Ćwiczenie nr 4
Wprowadzenie do sztucznej inteligencji

Autor
Maciej Prostak

1 Wprowadzenie

Tematem zadania jest klasyfikacja za pomocą zaimplementowanego drzewa decyzyjnego przy pomocy algorytmu ID3. Po odpowiednim zbudowaniu drzewa następne zostaną przeprowadzone próby za pomocą metody k-krotnej walidacji krzyżowej.

1.1 Technologia

Rozwiązanie zostało zaimplementowane w języku Python (3.9.0) z wykorzystaniem biblioteki pandas (1.3.4) oraz numpy (1.12.4). Dodatkowo dla poprawy czytelności kodu zostały dołączone biblioteki black (21.9b0), flake8 (4.0.1) oraz isort (5.9.3).

1.2 Zbiór danych

Dane składają się z 6 atrybutów (”buying”, ”maint”, ”doors”, ”persons”, ”lugboot”, ”safety”) i ostatecznej przydzielonej klasy ”class”.

1.3 Walidacja k-krzyżowa

Do sprawdzenia jakości modelu (w naszym przypadku zbudowanego drzewa decyzyjnego), użyjemy walidacji k-krzyżowej, która umożliwi nam sprawdzenie, czy wyniki z odpowiednio dobranego zbioru testującego rzeczywiście odzwierciedlają jakość modelu czy nie.

1.4 Analiza danych

Za pomocą histogramu możemy przeanalizować częstotliwości występowania klas dla różnych atrybutów. Możemy zauważyć, że najczęściej występującą klasą jest unacc. Występuje ona w każdym atrybucie. Kolejną wartością, która występuje dość często jest acc, ale w niektórych przypadkach nie występuje. Najmniej wystąpień mają klasy good oraz vgood.

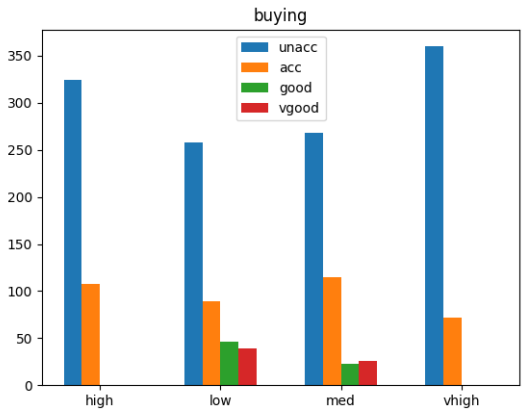

Fig. 1 Częstotliwość klasy buying.

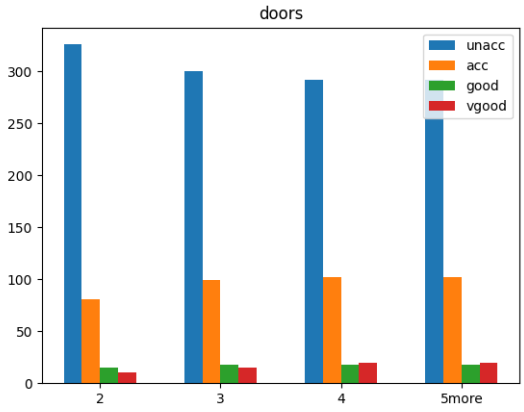

Fig. 2 Częstotliwość klasy doors.

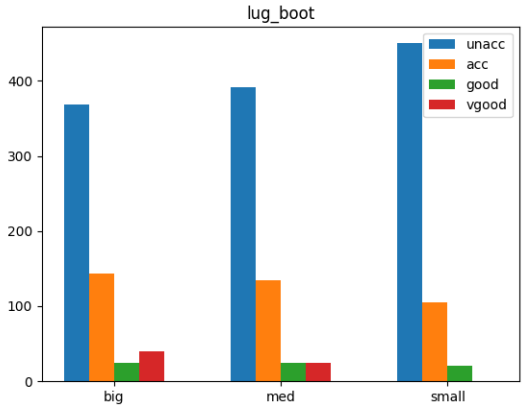

Fig. 3 Częstotliwość klasy lug_boots.

