Mateusz Guściora, 228884

Uwagi\* Nie udało mi się zmienić nazwy relacji i plik, który nazywa 228884\_bank później w wynikacj jak i reletions w programie Weka nazywa się xxxxxxx\_bank-weka (tak go nazwałem roboczy gdy pracowałem nad źródłem danych bank additional-full).

1a)

Ranker dla klienci\_2

Search Method: Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 8 przedzial): Information Gain Ranking Filter

Ranked attributes:

0.04694 1 numer klienta

0.0368 7 DzienTygodnia

0.01754 2 przedstawiciel

0.01615 3 Oddzia³

0.01304 4 Region

0.01086 6 przedzial\_czas rozmowy

0.00569 5 miesi¹c

Ranker dla klienci6

Search Method: Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 9 przedzialkwotowy): Information Gain Ranking Filter

Ranked attributes:

0.02259 1 numerklienta

0.01084 4 przedzialczasrozmowy

0.01016 3 dzientygodnia

0.00841 5 przedstawiciel

0.00816 7 oddzial

0.00632 6 region

0.00285 2 miesiac

0.00162 8 plec

Ranker dla klienci6test

Search Method: Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 9 przedzialkwotowy): Information Gain Ranking Filter

Ranked attributes:

0.12116 1 numerklienta

0.04009 4 przedzialczasrozmowy

0.03135 3 dzientygodnia

0.0201 5 przedstawiciel

0.00827 8 plec

0.00546 7 oddzial

0.00432 2 miesiac

0.00359 6 region

1b) zapisane poprzez „Save result buffer”

1d) zapisane poprzez „Save result buffer”, testowany na zbiorze klienci6test

1e) zostały utworzone pliki csv :

Klienci6newwynik z algorytmem JRip

Klienci6newwynik2 z algorytmem J48.

Po ich edycji w notepad możemy zobaczyć jak zostały zaklasyfikowane nowe przypadki (przypadki z pliku klienci6new) w zmiennej ‘predicted values’.

W pliku są dwie grupy wartości ‘predicted values’ „150-250” i „50-150”

1f)

Walidacja krzyżowa klienci6

Przewidywana przedzialkwotowy: 50-150

|  |  |  |
| --- | --- | --- |
|  | Algorytm JRip | Algorytm J48 |
| Correctly Classified Instances | 45.9 % | 45.8 % |
| Kappa statistic | -0.0068 | 0.0231 |
| Mean absolute error | 0.4059 | 0.3942 |

Walidacja krzyżowa klienci6 ze zbiorem testowym

Przewidywana przedzialkwotowy: 50-150

|  |  |  |
| --- | --- | --- |
|  | Algorytm JRip | Algorytm J48 |
| Correctly Classified Instances | 45.749 % | 51.8219 % |
| Kappa statistic | -0.0186 | 0.1206 |
| Mean absolute error | 0.4043 | 0.3684 |

Walidacja ze zbiorem nowych transakcji klienci6new

Zaklasyfikowalem nowe przypadki przewidywanych przedziałówkwotowych🡪 punkt 1e)

a następnie przeprowadzadziłem predykcje przedziału kwotowego używając algorytmów JRip i J48 dla zbioru klienci6newwynik\_probny2.arff na zbiorze testowym (‘supplied test’ klienci6test).

Przewidywana przedzialkwotowy: 50-150

|  |  |  |
| --- | --- | --- |
|  | Algorytm JRip | Algorytm J48 |
| Correctly Classified Instances | 54.4186 % | 54.8837 % |
| Kappa statistic | -0.0278 | -0.0014 |
| Mean absolute error | 0.4558 | 0.467 |

2g

**Wynik dla Percent\_correct**

Dataset (1) rules.JR | (2) trees (3) trees (4) funct (5) funct (6) meta.

------------------------------------------------------------------------------------------

klienci6 (100) **47.11 | 45.40 43.08 \* 47.05 45.47 47.44**

------------------------------------------------------------------------------------------

(v/ /\*) | (0/1/0) (0/0/1) (0/1/0) (0/1/0) (0/1/0)

Key:

(1) rules.JRip '-F 3 -N 2.0 -O 2 -S 1' -6589312996832147500

(2) trees.J48 '-C 0.25 -M 2' -217733168393644448

(3) trees.RandomForest '-P 100 -I 20 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428740

(4) functions.SimpleLogistic '-I 0 -M 50 -H 50 -W 0.0' 7397710626304705500

(5) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 50 -V 10 -S 0 -E 20 -H a -R' -5990607817048210400

(6) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.DecisionStump' -1178107808933117950

Dokładnośc dla algorytmów JRip, J48, RandomForest, SimpleLogistic, MultilayerPerceptron, AdaBoost wynosi kolejno 47.11, 45.40, 43.08, 47.05, 45.47, 47.44. Najbardziej dokładny okazała się funkcja MultilayerPerceptron.

**Wynik dla Kappa\_statistic**

Dataset (1) rules.J | (2) tree (3) tree (4) funct (5) func (6) meta.

