \qquad (1)$$

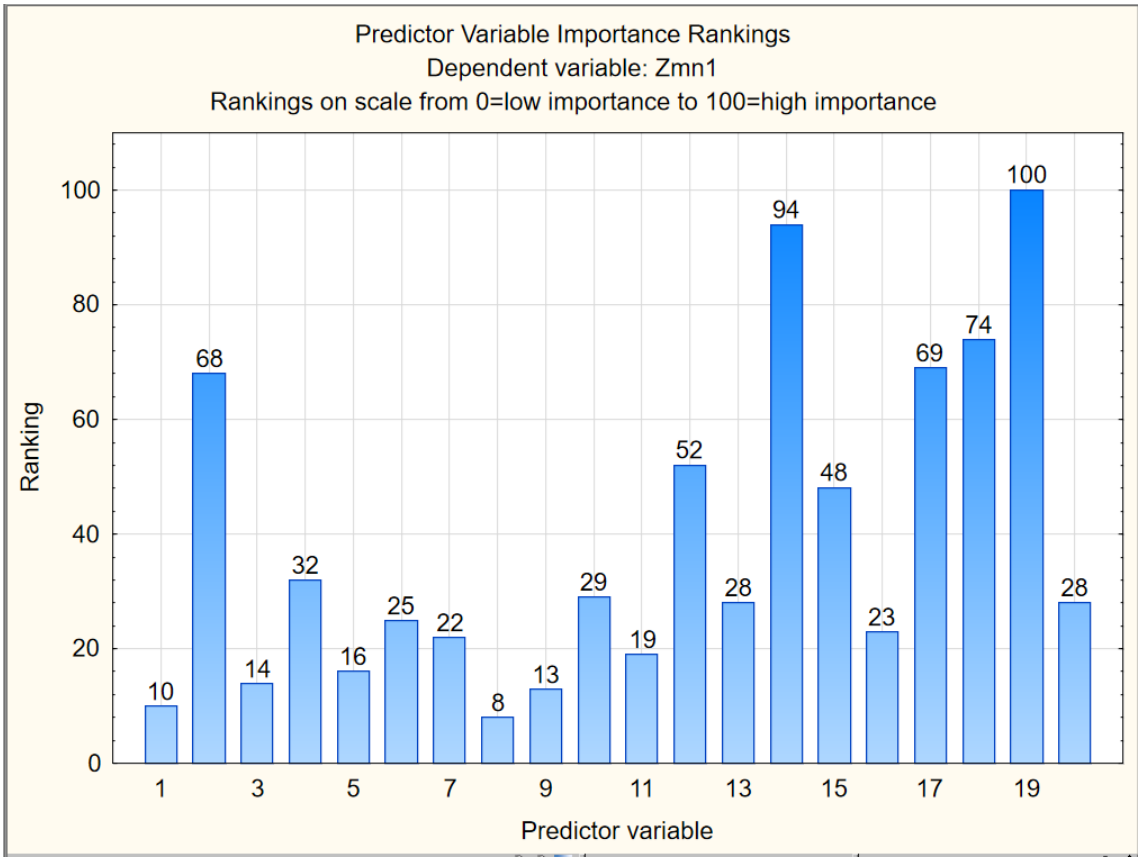
r - liczba przedziałów klasowych
n - liczba obserwacji
np_i - liczba jednostek, które powinny znaleźć się w i-tym przedziale przy założeniu, że cecha ma rozkład zgodny z hipotezą zerową. [3]

Ranking cech został stworzony w aplikacji `Tibco Statistica 13` na podstawie danych o wszystkich obiektach i znajduje się w poniższej tabeli 3.

Tabela 3: Wartość statystyki chi-kwadrat dla poszczególnych cech

Numer cechy	Nazwa cechy	Wartość χ^2
4	Miejsce krwawienia	187,43
19	Poziom limfocytów	53,43
18	Główne komórki w szpiku	37,61
14	Obniżenie liczby RBC (erytrocytów)	36,34
2	Anemia	35,16
6	Wrażliwość mostka	31,84
1	Temperatura	30,63
16	Niedojrzałe komórki (blastyczne)	28,26
10	Powiększenie jąder	26,62
3	Stopień krwawienia	26,26
7	Powiększenie węzłów chłonnych	25,78
5	Bóle kości	23,46
17	Stan pobudzenia szpiku	21,55
15	Liczba płytek krwi	20,26
11	Uszkodzenie w sercu, płucach, nerce	20,08
9	Centralny układ nerwowy (ból głowy, wymioty, drgawki, senność, śpiączka)	17,23
12	Gałka oczna (zaburzenia w widzeniu, krwawienie do siatkówki, wytrzeszcz oczu)	16,04
8	Powiększenie wątroby i śledziony	15,99
13	Poziom WBC (leukocytów)	14,47
20	Reakcja	14,47

Dla sprawdzenia, czy ranking cech został wygenerowany poprawnie na podstawie testu χ^2, wykorzystano również metodę drzew klasyfikacyjnych. Uzyskano go w następujący sposób: Zakładka Statystyka => Analizy wielowymiarowe => Drzewa klasyfikacyjne. Następnie wybrano zmienną zależną/grupującą (klasę) i zmienne niezależne(predyktory ilościowe), czyli wartości cech.



Rysunek 2: Ranking cech uzyskany na podstawie metody drzew.

Na podstawie macierzy korelacji uzyskanych w programie `Tibco Statistica 13` określono również wartości p dla każdej cechy, które posortowano malejąco. Uzyskano je w następujący sposób: Zakładka Statystyka => Statystyki podstawowe => Macierze korelacji => Dwie listy zmiennych. Następnie wybrano zmienne niezależne(predyktory ilościowe), czyli wartości cech jako pierwsza lista i zmienną zależną jako druga lista. Następnie wybrano Opcje => Wyświetl r. p i N => Podsumowanie.

