Politechnika Wrocławska

Indukcyjne Metody Analizy Danych

Ćwiczenie 2 - Indukcja drzew decyzyjnych C4.5 w R

Marcel Cielinski
Index: 236747

prowadzący
dr inż. Paweł Myszkowski

23 kwietnia 2020

Wprowadzenie

1.1 Problem

Celem ćwiczenia było poznanie wybranego algorytmu drzewa decyzyjnego, jakim jest *C4.5*. Do składowych zadania zalicza się zapoznanie się z platformą *R*, implementację wspomnianego algorytmu w tym języku oraz przeprowadzenie badań na trzech, ustalonych na potrzeby ćwiczenia, zbiorach danych. Należało przetestować różne ustawienia parametrów samego algorytmu oraz zbadać wpływ użycia dwóch rodzajów walidacji krzyżowej do oceny skuteczności algorytmu poprzez obserwację metryk mówiących o jakości klasyfikatora.

1.2 Drzewa decyzyjne

Drzewo decyzyjne jest klasyfikatorem, którego zadaniem jest rekurencyjne dzielenie danych na kolejne podzbiory, wykorzystując odpowiednie mechanizmy. Drzewa zawierają się w grupie uczenia z nadzorem (ang. supervised learning) i mogą być używane zarówno dla danych dyskretnych, jak i ciągłych. Są one jednymi z najczęściej wykorzystywanych technik analizy danych. Przechodząc od korzenia drzewa (pierwszego i głównego węzła) do jednego z jego liści (wybierając odpowiednie decyzje w węzłach decyzyjnych), otrzymywana jest klasa, do której algorytm przyporządkowuje. Drzewa decyzyjne charakteryzują się następującymi własnościami:

- W każdym węźle drzewa umieszczony jest jeden z atrybutów.
- Każda krawędź wychodząca z danego węzła jest etykietowana jedną z możliwych wartości atrybutu ojca.
- Liściem w takim drzewie jest wartość ze zbioru kategorii, jaką przyporządkujemy rekordom mającym takie wartości, jakie znajdują się na ścieżce od korzenia do liścia.
- Na każdym poziomie w drzewie mogą się znajdować zarówno węzły z atrybutami, jak i liście.

Istotnym jest również rozmieszczenie atrybutów w węzłach. W korzeniu umieszczany jest ten, który zwraca największy zysk informacji - dzieli dane najczyściej spośród wszystkich możliwych podziałów ze względu na pojedynczy atrybut. Reguła ta obowiązuje dla każdego kolejno budowanego poddrzewa.

Miarą określającą nieuporządkowanie danych jest entropia. Znając rozkład prawdopodobieństwa wybrania obiektu danej klasy $P = (p_1, p_2, \ldots, p_n)$, możemy ją wyznaczyć:

$$Entropy(P) = -\sum_{i=1}^{n} p_i \cdot log_2\, p_i \tag{1}$$

Natomiast korzystając z entropii, można wyznaczyć zysk informacji:

$$InfoGain(X, A) = Entropy(X) - \sum_{i=1}^{n} \frac{|X_i|}{|X|} \cdot Entropy(X_i) \tag{2}$$

, gdzie X jest podzbiorem rekordów, a A to wybrany atrybut do podziału.

Najpopularniejsze algorytmy oparte o budowę drzewa decyzyjnego to: *ID3*, *CART*, *C4.5* oraz *C5.0*. Najprostszy z nich, *ID3*, obsługuje jedynie dane o atrybutach o wartościach dyskretnych. Do niepożądanych cech tego algorytmu należy także zaliczyć ograniczenie jakim jest nie radzenie sobie z brakującymi wartościami oraz oraz może zbytnio się rozrastać - może to powodować przeuczanie się. Sposobem na wszystkie te ograniczenia jest stosowanie na przykład *C4.5*, co jest przedmiotem referatu.