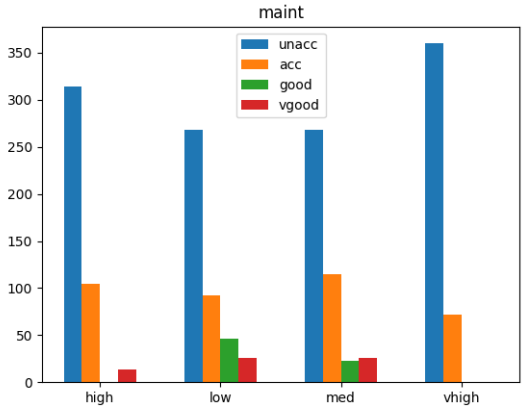

Fig. 4 Częstotliwość klasy maint.

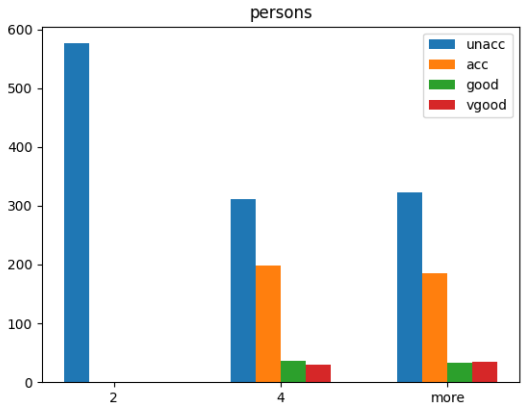

Fig. 5 Częstotliwość klasy persons.

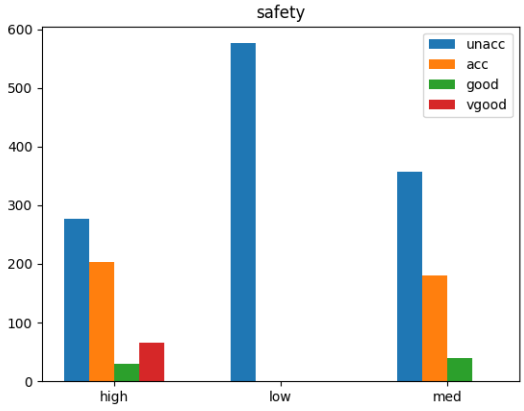

Fig. 6 Częstotliwość klasy safety.

2 Wyniki

2.1 Wyniki w przypadku losowego mieszania danych

W kolejnych podanych tabelach umieszczone zostały wyniki jakości zbudowanego drzewa decyzyjnego. Wyniki predykcji danych testowych są dość wysokie. Widać jednak, że k-krotna walidacja pozwoliła nam pokazać obraz modelu na przestrzeni całego zbioru. Szczególnie widocznie jest to w tabeli nr 2 w momencie porównywania pierwszego rządu z ostatnim. Metryki są zbliżone zbliżone do siebie, ale różnica na poziomie 3% jest wciąż wysoka. Może to wynikać z przetrenowania lub małego zbioru danych.

recall	fallout	precision	accuracy	f1-score
0.9305	0.0231	0.9305	0.9652	0.9305
0.9166	0.0277	0.9166	0.9583	0.9166

Tabela 1 Pomiary wyników zbudowanego drzewa ID3 dla k=2

recall	fallout	precision	accuracy	f1-score
0.9427	0.019	0.9427	0.9713	0.9427
0.927	0.0243	0.927	0.9635	0.927
0.8993	0.0335	0.8993	0.9496	0.8993

Tabela 2 Pomiary wyników zbudowanego drzewa ID3 dla k=3

recall	fallout	precision	accuracy	f1-score
0.9188	0.027	0.9188	0.9594	0.9188
0.9362	0.0212	0.9362	0.9681	0.9362
0.9594	0.0135	0.9594	0.9797	0.9594
0.942	0.0193	0.942	0.971	0.942
0.9275	0.0241	0.9275	0.9637	0.9275