--------------------------------------------------------------------------------------

**klienci6 (100) 0.01 | 0.01 0.02 -0.00 0.00 -0.00**

--------------------------------------------------------------------------------------

(v/ /\*) | (0/1/0) (0/1/0) (0/1/0) (0/1/0) (0/1/0)

Key:

(1) rules.JRip '-F 3 -N 2.0 -O 2 -S 1' -6589312996832147500

(2) trees.J48 '-C 0.25 -M 2' -217733168393644448

(3) trees.RandomForest '-P 100 -I 20 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' 1116839470751428740

(4) functions.SimpleLogistic '-I 0 -M 50 -H 50 -W 0.0' 7397710626304705500

(5) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 50 -V 10 -S 0 -E 20 -H a -R' -5990607817048210400

(6) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.DecisionStump' -1178107808933117950

Wskaźnik Kappa\_statistic dla algorytmów JRip, J48, RandomForest, SimpleLogistic, MultilayerPerceptron, AdaBoost to kolejno: 0.01 , 0.01, 0.02, -0.00, 0.00, -0.00 . Najlepszy wskaźnik jest przy algorytmie drzewa RandomForrest(choć wciąż niski i niezadowalający).

3\*

3ab) odcinamy 80% z pierwotnego pliku ale najpierw stosując filter randomize. Pozostało 8638 wyników.

3d)

**Wynik dla Percent\_correct**

Wyniki zapisano do plików : wyniki\_bank.csv oraz wyniki\_bank\_2repetitions.csv.

Dla algorytmów kolejno JRip,J48, RandomForest AdaBoost.

Dataset (1) rules.JR | (2) trees (3) trees (4) meta.

----------------------------------------------------------------------

'xxxxxxx\_bank-weka.filter**(100) 88.93 | 89.26 87.71 \* 88.66**

----------------------------------------------------------------------

(v/ /\*) | (0/1/0) (0/0/1) (0/1/0)

Dla algorytmów kolejno SimpleLogistic i MultilayerPerceptron z 2 powtórzeniami(repetitions).

Dataset (1) function | (2) funct

--------------------------------------------------

'xxxxxxx\_bank-weka.filter (20) 89.33 | 88.85 \*

--------------------------------------------------

(v/ /\*) | (0/0/1)

Dokładności algorytmów kształtują się na podobnym poziomie. Najlepszą dokładność ma algorytm J48.

**Wynik dla Kappa\_statistic**

Dla Algorytmów kolejno JRip,J48, RandomForest, AdaBoost

Dataset (1) rules.J | (2) tree (3) tree (4) meta

------------------------------------------------------------------

'xxxxxxx\_bank-weka.filter**(100) 0.37 | 0.40 0.36 0.27 \***

------------------------------------------------------------------

(v/ /\*) | (0/1/0) (0/1/0) (0/0/1)

Dla algorytmów kolejno SimpleLogistic i MultilayerPerceptron z 2 powtórzeniami(repetitions).

Dataset (1) functio | (2) func

------------------------------------------------

'xxxxxxx\_bank-weka.filter (20) 0.38 | 0.37

------------------------------------------------

(v/ /\*) | (0/1/0)

Najlepszy wynik Kappa\_statistic ma algorytm drzewa J48 wynosi on 0,4. Jest to wynik zdecydowanie wyższy niż w poprzednim zadaniu(klienci). Co świadczy o poprawie przydatności danych do predykcji zmiennej.

3e)

Została stworzona nowy zestaw danych ze zredukowaną liczbą zmiennych (3e\_zredukawana\_228884\_bank6a.arff). Następnie na tym zestawie oraz na zesstawie danych (228884\_bank6) ze większą ilość zmiennych została uruchomione algorytmy JRip, J48(ze zmienionymi parametrami), JRip(ze zmienionymi parametrami), J48.

Percent correct

Dataset (1) rules.JR | (2) trees (3) rules (4) trees

----------------------------------------------------------------------

'xxxxxxx\_bank-weka.filter(100) 88.89 | 88.97 88.68 88.95

'xxxxxxx\_bank-weka.filter(100) 88.93 | 89.22 88.35 \* 89.26

----------------------------------------------------------------------

(v/ /\*) | (0/2/0) (0/1/1) (0/2/0)

Zredukowanie zmiennych zmniejszyło dokładność predykcji oprócz w algorytmie JRip (ze zmienionymi parametrami-usePruning-False)

Kappa\_statistic

Dataset (1) rules.J | (2) tree (3) rule (4) tree

------------------------------------------------------------------

'xxxxxxx\_bank-weka.filter(100) 0.35 | 0.30 \* 0.26 \* 0.30 \*

'xxxxxxx\_bank-weka.filter(100) 0.37 | 0.39 0.19 \* 0.40

------------------------------------------------------------------

(v/ /\*) | (0/1/1) (0/0/2) (0/1/1)

Redukcja obniżyła wartość Kappa\_statisctic negatywnie wpływając na predykcje znów oprócz JRip (ze zmienionymi parametrami-usePruning-False MinNu-4)

\*uwaga Zbiór pierwszy jest zbiorem ze zredukowaną liczbą zmiennych(3e\_zredukawana\_228884\_bank6a) a ten zbiór niżej(xxxxx\_bank) jest na zbiorze z większą ilością zmiennych.