1.2.1 C4.5

Algorytm *C4.5* jest rozszerzeniem algorytmu *ID3*, wychodzącym naprzeciw problemom tam spotykanym. Czyli:

- wartości ciągłe - atrybuty o typie wartości ciągłych - rekordy ustawiane są w kolejności rosnącej ze względu na ten atrybut, kolejno wybierany jest próg, który podzieli dane - na mniejsze od progu oraz większe
- brakujące dane - poprzez wyliczenie prawdopodobieństwa możliwych wyników
- rozrastanie się drzewa - aby tego uniknąć stosuje się tzw. przycinanie (ang. pruning), w celu zwiększenia generalizacji oceny. Metoda działa od strony liścia, w górę. Heurystycznie sprawdzany jest wpływ na wartość przewidywanego błędu przycięcie - czyli zastąpienie poddrzewa pojedynczym liściem z kategorią najpopularniejszą wśród jego liści

Implementacja

Do implementacji wybranego algorytmu drzew decyzyjnych, jakim jest *C4.5* użyto biblioteki uczenia maszynowego *RWeka* dla języka *R*. Do stworzenia *C4.5* wykorzystywany jest *J48*. Przetestowany zostanie wpływ ustawiania parametrów takich jak minimalna liczba instancji na liść (M) oraz miara pewności przycinania (C) na metryki oceniające jakość modelu.

2.1 Ocena jakości

Do oceny jakości klasyfikatora stosuje się metryki związane, bądź wynikające z macierzy błędów (ang. Confusion matrix). Po wykonaniu predykcji dopasowania etykiet (klas), możliwe jest zbadanie prawidłowości dopasowań. Właśnie w tym celu posługujemy się tablicą pomyłek, która zestawia liczebność instancji prawidłowo i nieprawidłowo oznaczonych przez klasyfikator. Podstawowe takie metryki to:

- *Accuracy* - dokładność - jak dobre są ogólne predykcje klasyfikatora. Jest to stosunek dobrze oznaczonych klas do wszystkich oznaczeń.
- *Precission* - precyzja - mówi o precyzyjności klasyfikatora. Jest to zdolność klasyfikatora do nie oznaczania negatywnych próbek jako pozytywne.
- *Recall* - czułość - jak dobrze klasyfikator radzi sobie ze znajdywaniem pozytywnych próbek. Jest to stosunek próbek dobrze zaklasyfikowanych jako pozytywne przez wszystkie rzeczywiście pozytywne.

- *F1-score* - oznaczane jako *FSC* - jest to w istocie średnia harmoniczna czułości oraz precyzji.

Do oceny eksperymentów, szczególną uwagę powinno zwracać się na *FSC*. Jest to często wykorzystywana miara, która pozwala na wymiernie dobre porównanie różnych podejść. W implementacji wykorzystany został pakiet *MLmetrics*, który udostępnia wszystkie powyżej wymienione metryki.

2.2 Kroswalidacja

Kroswalidacja (sprawdzian krzyżowy) jest stosowana w celu lepszej oceny działania modeli, na przykład takich jak drzewa decyzyjne. Procedura determinuje podział danych na treningowe i testowe. A jej wykorzystanie tłumaczy się unikaniem overfittingu.

W realizacji zadania wykorzystano metody, dostępne w pakiecie *caret*. Implementują one:

- kroswalidację - gdzie całkowity zbiór danych jest dzielony na K równolicznych podzbiorów (parametr *n_splits*), z których kolejno każdy jest traktowany jako zbiór testowy, kiedy model trenowany jest na połączonych (*K-1*) pozostałych podzbiorach. Otrzymuje się wówczas K wyników, z których następnie liczona jest średnia.
- kroswalidację stratyfikowaną - jej działanie jest analogiczne do powyżej opisanego, wariantu zwykłego. Różnica polega na tym, że całkowity zbiór jest dzielony (na K części, także parametrem *n_splits*) w miarę możliwości na podzbiory o proporcjonalnym rozkładzie klas - zgodnie z proporcją istniejącą w całościowym zbiorze.