Tabela 4: Wartość p dla poszczególnych cech

Numer cechy	Nazwa cechy	Wartość p
18	Główne komórki w szpiku	0,964
7	Powiększenie węzłów chłonnych	0,900
15	Liczba płytek krwi	0,875
1	Temperatura	0,737
6	Wrażliwość mostka	0,702
14	Obniżenie liczby RBC (erytrocytów)	0,518
10	Powiększenie jąder	0,480
8	Powiększenie wątroby i śledziony	0,249
5	Bóle kości	0,229
3	Stopień krwawienia	0,216
13	Poziom WBC (leukocytów)	0,054
20	Reakcja	0,054
16	Niedojrzałe komórki (blastyczne)	0,010
19	Poziom limfocytów	0,009
11	Uszkodzenie w sercu, płucach, nerce	0,004
9	Centralny układ nerwowy (ból głowy, wymioty, drgawki, senność, śpiączka)	0,004
2	Anemia	0,000
4	Miejsce krwawienia	0,000
12	Gałka oczna (zaburzenia w widzeniu, krwawienie do siatkówki, wytrzeszcz oczu)	0,000
17	Stan pobudzenia szpiku	0,000

2 Implementacja środowiska eksperymentowania

2.1 Przygotowanie środowiska eksperymentalnego

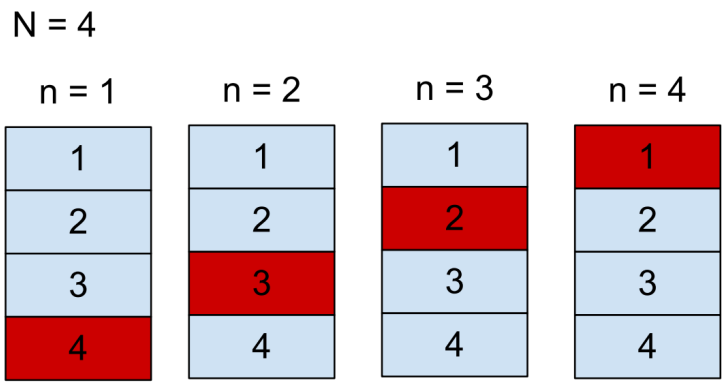
Zaimplementowane przez nas środowisko eksperymentalne powstało w oparciu o język programowania `Python` w wersji 3.9.0 wspomagany bibliotekami `pandas` oraz `sklearn`. Zgodnie z poleceniem, klasyfikatorem użytym przez nas jest klasa `DecisionTreeClassifier` z biblioteki `scikit-learn`. Jako początkową maksymalną głębokość drzewa wyznaczono wartość 3, a jako początkowe kryterium podziału wartość `gini`.

Poniżej znajdują się parametry, dla których zostaną przeprowadzone badania eksperymentalne.

- Typy kategoryzacji - `GINI, ENTROPY`
- Liczba cech - `[1, 20]`
- Głębokość drzewa - `[3,5,7]`

2.2 Walidacja krzyżowa

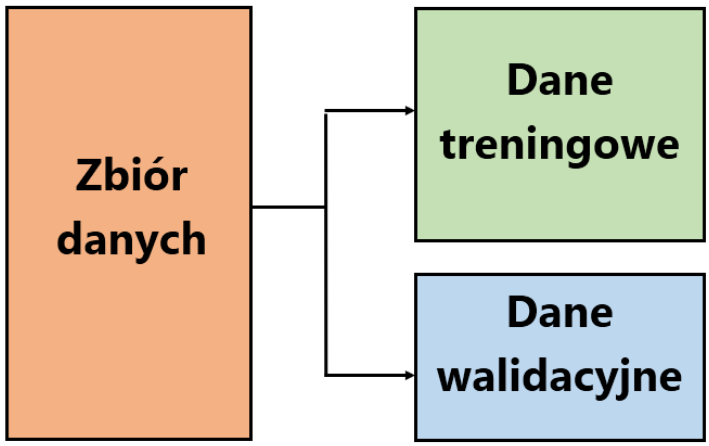
Walidacja krzyżowa polega na losowym podziale zbioru danych na N (najczęściej przyjmuje się N = 10) w miarę równo rozłożonych części (tzn. foldów). Walidacja odbywa się poprzez N-krotne wyuczenie klasyfikatora na zbiorze składającym się N - 1 części i przetestowaniu go na N-tej, niewykorzystanej w uczeniu części. Istotą tej metodyki testowania jest to, że w każdym kroku proces testowania odbywa się na innej części zbioru, a każda obserwacja ze zbioru będzie dokładnie raz przetestowana w procesie walidacji. Przykład działania metody walidacji krzyżowej (dla 4 foldów) pokazuje poniższy rysunek:



Rysunek 3: Przykład walidacji krzyżowej dla 4 foldów.

W pierwszym kroku (n=1) klasyfikator jest uczony z wykorzystaniem elementów 1,2,3 (kolor niebieski), a testowanie odbywa się na elemencie 4 (kolor czerwony). W następnym kroku (n=2) do testowania brany jest zbiór, który nie był jeszcze testowany, przykładowo ten o indeksie 3, a pozostałe części wykorzystywane są do uczenia. Proces jest powtarzany do momentu, w którym każda z części nie zostanie wykorzystana do testowania.[4]

W tym zadaniu należało przeprowadzić 5 razy powtórzoną 2-krotną walidację krzyżową. Podzielono zbiór uczący na N=2 części, a następnie każdy z foldów wykorzystywany jest jako zbiór uczący (dane treningowe), a drugi jako zbiór testowy (dane walidacyjne). Podział ten pokazano na poniższym rysunku:



Rysunek 4: Podział zbioru danych.

2.3 Implementacja walidacji krzyżowej

Do przeprowadzenia walidacji krzyżowej wykorzystano funkcję biblioteki `scikit-learn` o nazwie `cross_val_score`. Parametry tej funkcji zostały pokazane na poniższym listingu:

```
sklearn.model_selection.cross_val_score(estimator, X, y=None, *, groups=None,
scoring=None, cv=None, n_jobs=None, verbose=0, fit_params=None, pre_dispatch='2*n_jobs',
error_score=nan)
```

Listing 1: Parametry funkcji cross_val_score

Do programu wczytywany jest plik .xls, z którego pobierane są dane dotyczące klas (różnych typów białaczek) i przynależących do nich wartości cech ustawionych w ranking. Dane te algorytm dzieli na dane przeznaczone do nauki oraz dane przeznaczone do testowania za pomocą funkcji `train_test_split`. Są one przypisywane do obu zbiorów losowo. Jest to potrzebne do tego, aby uniknąć zjawiska `overfittingu`, które powoduje, że algorytm zamiast uczyć się, to "zapamiętuje" zbiór testowy.

```
sklearn.model_selection.train_test_split(*arrays, test_size=None, train_size=None,
random_state=None, shuffle=True, stratify=None)
```

