Data: 09.06.2020

Dokumentacja

Opis zadania:
Przewidywanie czy grzyb jest jadalny przy użyciu zmodyfikowanej implementacji algorytmu ID3. Do wyboru testu w drzewie stosowany jest turniej. Do turnieju wybierane są dwa testy losowo. Należy porównać działanie ID3 z klasycznym testem.

Autorzy:
Artur Wyrozębski – implementacja algorytmu ID3 w języku Python oraz k-krotnej walidacji krzyżowej
Filip Gralak – sporządzenie dokumentacji projektu

Decyzje projektowe:
1. Dane mogą być tylko wczytywane z pliku
2. Dane treningowe muszą zawierać w pierwszej kolumnie pliku klasę obiektu, natomiast w kolejnych atrybuty
3. Wartość klasy oraz wartości atrybutów obiektu w linii mogą zawierać litery, cyfry oraz znak zapytania. Każdy inny znak traktowany jest jako separator.
4. Graf utworzonego drzewa jest reprezentowany w postaci tekstowej w taki sposób, że linijka określa dany węzeł drzewa i opis węzła jest postaci:
   „nr_kolumny_atrybutu -> wartość_atrybutu: klasa_lub_nr_kolumny_atrybutu, …”
   Linijki zawierające numery kolumny atrybutu mogą się powtarzać, ale należy zwrócić uwagę na tę, która jest najbliższa od końca tekstu wyjścia. Informacje o węzłach są wypisywane rekurencyjnie.
5. W k-krotnej walidacji krzyżowej liczba sprawdzeń algorytmu ID3 z turniejowym wyborem testu wynosi 100, natomiast z klasycznym wynosi 1, ponieważ klasyczny jest deterministyczny. Współczynnik błędu dla turniejowego algorytmu ID3 jest uśredniany dla 100 sprawdzeń.

Wykorzystane narzędzia:
python 3.7.3

Instrukcja uruchomienia:
python id3.py c *ścieżka_do_danych_treningowych* – utworzenie drzewa klasycznym wyborem testu
python id3.py t *ścieżka_do_danych_treningowych* – utworzenie drzewa turniejowym wyborem testu
python id3.py cmp [*wartość_k*] *ścieżka_do_danych* – dokonanie walidacji krzyżowej na obu poprzednich podejściach

Cel badań:
Celem badań jest porównanie podejścia turniejowego wyboru testu algorytmu ID3 z klasycznym wyborem testu poprzez porównanie jakości utworzonych drzew.

Teza:
Drzewa utworzone podejściem klasycznym mają mniejszy współczynnik błędu i mają mniej wierzchołków.

Sposób przeprowadzania badań:

Do wyznaczenia współczynnika błędu stosowana jest k-krotna walidacja krzyżowa. Najpierw dane otrzymane na wejściu są dzielone na dane treningowe i dane walidacyjne, gdzie do danych walidacyjnych trafia *całkowita_ilość_danych/k* obiektów, a reszta trafia do danych treningowych. Potem tworzone jest drzewo na podstawie danych treningowych i wyznaczany jest współczynnik

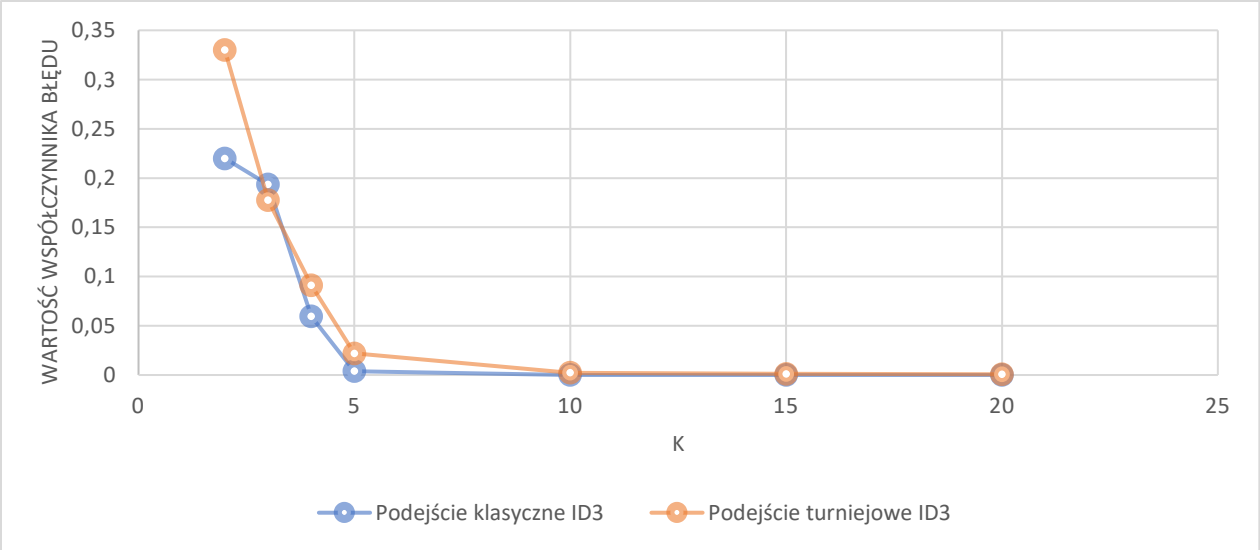
błędu poprzez sprawdzenie ile razy zostanie popełniony błąd klasyfikacji po wykonaniu walidacji danymi walidacyjnymi. Na koniec współczynnik błędu jest dzielony przez ilość danych walidacyjnych. Cały ten algorytm jest dokonywany k razy przy czym do danych walidacyjnych trafia zawsze inna część danych całkowitych.