Tabela 3 Pomiary wyników zbudowanego drzewa ID3 dla k=5

recall	fallout	precision	accuracy	f1-score
0.9837	0.0054	0.9837	0.9918	0.9837
0.9512	0.0162	0.9512	0.9756	0.9512
0.939	0.0203	0.939	0.9695	0.939
0.9349	0.0216	0.9349	0.9674	0.9349
0.939	0.0203	0.939	0.9695	0.939
0.9105	0.0298	0.9105	0.9552	0.9105
0.9552	0.0149	0.9552	0.9776	0.9552

Tabela 4 Pomiary wyników zbudowanego drzewa ID3 dla k=7

2.2 Wyniki bez losowego mieszania danych

W przypadku trenowania danych na nielosowych danych z zbioru uczącego, widzimy duży spadek jakości modelu. Wynika to oczywiście z tego, że niektóre wartości atrybutów mogą pojawić się z zbiorze testującym a nie być w zbiorze uczącym.

recall	fallout	precision	accuracy	f1-score
0.6967	0.101	0.6967	0.8483	0.6967
0.6793	0.1068	0.6793	0.8396	0.6793

Tabela 5 Pomiary wyników zbudowanego drzewa ID3 dla k=2

recall	fallout	precision	accuracy	f1-score
0.769	0.0769	0.769	0.8845	0.769
0.7118	0.096	0.7118	0.8559	0.7118
0.6493	0.1168	0.6493	0.8246	0.6493

Tabela 6 Pomiary wyników zbudowanego drzewa ID3 dla k=3

recall	fallout	precision	accuracy	f1-score
0.7739	0.0753	0.7739	0.8869	0.7739
0.8376	0.0541	0.8376	0.9188	0.8376
0.8289	0.057	0.8289	0.9144	0.8289
0.771	0.0763	0.771	0.8855	0.771
0.6927	0.1024	0.6927	0.8463	0.6927

Tabela 7 Pomiary wyników zbudowanego drzewa ID3 dla k=5

recall	fallout	precision	accuracy	f1-score
0.6991	0.1002	0.6991	0.8495	0.6991
0.817	0.0609	0.817	0.9085	0.817
0.9756	0.0081	0.9756	0.9878	0.9756
0.8902	0.0365	0.8902	0.9451	0.8902
0.8577	0.0474	0.8577	0.9288	0.8577
0.7235	0.0921	0.7235	0.8617	0.7235
0.7113	0.0962	0.7113	0.8556	0.7113

Tabela 8 Pomiary wyników zbudowanego drzewa ID3 dla k=7

W przypadku sortowania kolejnych atrybutów, natrafiłem na ciekawy przypadek, w którym bez losowania uzyskujemy dość wysokie wyniki. Wybrany do sortowania atrybut: "doors", dla k=4.

recall	fallout	precision	accuracy	f1-score
0.8819	0.0393	0.8819	0.9409	0.8819
0.956	0.0146	0.956	0.978	0.956
0.9583	0.0138	0.9583	0.9791	0.9583
0.9583	0.0138	0.9583	0.9791	0.9583

Tabela 9 Pomiary wyników zbudowanego drzewa ID3 dla k=4 po posortowaniu atrybutu "doors"

Posortowanie danych po ostatecznej klasie również daje ciekawe wyniki:

recall	fallout	precision	accuracy	f1-score
0.0532	0.3155	0.0532	0.5266	0.0532
0.5393	0.1535	0.5393	0.7696	0.5393
0.8541	0.0486	0.8541	0.927	0.8541
0.7013	0.0995	0.7013	0.8506	0.7013

Tabela 10 Pomiary wyników zbudowanego drzewa ID3 dla k=4 po posortowaniu ostateczną klasą "class"

2.3 Przykład zbudowanego drzewa za pomocą algorytmu ID3

buying	maint	doors	persons	lugboot	safety	class
high	low	4	2	med	high	unacc
med	high	3	4	small	high	acc
med	high	3	2	small	med	unacc
low	vhigh	5more	4	small	med	unacc
med	med	2	2	big	low	unacc
med	med	5more	2	big	med	unacc
high	low	3	2	small	med	unacc
high	vhigh	5more	4	med	high	unacc
vhigh	vhigh	2	2	big	low	unacc
high	med	3	4	big	med	acc
low	high	2	more	big	high	vgood
med	low	5more	4	med	high	vgood
low	low	3	4	big	low	unacc
high	vhigh	4	4	big	low	unacc
med	med	3	4	big	med	acc
vhigh	vhigh	3	more	small	med	unacc

Tabela 11 Przykładowe wylosowane dane do wizualizacji drzewa.