Listing 2: Parametry funkcji train_test_split

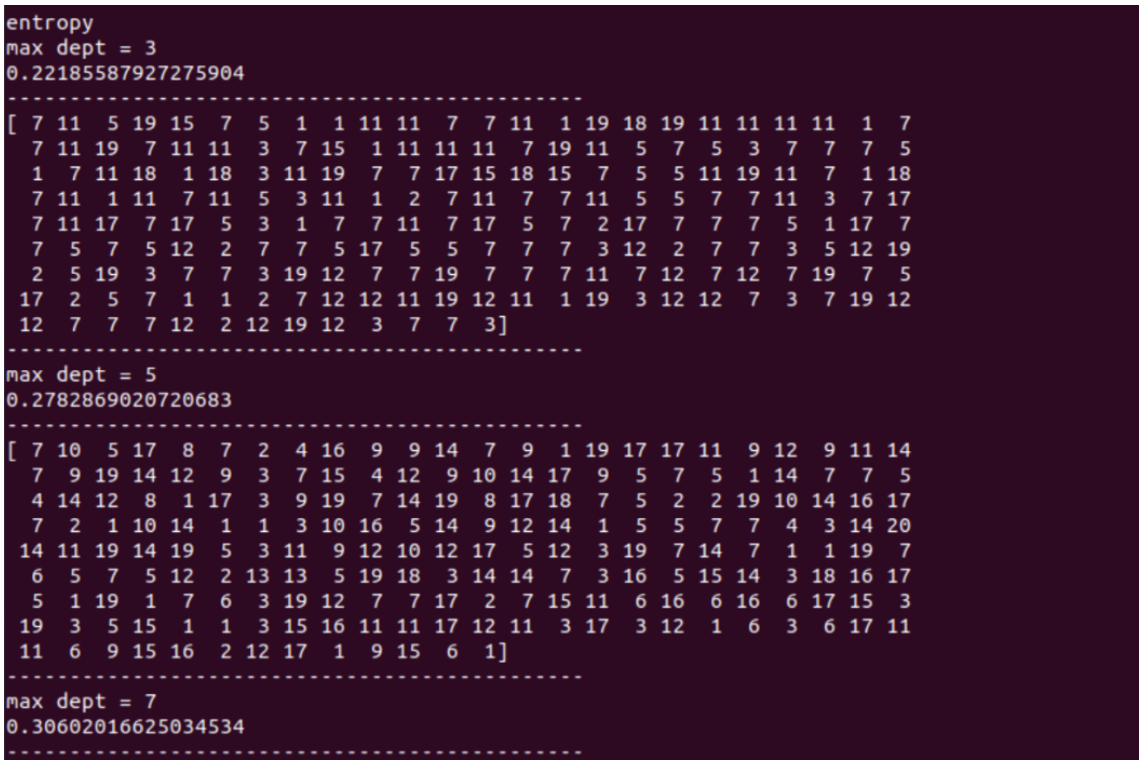
Następnie algorytm wykonuje dla dwóch wybranych rodzajów kategoryzacji (`Gini Impurity` oraz `Entropy`) naukę oraz testowanie dla drzewa `CART` o kolejnych maksymalnych jego głębokościach: 3, 5 oraz 7. Po wykonaniu algorytm zwraca dokładność, z jaką zostały zdiagnozowane kolejne typy białaczki. Wyświetlany jest również numer klasy, do której algorytm zakwalifikował dany przypadek.

Na poniższym rysunku znajduje się pseudokod przedstawiający działanie opisanego powyżej algorytmu.

```
INPUT:  X - array with instances
OUTPUT: Decision tree and classification scores
REQUIREMENTS: X !=0, data[] > 0
PARAMETERS: criteria = ['gini','entropy'], depth = [3,5,7]
for crit in criteria:
      for num in depth:
            clf <- DecisionTreeClassifier(crit, num)
            clf <- cfl.fit(trainingData, trainingAnswers)
            score <- cross_val_score(clf, testingData, testingAnswers, 5)
            return score.mean()
      end for
end for
```

Rysunek 5: Pseudokod dla algorytmu klasyfikacji CART.

Na poniższym zrzucie ekranu pokazano działanie opisanego powyżej algorytmu.



Rysunek 6: Działanie środowiska eksperymentalnego w języku Python.

2.4 Implementacja parowego testu t-studenta

Testy t-Studenta służą do porównania ze sobą dwóch grup, nie więcej. Korzystamy z nich wtedy, gdy mamy wyniki dla dwóch grup i chcemy porównać je ze sobą, tzn. stwierdzić, czy wyniki w jednej grupie są większe bądź mniejsze niż w drugiej grupie. Nie można porównywać ze sobą kilku grup, wykonując kilkukrotnie test t-Studenta. Jeżeli mamy więcej niż 2 grupy to musimy skorzystać z innych testów statystycznych.

Do implementacji testu t-studenta wykorzystano funkcję biblioteki `scipy.stats` o nazwie `ttest_ind`. Parametry tej funkcji zostały pokazane na poniższym listingu:

```
scipy.stats.ttest_ind(a, b, axis=0, equal_var=True, nan_policy='propagate',
alternative='two-sided')
```

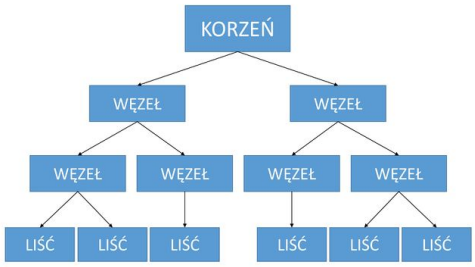
Listing 3: Parametry funkcji ttest_ind

3 Opis algorytmu klasyfikacji

3.1 Algorytm drzewa decyzyjnego

Drzewo decyzyjne to graficzny sposób wspierania procesu decyzyjnego. Drzewo stosowane jest w teorii decyzji i ma sporo zastosowań. Może zarówno rozwiązać problem decyzyjny, jak i stworzyć plan. Metoda drzew decyzyjnych sprawdza się przede wszystkim, kiedy mamy problemy decyzyjne z wieloma rozgałęziającymi się wariantami oraz kiedy podejmujemy decyzję w warunkach ryzyka. Drzewa znalazły zastosowanie w takich dziedzinach jak botanika i medycyna.

Technika drzew decyzyjnych pozwala na wyznaczenie zasad decyzyjnych opisujących reguły przypisywania obiektów do wyróżnionych klas oraz analizowanie zbioru obiektów opisywanych przez przyjęty zestaw atrybutów. Celem analizy jest doskonalenie podziału obiektów na jednorodne klasy, gdzie metoda dokonywania podziału ma charakter hierarchiczny. Punktem wyjścia jest zbiór zawierający wszystkie analizowane obiekty. W trakcie analizy jest on dzielony na określoną liczbę podzbiorów. W kolejnych krokach każdy z podzbiorów podlega dalszemu podziałowi, a na końcu analizy każdy obiekt stanowi oddzielną klasę.



Rysunek 7: Schemat drzewa decyzyjnego.