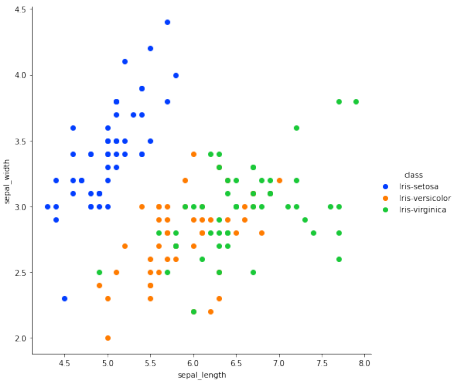
Przed przystąpieniem do badań, zastosowano przemieszanie danych (przy użyciu instrukcji $sample(nrow(dataset))$). Metoda ta jest wprowadzana w celu zapewnienia wstępnego losowego rozłożenia rekordów (często zbiory danych wejściowych są sortowane po klasach). Testowany był także wpływ ilości foldów (K) na otrzymywane wyniki, dla obu metod.

2.3 Zbiory danych

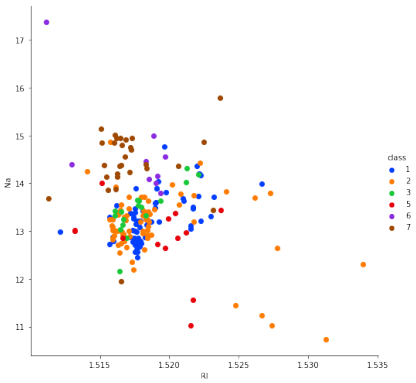
Do walidacji zaimplementowanych funkcjonalności wykorzystano łącznie cztery zbiory danych. Jeden w celu sprawdzenia działania i trzy pozostałe w celach testów i przeprowadzenia badań. Te zbiory, to:

- *iris* - gdzie liczba atrybutów to 4, a klas 3. Zbiór przedstawia odmiany kwiatów Iris, a atrybuty odpowiadają ich cechom.
- *glass* - gdzie liczba atrybutów to 9, a klas 7. Zbiór przedstawia typy szkła, gdzie atrybuty opisują skład chemiczny.
- *wine* - gdzie liczba atrybutów to 13, a klas 3. Zbiór ten zawiera dane o analizie chemicznej win z tego samego regionu, jednak z 3 różnych upraw.
- *seeds* - gdzie liczba atrybutów wynosi 7, a klas 3. Zbiór ten zawiera miary geometrycznych własności 3 różnych odmian ziaren pszenicy.

Na Rysunkach 1, 2, 3, 4 przedstawiono przykładowe rozkłady klas względem wybranych atrybutów.



Rysunek 1: Iris



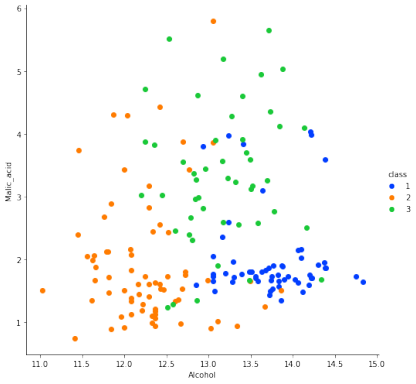
Rysunek 2: Glass

Klasa	Instancje	% rekordów
Setosa	50	33 (%)
Versicolor	50	33 (%)
Virginica	50	33 (%)

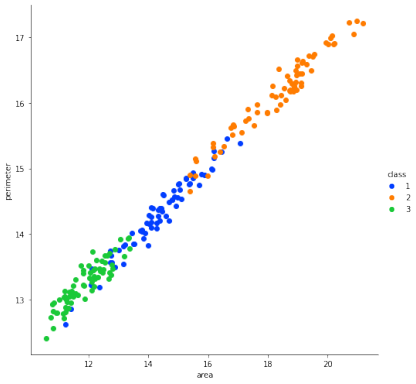
Tablica 1: Klasy zbioru *Iris*

Klasa	Instancje	% rekordów
1	70	33 (%)
2	76	36 (%)
3	17	8 (%)
4	0	0 (%)
5	13	6 (%)
6	9	4 (%)
7	29	13 (%)