Opis danych wykorzystanych w eksperymencie:
W pliku *agaricus-lepiota.data* znajduje się 8124 obiektów grzybów. Pierwsza kolumna określa klasę grzyba – czy jest jadalny czy trujący. Każdy grzyb może mieć 22 atrybutów.

Wyniki eksperymentów:

k	Podejście klasyczne ID3	Podejście turniejowe ID3
2	0,22	0,33
3	0,1935	0,1775
4	0,0595	0,0912
5	0,0039	0,022
10	0	0,00231
15	0	0,0011
20	0	0,00073

Tabela wartości współczynnika błędu dla danego k i podejścia.
Dla podejścia turniejowego współczynnik błędu jest średnią ze 100 sprawdzeń.



Drzewo utworzone podejściem klasycznym:
5 -> p: p, a: e, l: e, n: 20, f: p, c: p, y: p, s: p, m: p
20 -> n: e, k: e, w: 18, h: e, r: p, o: e, y: e, b: e
18 -> t: e, o: 13
13 -> f: e, y: p, s: 21
21 -> c: p, v: e

Typowe drzewo utworzone podejściem turniejowym:
5 -> p: p, a: e, l: e, n: 1, f: p, c: p, y: p, s: p, m: p
1 -> x: 15, s: e, f: 18, b: 15, k: 7, c: p
15 -> w: e, p: e, g: e, e: e, n: 12, y: p, o: e

12 -> s: e, k: p, f: e, y: e
18 -> o: 21, t: 11
21 -> a: e, y: e, v: 10, s: e, c: 19
10 -> e: 15, t: e
15 -> w: e, n: 20, y: p, o: e
20 -> w: 11
11 -> b: e, ?: p
19 -> p: 7, e: p
7 -> w: p, c: e
11 -> ?: e, b: 17
17 -> w: 4
4 -> t: 12, f: e
12 -> s: 19
19 -> p: 2
2 -> s: 21, y: 13
21 -> v: 15, y: e
15 -> w: 20
20 -> r: p, w: e
13 -> s: 20
20 -> r: p, w: e
15 -> w: 4, n: 18, y: p, o: e
4 -> t: p, f: e
18 -> o: p, t: e
7 -> w: 21, c: 14
21 -> v: e, c: p, s: e, n: e
14 -> w: 15, e: e, o: e, n: e
15 -> e: e, n: p, w: e, y: p

Omówienie wyników i wnioski:
Jak można zauważyć współczynnik błędu jest zwykle niższy dla podejścia klasycznego co dowodzi, że podejście klasyczne wyboru testu jest lepsze niż podejście turniejowe. Różnica współczynników błędu dla każdego k jest jednak nieznaczna, gdzie raz nawet współczynnik błędu był niższy u podejścia turniejowego dla k=3.

Drzewo dla klasycznego testu ma mniej węzłów, niż drzewo dla podejścia turniejowego.
Można zauważyć w drzewie turniejowym przypadki, gdzie dany węzeł ma tylko jedną krawędź. Zysk informacyjny atrybutu, który został wybrany, wynosi wtedy 0 i atrybut z tym zyskiem informacyjnym jest wybierany wtedy, gdy w turnieju biorą udział atrybuty o tym samym zysku informacyjnym wynoszącym 0. Taki zysk wynika z tego, że do danego węzła trafia zbiór danych, gdzie w tych atrybutach są takie same wartości. Można zlikwidować takie przypadki tworząc ponownie turnieje, aż zysk informacyjny ostatecznego atrybutu będzie różny od 0.

Turniejowe podejście wyboru testu z dwoma atrybutami w algorytmie ID3 można wykorzystać dla przypadku, gdzie dane zawierają mnóstwo atrybutów np. 100. W podejściu tym zawsze wybierane są dwa atrybuty do obliczenia zysku informacyjnego i wykorzystania tego z wyższą wartością. Powoduje to, że tworzenie drzewa będzie zachodziło szybciej w tej sytuacji, niż dla klasycznego podejścia wyboru testu w ID3, które musi obliczać zysk informacyjny w każdym węźle dla statystycznie większości atrybutów. Obliczanie zysku informacyjnego jest kosztowne ze względu na to, że algorytm obliczający musi przejść przez dane treningowe, które otrzymał w danym węźle.