SPRAWOZDANIE

Zajęcia: Analiza Procesów Uczenia

Prowadzący: prof. dr hab. Vasyl Martsenuk

Laboratorium 4

Data: 06.03.2020r

Temat: Uczenie maszynowe z użyciem drzew decyzyjnych

Wariant 1

Aleksander Słodczyk

Informatyka II stopień

Stacjonarne

1 semestr

Grupa 2

1. **Polecenie**:

Zadanie dotyczy prognozowania oceny klientów (w skali 5-punktowej, Error < 5%) urządzeń RTV AGD. Używając metody indukcji drzewa decyzji C5.0 opracować plik w języku R z wykorzystaniem paczki C50.

Mój wariant zawiera Smartfony Samsung, w których uwzględniam: wyświetlacz, pamięć RAM, pamięć wbudowaną oraz aparat foto. Dane pobrałem ze strony http://www.euro.com.pl

2. Wprowadzane dane

Na poniższym obrazku są przedstawione dane 11 smartfonów. Opinia klientów została ściągnięta ze strony ceneo.pl i składa się z czterech klas: 4.6, 4.7, 4.8, 4.9, które odpowiadają skali 5-punktowej

	wyswietlacz	pamiec_RAM	pamiec_wbudowana	aparat_foto	opinia_klientow
1	6.7	8	128	64	4.8
2	6.7	6	128	64	4.6
3	5.9	4	64	16	4.7
4	6.4	4	64	25	4.7
5	6.7	6	128	32	4.7
6	6.5	4	128	48	4.8
7	6.1	8	128	16	4.8
8	6.2	2	32	13	4.7
9	6.7	8	128	48	4.9
10	6.7	6	128	12	4.8
11	5.8	3	32	13	4.7

3. Wykorzystane komendy:

- Komenda **C5.0** dopasowuje klasyfikację modelu drzewa używając algorytm Quinlan C5.0. Na wyjściu jest model C5.0

```
smartphones_tree <- C5.0(smartphones[,-5], as.factor(smartphones[,5]))
```

- Komenda **summary** wypisuje szegółowe podsumowanie modelu C5.0

summary(smartphones_tree)

- Komenda **plot** rysuje sklasyfikowane drzewo wyboru

plot(smartphones_tree, main = 'Drzewo wyboru smartfonów')

- Parametr "**rules**" w komendzie **C5.0** dekomponuje drzewo do modelu opartego o role

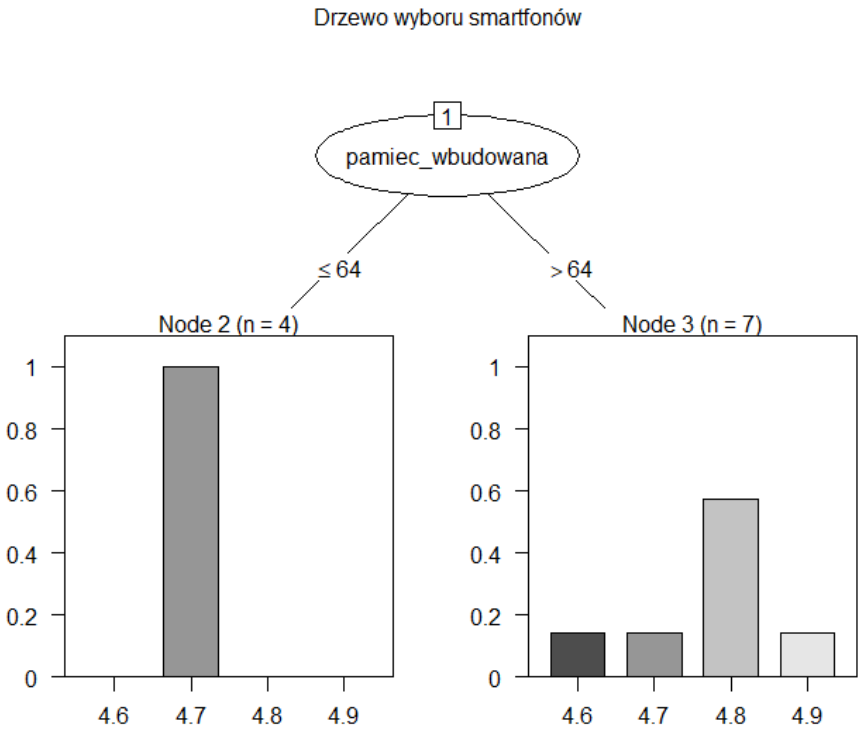
smartphones_tree_rules <- C5.0(smartphones[,-5], as.factor(smartphones[,5]), **rules** = TRUE)