Drzewem decyzyjnym jest graf - drzewo, które składa się z korzenia, węzłów, krawędzi oraz liści. Liście to węzły, z których nie wychodzą już żadne krawędzie. Korzeń drzewa tworzony jest przez wybrany atrybut, natomiast poszczególne gałęzie reprezentują wartości tego atrybutu. Dzięki drzewu decyzyjnemu, zbudowanemu na podstawie danych empirycznych, można sklasyfikować nowe obiekty, które nie brały udziału w procesie tworzenia drzewa. Drzewa decyzyjne charakteryzują się strukturą hierarchiczną. Oznacza to, że w kolejnych krokach dzieli się zbiór obiektów, poprzez odpowiedzi na pytania o wartości wybranych cech lub ich kombinacji liniowych. Ostateczna decyzja zależy od odpowiedzi na wszystkie pytania. W algorytmach konstrukcji drzew jednym z kluczowych elementów jest wybór kolejności cech, według których, na poszczególnych etapach, będzie dokonywany podział zbioru obiektów.

Ogólną zasadę konstrukcji drzew decyzyjnych można opisać w następujący sposób:

1. Zbadanie, czy zbiór obiektów jest jednorodny. Jeśli jest, algorytm kończy pracę. Jeśli nie, to wykonywana jest dalsza część algorytmu.
2. Rozpatrywanie wszystkich możliwych podziałów zbioru obiektów na podzbiory oraz określenie, który z podziałów tworzy najbardziej jednorodne zbiory.
3. Podział zbioru w najlepszy sposób ze względu na przyjęte kryterium.
4. Użycie powyższego algorytmu do wszystkich podzbiorów.
5. Kategoryzacja drzewa, czyli likwidacja fragmentów drzewa o małym znaczeniu dla jakości rezultatów klasyfikacji.
6. Zastosowanie drzewa do klasyfikacji nowych obiektów.[5]

3.2 Algorytm CART

Opracowana w 1984 roku metoda `CART (Classification and regression trees)` jest nadal bardzo popularna i szeroko stosowana. Drzewa decyzyjne tworzone przez ten algorytm są binarne i zawierają dokładnie dwie gałęzie w każdym z węzłów.

`CART` może być stosowane zarówno do danych ciągłych jak i dyskretnych. W przypadku danych ciągłych stosuje się technikę zbliżoną do tej z algorytmu C4.5 - rozpatruje się wszystkie podziały na dwa przedziały zdeterminowane punktem podziału *a*. [6]

Metoda CART związana jest z dwoma typami drzew:

- Drzewa klasyfikacyjne wykorzystuje się do wyznaczania przynależności przypadków lub obiektów do klas jakościowej zmiennej zależnej na podstawie pomiarów jednej lub więcej zmiennych objaśniających (predyktorów). Analiza drzew klasyfikacyjnych jest jedną z podstawowych technik wykorzystywanych w tzw. Zgłębianiu danych (Data Mining). [7]

- Drzewa regresyjne znajdują szerokie zastosowanie w zadaniach związanych z poststratyfikacją, prognozowaniem oraz segmentacją. Są również bardzo użyteczną techniką eksploracji zbioru danych, odkrywania struktury związków pomiędzy zmiennymi i poszukiwania najlepszych predyktorów. Za pomocą czytelnych reguł, zwizualizowanych w formie przypominającej drzewo, zbiór danych dzielony jest na mniejsze segmenty o różniącej się od siebie średniej wartości zmiennej przewidywanej. Sprawia to, że drzewa regresyjne znakomicie sprawdzają się podczas szacowania wartości klienta, prognozowaniu wartości zakupów, czy też czasu spędzanego na stronie internetowej. Dzięki swoim właściwościom mogą stanowić rozszerzenie możliwości regresji liniowej czy też analizy wariancji. [8]

Aby rozwiązać badany problem komputerowego wspomagania diagnozowania białaczek u dzieci z wykorzystaniem drzewa decyzyjnego wykorzystano klasę `DecisionTreeClassifier` z biblioteki `scikit-learn`. Konstruktor tej klasy został pokazany na poniższym listingu:

```
class sklearn.tree.DecisionTreeClassifier(*, criterion='gini', splitter='best',
max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0,
max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0,
min_impurity_split=None, class_weight=None, ccp_alpha=0.0)
```

Listing 4: Konstruktor klasy DecisionTreeClassifier

`DecisionTreeClassifier` to klasa zdolna do wykonywania klasyfikacji wielu klas w zbiorze danych. Podobnie jak w przypadku innych klasyfikatorów, klasa `DecisionTreeClassifier` przyjmuje jako dane wejściowe dwie tablice: tablicę X w postaci `(n_samples, n_features)` przechowującą próbki uczące oraz tablicę Y wartości całkowitych, w postaci `(n_samples,)`, zawierającą etykiety klas dla próbek uczących.

3.3 Kryteria podziału drzewa

Drzewa decyzyjne mają postać struktury rozwijającej się w drzewo, gdzie każdy wewnętrzny węzeł (nie dotyczy liścia) oznacza test na atrybucie, każda gałąź drzewa reprezentuje wynik tego testu, natomiast każdy liść (końcowy element drzewa) zawiera przynależność do danej klasy.

Pierwszy węzeł drzewa nosi nazwę korzenia. Algorytm tworzy strukturę na podstawie wektorów wejściowych reprezentujących atrybuty wejściowe i decyzyjne oraz algorytmu określającego kryterium podziału. Po utworzeniu węzła, decyduje się o jego przeznaczeniu. Jeżeli dany węzeł posiada wektory danych tej samej klasy, wówczas oznaczany jest jako liść drzewa. W przeciwnym przypadku wyznaczone zostaje kryterium podziału oraz punkt podziału na podstawie obranego algorytmu podziału. Następnie tworzy się nowe węzły pod aktualnym.

Kryterium `Giniego` bazuje na pomiarze „zanieczyszczenia" partycji D, która zawiera zbiór wektorów danych wejściowych, co można przedstawić za pomocą poniższego wzoru:

$$G(D) = 1 - \sum_{i=1}^{m} p_i^2, \qquad (2)$$

p_i - prawdopodobieństwo obliczane według wzoru $|C_{i,d}|/|D|$
$C_{i,d}$ - zbiór próbek należących do i-klasy
D - zbiór wszystkich próbek w danym węźle

Dla każdego atrybutu testowane są wszystkie z możliwych podziałów. Dla atrybutów dyskretnych podzbiór, który uzyska najmniejszą wartość wskaźnika Giniego z przedziału [0,1]dla wybranego atrybutu jest wybrany jako kryterium podziału. W przypadku atrybutów ciągłych testuje się punkty podziału pomiędzy każdą parą posortowanych wartości atrybutu, dzięki któremu uzyskujemy punkt podziału.