Tablica 2: Klasy zbioru *Glass*



Rysunek 3: Wine



Rysunek 4: Seeds

Klasa	Instancje	% rekordów
1	59	33 (%)
2	71	40 (%)
3	48	27 (%)

Tablica 3: Klasy zbioru *Wine*

Klasa	Instancje	% rekordów
1	70	33 (%)
2	70	33 (%)
3	70	33 (%)

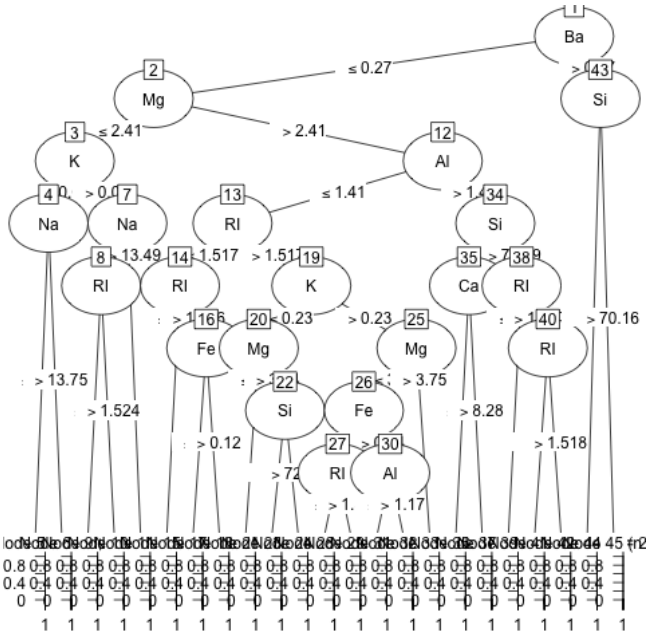
Tablica 4: Klasy zbioru *Seeds*

Wyniki eksperymentów

3.1 Zbiór Glass

Poniższa tabela (Tabela 5) przedstawia wszystkie przeprowadzone testy na zbiorze *Glass*, z podziałem na typ kroswalidacji, wielkość parametru K oraz wielkości parametrów algorytmu *C4.5*: C i M. Możemy wywnioskować, że dla kroswalidacji (zarówno zwykłej jak i stratyfikowanej), lepsze wyniki otrzymujemy dla wielkości parametru $K = 5$ oraz 15. Obserwujemy także, że mniejsza wartość parametru M zwykle przekłada się na wyższe wskazania metryk (choć nie zawsze). Parametr C także ma wpływ na rezultaty. Dodatkową obserwacją jest fakt, że algorytm drzewa decyzyjnego lepiej radzi sobie z testowanym zbiorem, niż zaimplementowany w poprzednim zadaniu - klasyfikator oparty o twierdzenie Bayesa.

Na rysunku (Rysunek 5) przedstawione zostało wygenerowane drzewo dla tego zbioru. Możemy odczytać, że algorytm zbudował stosunkowo dużą strukturę (nie jest szczególnie czytelna), w której używane są wszystkie atrybuty, jak również występują ich powtórzenia w węzłach w ramach konkretnych ścieżek.