4. Wynik działania

Po wykonaniu metody C5.0 oraz summary dostałem następujące wyniki:

```
Call:
C5.0.default(x = smartphones[, -5], y = as.factor(smartphones[, 5]))


C5.0 [Release 2.07 GPL Edition]         Tue Apr 21 12:44:20 2020
-------------------------------

Class specified by attribute `outcome'

Read 11 cases (5 attributes) from undefined.data

Decision tree:

pamiec_wbudowana <= 64: 4.7 (4)
pamiec_wbudowana > 64: 4.8 (7/3)


Evaluation on training data (11 cases):

           Decision Tree
         ----------------
         Size      Errors

            2    3(27.3%)   <<


          (a)   (b)   (c)   (d)    <-classified as
         ----  ----  ----  ----
                           1          (a): class 4.6
                     4     1          (b): class 4.7
                           4          (c): class 4.8
                           1          (d): class 4.9


        Attribute usage:

        100.00% pamiec_wbudowana


Time: 0.0 secs
```

Po wykonaniu komendy plot otrzymałem następujący obraz drzewa decyzyjnego:



Na podstawie dostarczonych danych okazuje się, że największy wpływ na pozytywną ocenę klienta ma pamięć wbudowana smartfona. Dla większej ilości pamięci wbudowanej, oceny klientów są dużo lepsze, średnio powyżej 64 gb jest to ocena 4.8, a wartości równe lub mniejsze otrzymują 4.7.

Błąd wyniósł 27.3%

Po wykonaniu metody C5.0 z parametrem rules = TRUE otrzymałem następujące wyniki:

```
Call:
C5.0.default(x = smartphones[, -5], y = as.factor(smartphones[, 5]), rules = TRUE)


C5.0 [Release 2.07 GPL Edition]         Tue Apr 21 12:44:20 2020
-------------------------------

Class specified by attribute `outcome'

Read 11 cases (5 attributes) from undefined.data

Rules:

Rule 1: (4, lift 1.8)
        pamiec_wbudowana <= 64
        ->  class 4.7  [0.833]

Rule 2: (7/3, lift 1.5)
        pamiec_wbudowana > 64
        ->  class 4.8  [0.556]

Default class: 4.7


Evaluation on training data (11 cases):

             Rules
       ----------------
         No      Errors

          2    3(27.3%)   <<


          (a)   (b)   (c)   (d)    <-classified as
          ----  ----  ----  ----
                       1          (a): class 4.6
                 4     1          (b): class 4.7
                       4          (c): class 4.8
                       1          (d): class 4.9


        Attribute usage:

        100.00% pamiec_wbudowana


Time: 0.0 secs
```

Są tu przedstawione dwie decyzje (> 64 lub <= 64) oraz najbardziej znaczące klasy oceny klienta dla każdej z nich. Odpowiednio (4.8 lub 4.7). Obok klas, w nawiasach kwadratowych znajduje się oczekiwana strata dla każdej decyzji. Straty wynoszą odpowiednio:

$$SPU(d_i) = \begin{cases} 0.833, & pamięć <= 64\ gb \\ 0.556, & pamięć > 64\ gb \end{cases}$$

5. **Wnioski:**

- Na podstawie straty można ocenić, że ocena klientów będzie wyższa dla pamięci większej niż 64 gb.
- Już po samych obserwacjach danych można stwierdzić, że pozytywna ocena klientów jest najbardziej skorelowana z większą ilością pamięci wewnętrznej smartfonu.
- Błąd modelu jest duży z powodu braku większej ilości danych do przetrenowania.

6. **Link do githuba:**

https://github.com/allo97/Analiza-procesow-uczenia-Programming-in-R/tree/master/lab4

Skrypt znajduje się w pliku DrzewoDecyzji/**decyzja_smartfonow.R**
Obraz sieci znajduje się w pliku DrzewoDecyzji/**drzewo_decyzji_smarfonow.png**
Wyniki z consoli znajdują się w pliku DrzewoDecyzji/**console_log**