Kryterium `entropii` bazuje na teorii informacji, która dotyczy zawartości informacyjnej w danych. Węzeł N zawiera wektory danych partycji D. Atrybut z największą zawartością informacyjną jest wybrany jako atrybut podziału dla węzła N i można przedstawić za pomocą poniższego wzoru:

$$I(D) = -\sum_{i=1}^{m} p_i \log_2(p_i), \qquad (3)$$

p_i - prawdopodobieństwo obliczane według wzoru $|C_{i,d}|/|D|$
$C_{i,d}$ - zbiór próbek należących do i-klasy
D - zbiór wszystkich próbek w danym węźle

`Entropia` to średnia ilość informacji, przypadająca na pojedynczą wiadomość ze źródła informacji. Innymi słowy jest to średnia ważona ilości informacji niesionej przez pojedynczą wiadomość, gdzie wagami są prawdopodobieństwa nadania poszczególnych wiadomości opisane poprzez powyższy wzór. [9]

3.4 Opis środowiska programistycznego

Poniższy kod programu przedstawiony na listingu przedstawia walidację krzyżową opisaną w podpunkcie 2.3. Jako poziom ufności alfa (α) przyjęliśmy wartość równą 0,05.

```
for i in range (len(X)):
    data.append([X[i][20], X[i][19], X[i][15], X[i][3], X[i][7],
    X[i][2], X[i][17], X[i][11], X[i][4]])
    ans.append(X[i][0])
dataTraining, dataTesting, ansTraining, ansTesting = train_test_split(data,
ans, test_size = 0.5, random_state=2)

for crit in criteria:
    print(crit)
    for num in depth:
        clf = tree.DecisionTreeClassifier(criterion=crit, max_depth=num)
        print('max dept = ' + str(num))

        clf = clf.fit(dataTraining, ansTraining)
        score = cross_val_score(estimator=clf, X=dataTesting, y=ansTesting,
        cv=5, n_jobs=4)
```

Listing 5:]Protokół eksperymentalny walidacji krzyżowej dwukrotnie powtórzonej 5 razy. [10]

Aby przeprowadzić testy statystyczne, wykorzystano również utworzone na podstawie zadanych parametrów z podpunktu 2.1 modele klasyfikacyjne.

```
clf = {
        '3gini': DecisionTreeClassifier(criterion='gini', max_depth=3),
        '5gini': DecisionTreeClassifier(criterion='gini', max_depth=5),
        '7gini': DecisionTreeClassifier(criterion='gini', max_depth=7),
        '3entropy': DecisionTreeClassifier(criterion='entropy', max_depth=3),
        '5entropy': DecisionTreeClassifier(criterion='entropy', max_depth=5),
        '7entropy': DecisionTreeClassifier(criterion='entropy', max_depth=7),
}
```

Listing 6:]Wyznaczone modele klasyfikacyjne. [10]

Obliczenie t-statystyki zostało zaimplementowane w oparciu o opisaną w podpunkcie 2.4 funkcję `ttest_ind`. Działa ona dla zbioru od 1 do 12 najlepszych cech, gdzie uzyskano najlepsze wyniki klasyfikacji.

```
for i in range (trait):
        for j in range (len(clf)):
                for k in range (len(clf)):
                        t[i][j][k], p[i][j][k]=ttest_ind(results[j+i*6],results[k+i*6])
```

Listing 7: .]Implementacja testu t-studenta [10].

Na poniższym listingu znajduje się implementacja tabel przewagi, istotności i obserwacji końcowych.

```
for index, t in enumerate(t):
        advantages[index][t > 0] = 1

for index, p_value in enumerate(p):
        importance[index][p_value <= alpha] = 1

for i in range(trait):
        betterStat.append(importance[i] * advantages[i])
```

Listing 8: .]Implementacja tabel przewagi, istotności i obserwacji końcowych [10].

4 Wyniki badań eksperymentalnych

4.1 Wyniki ewaluacji eksperymentalnej

4.1.1 Tabele wyników

W poniższej tabeli 5. znajdują się najlepsze uzyskane dokładności dla poszczególnych głębokości drzewa i typów kategoryzacji (Gini, Entropy). Najlepszy uzyskany wynik - 0,586 - osiągnięto dla klasyfikatora 7 gini i przy 12 cechach.

Tabela 5: Najlepsze uzyskane wyniki klasyfikacji

Klasyfikator		Najlepszy uzyskany wynik
Maksymalna głębokość drzewa	Typ kategoryzacji	
3	Gini	0,229
	Entropy	0,210
5	Gini	0,399
	Entropy	0,399
7	Gini	0,586
	Entropy	0,584

Poniższa tabela 6. przedstawia wyniki ewaluacji eksperymentalnej dla każdego z następujących klasyfikatorów: 3 gini, 5 gini, 7 gini, 3 entropy, 5 entropy i 7 entropy. Każdy wiersz zawiera dane uzyskane podczas eksperymentu, czyli liczbę cech wykorzystanych podczas konkretnego badania ustalonych na podstawie rankingu cech z punktu `1.5` oraz średnią wartość klasyfikacji (`Accuracy`).