Rysunek 5: Wygenerowane drzewo dla zbioru *Glass*

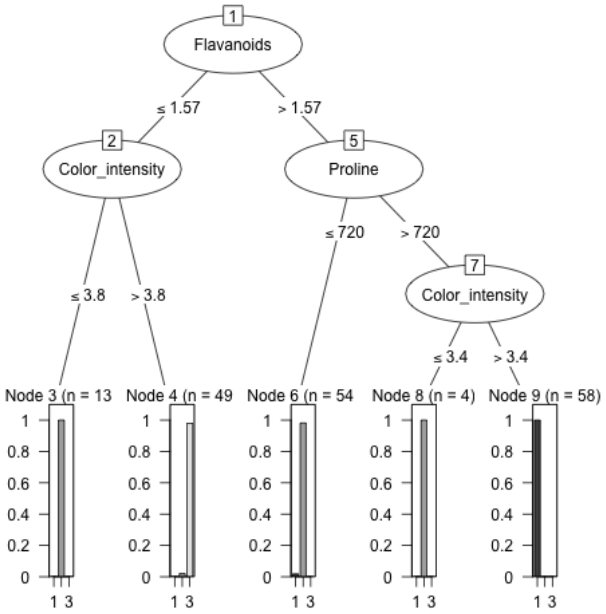
MultinominalNB	smoothing	0.52	0.68	0.60	0.50
C	M	ACC	PREC	REC	FSC
K = 5					
0.10	2.00	0.67	0.72	0.74	0.72
0.10	15.00	0.58	0.58	0.73	0.64
0.25	2.00	0.65	0.68	0.71	0.69
0.25	15.00	0.58	0.58	0.73	0.64
0.50	2.00	0.64	0.67	0.71	0.68
0.50	15.00	0.58	0.58	0.70	0.63
K = 10					
0.10	2.00	0.66	0.66	0.69	0.67
0.10	15.00	0.58	0.52	0.59	0.54
0.25	2.00	0.67	0.67	0.69	0.67
0.25	15.00	0.59	0.54	0.54	0.54
0.50	2.00	0.66	0.67	0.69	0.68
0.50	15.00	0.60	0.56	0.56	0.55
K = 15					
0.10	2.00	0.68	0.73	0.73	0.71
0.10	15.00	0.67	0.65	0.76	0.69
0.25	2.00	0.67	0.72	0.71	0.70
0.25	15.00	0.67	0.63	0.74	0.72
0.50	2.00	0.67	0.72	0.71	0.70
0.50	15.00	0.66	0.67	0.66	0.68
Stratified K = 5					
0.10	2.00	0.67	0.67	0.77	0.72
0.10	15.00	0.60	0.63	0.77	0.68
0.25	2.00	0.68	0.68	0.80	0.74
0.25	15.00	0.60	0.63	0.77	0.68
0.50	2.00	0.68	0.68	0.80	0.74
0.50	15.00	0.60	0.64	0.73	0.67
Stratified K = 15					
0.10	2.00	0.69	0.72	0.72	0.71
0.10	15.00	0.67	0.68	0.73	0.69
0.25	2.00	0.68	0.72	0.72	0.71
0.25	15.00	0.66	0.66	0.71	0.67
0.50	2.00	0.69	0.72	0.72	0.71
0.50	15.00	0.67	0.70	0.69	0.68

Tablica 5: Wpływ kroswalidacji na metryki dla zbioru *Glass*

3.2 Zbiór Wine

Poniższa tabela (Tabela 6) przedstawia wszystkie przeprowadzone testy na zbiorze *Wine*, z podziałem na typ kroswalidacji, wielkość parametru K oraz wielkości parametrów algorytmu *C4.5*: C i M. W tym przypadku, możemy wywnioskować, że dla kroswalidacji (zarówno zwykłej jak i stratyfikowanej), najlepsze wyniki otrzymujemy dla $K = 15$, jednak dla pozostałych również nie są złe. Podobnie jak dla zbioru *Glass*, obserwujemy, że mniejsza wartość parametru M przekłada się na wyższe wskazania metryk.

Na rysunku (Rysunek 6) przedstawione zostało wygenerowane drzewo dla tego zbioru. W tym przypadku także możemy odczytać, że algorytm zbudował stosunkowo niewielką strukturę, w której wykorzystywane są 3 atrybuty.