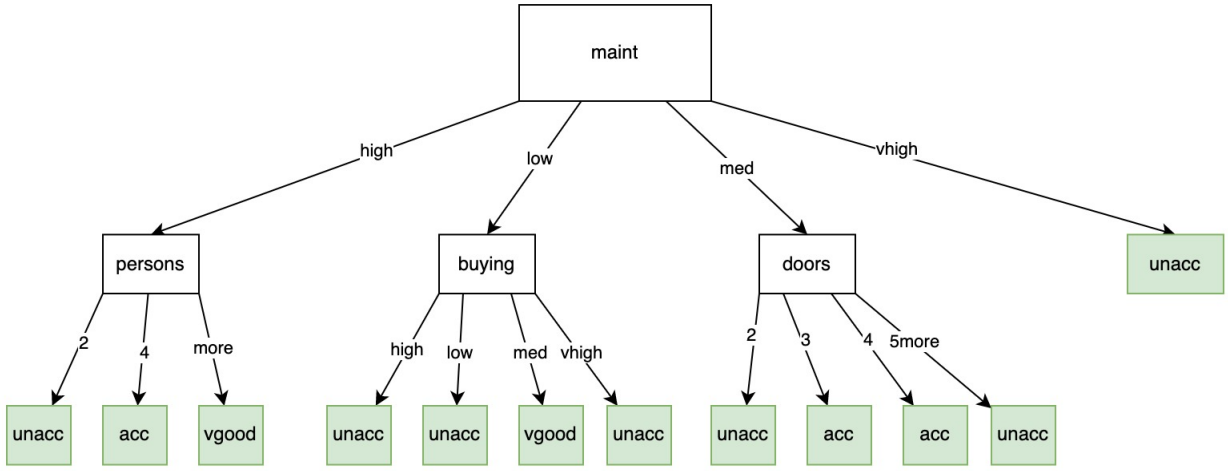

Fig. 7 Wizualizacja drzewa ID3. Kolorem zielonym zostały oznaczone liście.

2.4 Confusion Matrix

W momencie robienia predykcji na danych testowych, będziemy odpowiednio aktualizować macierz pomyłek. Przykład wyglądu macierzy została wizualizowana na rysunku (Fig. 8). Można zauważyć, że najwięcej wartości znajduję się w polach znajdujących się na diagonalnej macierzy. Oznacza to, że model w dużej ilości poprawnie sklasyfikował ostateczną klasę.

	unacc	acc	good	vgood
unacc	286	1	0	0
acc	18	96	3	1
good	1	1	10	1
vgood	0	2	1	11

Fig. 8 Przykładowa macierz pomyłek.

2.5 Pomiary dla każdej klasy

Tabela widoczna na rysunku (Fig. 9) została umieszczona w celu pokazania, że do każdej klasy jesteśmy w stanie zmierzyć odpowiednie metryki. Jednak lepiej jest te metryki wyznaczyć dla wszystkich klas łącznie.

	recall	fallout	precision	accuracy	f1-score
unacc	0.99	0.05	0.98	0.97	0.98
acc	0.90	0.02	0.93	0.96	0.91
good	0.46	0.02	0.61	0.95	0.52
vgood	0.75	0.02	0.55	0.97	0.63

Fig. 9 Tabela metryk dla każdej klasy.

3 Wnioski

Algorytm ID3 dla przykładowych danych okazał się dość dobrym rozwiązaniem. Jednak jest on dość podatny na przetrenowanie (overfitting). Szczególnie widoczne jest to w momencie sortowania po atrybucie "doors". Widać również niedotrenowanie w przypadku trenowania na posortowanych danych po wynikowej klasie. W celu poprawienia jakości modelu i zwiększenie odporności na przetrenowanie, można by było użyć lasu, czyli stworzenie dużo małych drzew (z losowymi usuniętymi atrybutami) i odpowiednie wybranie ostatecznych klas.