Tabela 6: Wyniki klasyfikacji dla poszczególnej liczby cech i klasyfikatorów

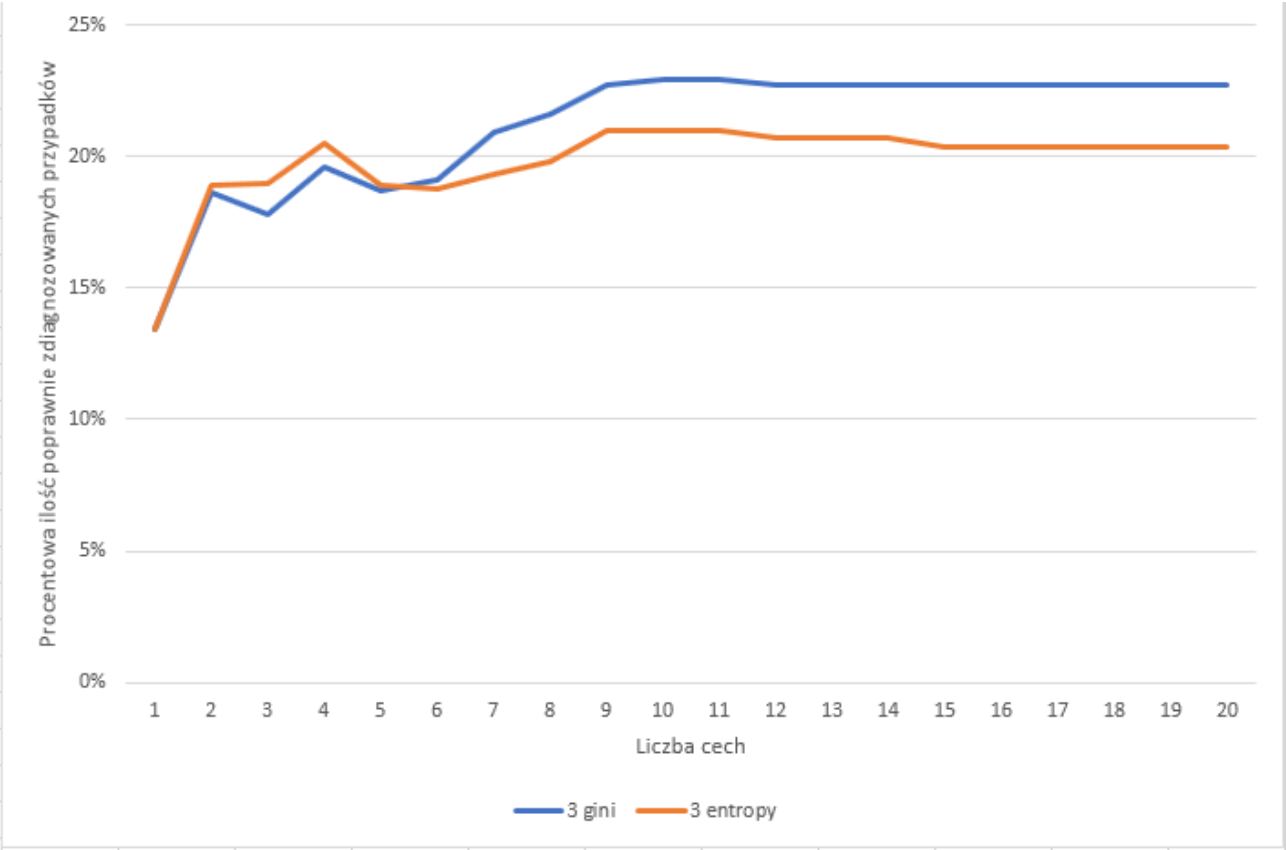
	Głębokość drzewa i kryterium podziału					
Liczba cech	3 gini	5 gini	7 gini	3 entropy	5 entropy	7 entropy
1	0,134	0,134	0,134	0,134	0,134	0,134
2	0,186	0,192	0,192	0,189	0,192	0,192
3	0,178	0,248	0,260	0,190	0,253	0,260
4	0,196	0,291	0,361	0,205	0,314	0,373
5	0,187	0,325	0,428	0,189	0,334	0,434
6	0,191	0,313	0,451	0,188	0,318	0,439
7	0,209	0,362	0,521	0,193	0,358	0,499
8	0,216	0,374	0,544	0,198	0,363	0,536
9	0,227	0,387	0,565	0,210	0,385	0,557
10	0,229	0,383	0,575	0,210	0,399	0,561
11	0,229	0,383	0,574	0,210	0,399	0,560
12	0,227	0,398	0,586	0,207	0,389	0,584
13	0,227	0,394	0,560	0,207	0,388	0,567
14	0,227	0,393	0,568	0,207	0,387	0,566
15	0,227	0,383	0,561	0,204	0,378	0,562
16	0,227	0,381	0,565	0,204	0,378	0,565
17	0,227	0,394	0,577	0,204	0,391	0,583
18	0,227	0,394	0,577	0,204	0,391	0,577
19	0,227	0,395	0,573	0,204	0,391	0,576
20	0,227	0,399	0,574	0,204	0,384	0,583

Tabela 7: Wartości odchyleń standardowych dla poszczególnej liczby cech i klasyfikatorów

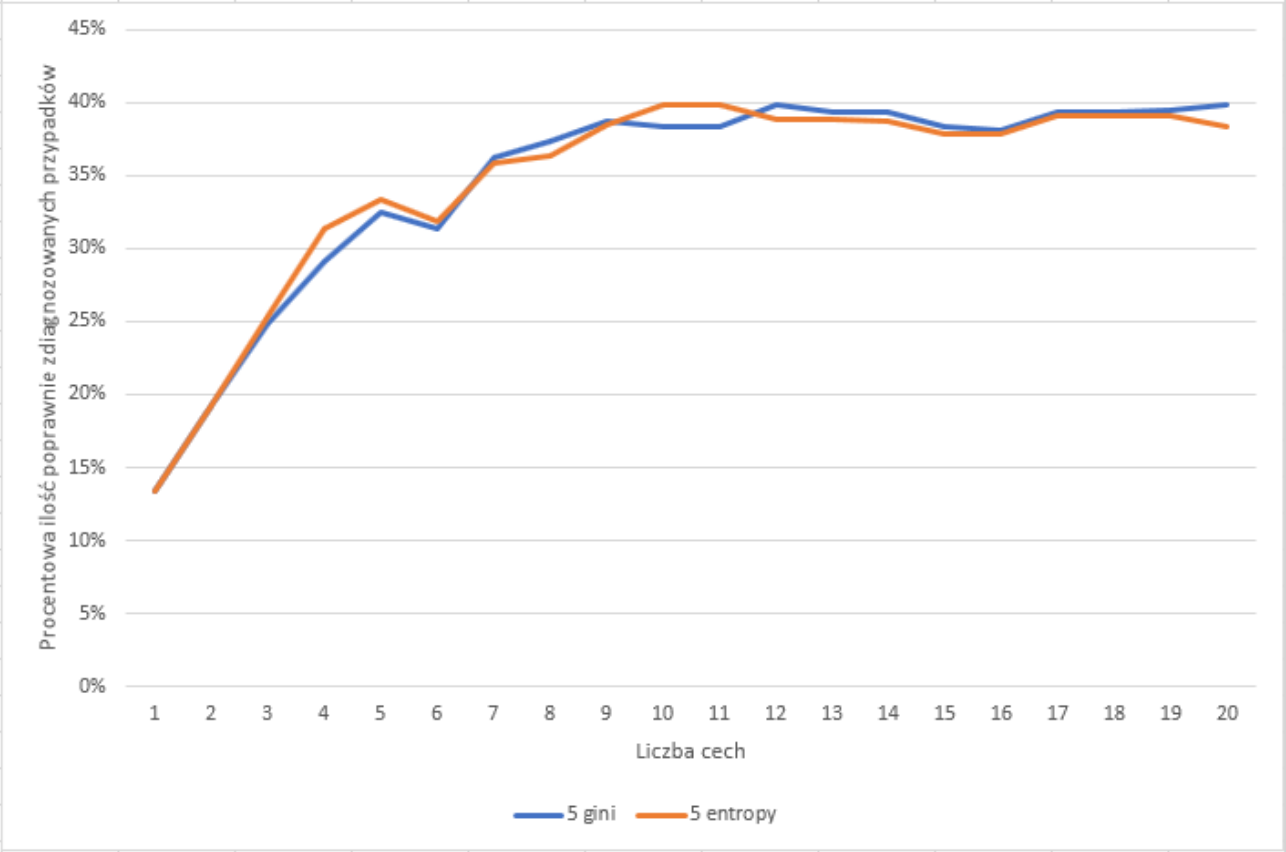
	Głębokość drzewa i kryterium podziału					
Liczba cech	3 gini	5 gini	7 gini	3 entropy	5 entropy	7 entropy
1	0,03	0,08	0,14	0,02	0,07	0,14
2	0,02	0,06	0,12	0,01	0,05	0,11
3	0,05	0,08	0,12	0,04	0,08	0,12
4	0,06	0,10	0,14	0,06	0,10	0,14
5	0,07	0,12	0,16	0,07	0,12	0,16
6	0,08	0,13	0,18	0,07	0,13	0,18
7	0,08	0,14	0,19	0,08	0,14	0,19
8	0,08	0,14	0,20	0,08	0,14	0,20
9	0,09	0,15	0,21	0,08	0,14	0,21
10	0,09	0,15	0,21	0,08	0,15	0,21
11	0,09	0,15	0,21	0,08	0,15	0,21
12	0,09	0,15	0,21	0,08	0,14	0,21
13	0,08	0,14	0,20	0,08	0,14	0,21
14	0,08	0,14	0,20	0,07	0,14	0,20
15	0,08	0,13	0,19	0,07	0,13	0,19
16	0,07	0,13	0,18	0,07	0,12	0,18
17	0,07	0,12	0,17	0,06	0,12	0,17
18	0,06	0,11	0,15	0,06	0,10	0,15
19	0,05	0,09	0,13	0,05	0,09	0,13
20	0,04	0,07	0,10	0,04	0,07	0,10