Rysunek 6: Wygenerowane drzewo dla zbioru *Wine*

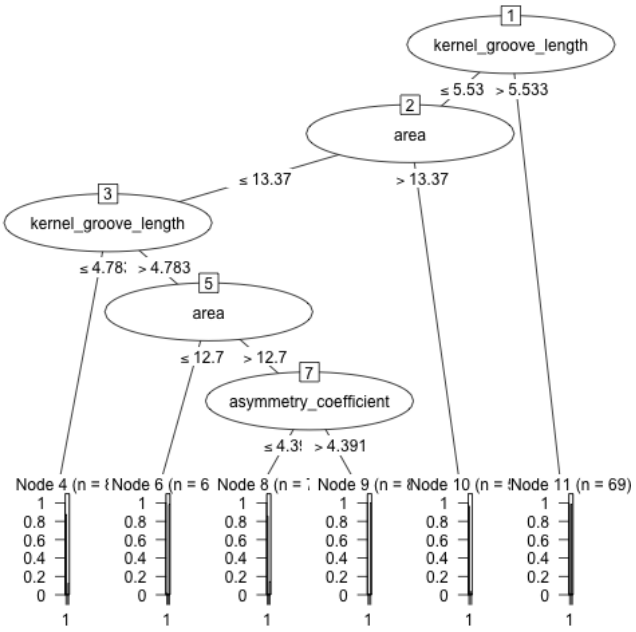
GaussianNB	smoothing	0.94	0.94	0.95	0.95
C	M	ACC	PREC	REC	FSC
K = 5					
0.10	2.00	0.94	0.98	0.93	0.95
0.10	15.00	0.91	0.93	0.97	0.95
0.25	2.00	0.94	0.98	0.93	0.95
0.25	15.00	0.91	0.93	0.97	0.95
0.50	2.00	0.94	0.98	0.93	0.95
0.50	15.00	0.91	0.93	0.97	0.95
K = 10					
0.10	2.00	0.92	0.94	0.98	0.96
0.10	15.00	0.90	0.93	0.98	0.95
0.25	2.00	0.92	0.94	0.98	0.96
0.25	15.00	0.90	0.93	0.98	0.95
0.50	2.00	0.92	0.94	0.98	0.96
0.50	15.00	0.90	0.93	0.98	0.95
K = 15					
0.10	2.00	0.94	0.97	0.97	0.97
0.10	15.00	0.86	0.89	0.95	0.90
0.25	2.00	0.94	0.97	0.97	0.97
0.25	15.00	0.86	0.89	0.95	0.90
0.50	2.00	0.94	0.97	0.97	0.97
0.50	15.00	0.86	0.89	0.95	0.90
Stratified K = 5					
0.10	2.00	0.93	0.94	0.97	0.95
0.10	15.00	0.85	0.88	0.88	0.87
0.25	2.00	0.93	0.94	0.97	0.95
0.25	15.00	0.85	0.88	0.88	0.87
0.50	2.00	0.93	0.94	0.97	0.95
0.50	15.00	0.85	0.88	0.88	0.87
Stratified K = 15					
0.10	2.00	0.94	0.96	0.98	0.97
0.10	15.00	0.89	0.87	0.97	0.91
0.25	2.00	0.94	0.96	0.98	0.97
0.25	15.00	0.89	0.87	0.97	0.91
0.50	2.00	0.94	0.96	0.98	0.97
0.50	15.00	0.89	0.87	0.97	0.91

Tablica 6: Wpływ kroswalidacji na metryki dla zbioru *Wine*

3.3 Zbiór Seeds

Poniższa tabela (Tabela 7) przedstawia wszystkie przeprowadzone testy na zbiorze *Seeds*, z podziałem na typ kroswalidacji, wielkość parametru K oraz wielkości parametrów algorytmu *C4.5*: C i M. W tym przypadku, możemy wywnioskować, że dla kroswalidacji zwykłej - optymalną wielkością parametru K jest 10, natomiast dla odmiany stratyfikowanej - 15. Na przykładzie tego zbioru, obserwujemy wpływ wartości obu testowanych parametrów algorytmu *C4.5* na wskazania metryk.

Na rysunku (Rysunek 7) przedstawione zostało wygenerowane drzewo dla tego zbioru. Algorytm zbudował stosunkowo niewielką strukturę, w której wykorzystywane są 3 atrybuty.