4.1.2 Wykresy

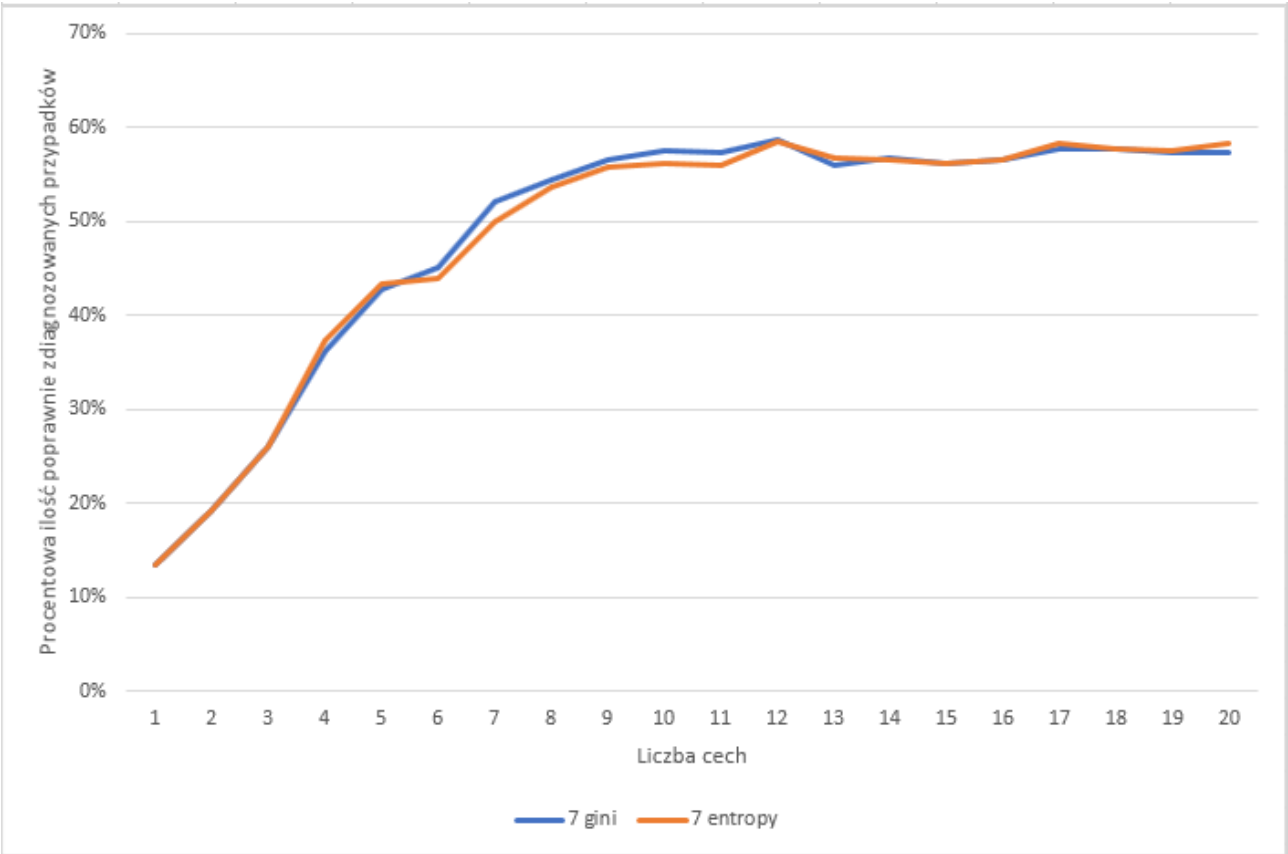
Wyniki z tabeli 6. zostały również zaprezentowane za pomocą trzech wykresów. Poniższe wykresy przedstawiają zależność pomiędzy liczbą cech wykorzystanych podczas konkretnego badania ustalonych na podstawie rankingu cech z punktu `1.5` oraz średnią wartość klasyfikacji (`Accuracy`).



Rysunek 8: Wyniki badań dla klasyfikatorów 3 gini i 3 entropy.



Rysunek 9: Wyniki badań dla klasyfikatorów 5 gini i 5 entropy.



Rysunek 10: Wyniki badań dla klasyfikatorów 7 gini i 7 entropy.

4.2 Wyniki testów statystycznych

Poniższa tabela 8. zawiera informacje na temat tego, który klasyfikator z pary klasyfikatorów osiągnął lepszy wynik klasyfikacji. Liczba 0 oznacza, że klasyfikator z pierwszej kolumny jest statystycznie porównywalny lub gorszy od klasyfikatora z pierwszego wiersza, a liczba 1 oznacza, że jest on lepszy statystycznie.

Tabela 8: Tabela przewagi

	3 gini	5 gini	7 gini	3 entropy	5 entropy	7 entropy
3 gini	0	0	0	1	0	0
5 gini	1	0	0	1	0	0
7 gini	1	1	0	1	1	0
3 entropy	0	0	0	0	0	0
5 entropy	1	0	0	1	0	0
7 entropy	1	1	0	1	1	0

Poniższa tabela 9. zawiera informacje o tym, czy różnica pomiędzy zadaną parą klasyfikatorów jest istotna statystycznie. Liczba 0 oznacza, że różnica między klasyfikatorem z pierwszej kolumny a klasyfikatorem z pierwszego wiersza nie jest istotna, a liczba 1 oznacza, że jest ona istotna statystycznie.

Tabela 9: Tabela istotności

	3 gini	5 gini	7 gini	3 entropy	5 entropy	7 entropy
3 gini	0	1	1	1	1	1
5 gini	1	0	1	1	0	1
7 gini	1	1	0	1	1	0
3 entropy	1	1	1	0	1	1
5 entropy	1	0	0	1	0	0
7 entropy	1	0	0	1	1	0

Poniższa tabela 10. zawiera wyniki końcowe dla najlepszych 12 cech, która pokazuje dla każdego z klasyfikatorów ten klasyfikator, od którego jest on statystycznie znacząco lepszy. Wartości w tej tabeli są iloczynami wartości poszczególnych komórek z tabel przewagi i istotności.

Tabela 10: Tabela wyników końcowych

	3 gini	5 gini	7 gini	3 entropy	5 entropy	7 entropy
3 gini	0	0	0	1	0	0
5 gini	1	0	0	1	0	0
7 gini	1	1	0	1	1	0
3 entropy	0	0	0	0	0	0
5 entropy	1	0	0	1	0	0
7 entropy	1	0	0	1	1	0

4.3 Dyskusja otrzymanych wyników

Na podstawie wykonanych badań oraz analizy ich wyników wyłoniony został zestaw cech, które pozwalają na uzyskanie najwyżej ze statystycznego punktu widzenia skuteczności klasyfikatora algorytmu `CART` dla problemu wspomagania diagnozowania białaczek u dzieci:

- Typ kategoryzacji - `GINI`
- Liczba najistotniejszych cech z rankingu cech - `12`
- Głębokość drzewa - `7`

Przedstawiony zestaw parametrów umożliwił uzyskanie średniej skuteczności na poziomie 58,6%.
Algorytm `CART` umożliwił rozwiązanie problemu diagnozowania białaczek u dzieci z całkiem wysoką skutecznością, pomimo tego, że nie posiadaliśmy zbyt wielu danych do analizy (20 klas, 410 zdiagnozowanych przypadków).

Dla małej głębokości drzewa (wartość 3) lepszą jakość klasyfikacji uzyskał klasyfikator z wykorzystaniem kryterium `Gini` o około 2,5%. Dla głębokości drzewa `5` nie widać znaczących różnych pomiędzy wynikami dla dwóch różnych kryteriów podziału drzewa, z wyjątkiem sytuacji, gdzie klasyfikator `Entropy` uzyskał lepsze wyniki klasyfikacji dla liczby cech z przedziału [9, 12] o około 1,5%.

Dla największej wartości głębokości drzewa `7` nie widać znaczących różnic w przebiegu wykresów. Dla tej głębokości drzewa uzyskano najlepszą wartość w całym eksperymencie dla 12 najlepszych cech dla kryterium `Gini` (0,586), jak i kryterium `Entropy` (0,584).

Dla każdego kryterium i głębokości drzewa stabilizacja otrzymywanych wyników klasyfikacji następuje przy 9 najlepszych cechach z rankingu cech. Oznacza to, że przy tej liczbie dodawanie kolejnych cech nie polepsza działania algorytmu, a jedynie utrzymuje na mniej więcej jednakowym poziomie.

Podczas działania algorytmu obliczenia dla małej liczby cech (z przedziału [1,7]) były znacznie szybsze niż dla dużej liczby cech (z przedziału [15,20]). Wynika to z tego, że algorytm ma więcej danych do nauki.

5 Podsumowanie i wnioski

Celem realizowanego przez nas projektu było nabycie umiejętności zastosowania algorytmu klasyfikacji nadzorowanej (w naszym przypadku algorytmu drzewa decyzyjnego) w zadaniu diagnozowania białaczek u dzieci. Ten proces wymagał od nas odpowiedniego wybrania cech, dzięki którym wybrany algorytm mógł rozpoznać chorobę u pacjenta. Dzięki danym rzeczywistym oceniliśmy jakość klasyfikacji i sprawdziliśmy, w jaki sposób ta jakość zależy od liczby atrybutów wykorzystanych do skonstruowania modelu klasyfikacyjnego.

Udało nam się zrealizować wszystkie założenia z podpunktu 1.1. W trakcie wykonywania algorytmu warto zastosować funkcję mierzącą czas jego wykonywania, której wyniki działania pozwoliłyby na stworzenie tabeli i wykresów przedstawiających szybkość pracy algorytmu dla różnej liczby cech i kryteriów podziału drzewa.

Literatura

[1] *http://gdudek.el.pcz.pl/index.php/zainteresowania-naukowe/10-selekcja-cech*, Selekcja cech.

[2] *Ocena testów sprawdzających wiedzę studenta metodami testowania hipotez statystycznych. modelowanie metodą chi-kwadrat*, Rajs R., Krosno, 2006.

[3] *Statystyka praktyczna*, Starzyńska W., wyd. PWN, Warszawa, 2000.

[4] *https://www.ii.pwr.edu.pl/ zieba/Lista5.pdf*

[5] *https://mfiles.pl/pl/index.php/Drzewo_decyzyjne*, Drzewo decyzyjne.

[6] *http://fizyka.umk.pl/publications/kmk/Prace-Mgr/08-Wilczewski-InTrees-drzewa-decyzji.pdf*

[7] *https://www.statsoft.pl/textbook/stathome.html*

[8] *https://predictivesolutions.pl/wykorzystanie-drzew-regresyjnych-do-analizy-wartosci-zakupow-cz-2*

[9] *Wykorzystanie drzew decyzyjnych oraz ekstrakcji reguł do klasyfikacji regułowej podatników*, Budziński R., Misztal L., Szczecin, 2010.

[10] *https://metsi.github.io/2020/04/03/kod4.html*, Parowe testy statystyczne.