Rysunek 7: Wygenerowane drzewo dla zbioru *Seeds*

GaussianNB	smoothing	0.89	0.90	0.90	0.90
C	M	ACC	PREC	REC	FSC
K = 5					
0.10	2.00	0.92	0.93	0.84	0.87
0.10	15.00	0.90	0.92	0.77	0.83
0.25	2.00	0.92	0.93	0.84	0.87
0.25	15.00	0.90	0.89	0.80	0.83
0.50	2.00	0.92	0.93	0.84	0.87
0.50	15.00	0.90	0.89	0.80	0.83
K = 10					
0.10	2.00	0.95	0.92	0.93	0.92
0.10	15.00	0.88	0.83	0.80	0.81
0.25	2.00	0.94	0.92	0.90	0.91
0.25	15.00	0.88	0.83	0.80	0.81
0.50	2.00	0.94	0.92	0.90	0.91
0.50	15.00	0.88	0.83	0.83	0.82
K = 15					
0.10	2.00	0.92	0.88	0.92	0.89
0.10	15.00	0.88	0.87	0.79	0.79
0.25	2.00	0.92	0.88	0.90	0.88
0.25	15.00	0.89	0.87	0.84	0.84
0.50	2.00	0.92	0.88	0.90	0.88
0.50	15.00	0.89	0.87	0.84	0.84
Stratified K = 5					
0.10	2.00	0.91	0.87	0.87	0.86
0.10	15.00	0.89	0.88	0.80	0.83
0.25	2.00	0.91	0.87	0.87	0.86
0.25	15.00	0.90	0.88	0.81	0.83
0.50	2.00	0.91	0.87	0.87	0.86
0.50	15.00	0.89	0.86	0.83	0.83
Stratified K = 15					
0.10	2.00	0.93	0.89	0.92	0.90
0.10	15.00	0.88	0.84	0.83	0.81
0.25	2.00	0.92	0.88	0.89	0.88
0.25	15.00	0.88	0.82	0.84	0.82
0.50	2.00	0.92	0.88	0.89	0.88
0.50	15.00	0.88	0.82	0.84	0.82

Tablica 7: Wpływ kroswalidacji na metryki dla zbioru *Seeds*

Podsumowanie

Zadanie pozwoliło na zaznajomienie się nie tylko z algorytmem drzewa decyzyjnego *C4.5*, ale także z wieloma narzędziami służącymi pracy na danych oraz, ogólnie rzecz ujmując, platformą języka *R*, która jak wiadomo jet powszechnie wykorzystywana do obliczeń statystycznych. W porównaniu z językiem *Python*, środowisko to zdaje się być bardzo dobrze przygotowane pod problemy bazujące na danych. Wiele elementów z tym związanych jest uprosz-

czonych, jak również dostępne biblioteki pozwalają na implementację złożonych funkcjonalności niewielkim nakładem pracy (a przynajmniej kodu). Sprawnie rozwiązany jest także aspekt wizualizacji danych oraz wyświetlania różnego rodzaju wykresów, z koeli sposób wykonywania poleceń także służy temu, do czego środowisko to zostało stworzone.

W ogólności, drzewa decyzyjne, jako struktura, są bardzo czytelne dla człowieka. W łatwy sposób można odtworzyć proces przyporządkowywania pojedynczej instancji, wskazanej klasy. Wystarczy jedynie wybierać odpowiednie decyzje, zaczynając od korzenia, a algorytm sam doprowadzi nas do wybranego liścia. Co wynika z przeprowadzonych eksperymentów, *C4.5* dobrze poradził sobie z wybranymi na potrzeby zadania zbiorami. Dla konkretnych ustawień parametrów, w każdym przypadku można zaobserwować chociaż nieznaczne polepszenie otrzymywanych metryk, w zestawieniu z najlepszym wariantem naiwnego klasyfikatora Bayesowskiego, z poprzedniego zadania.