Monika Szymańska 238359

Zadanie 2.5

Zadanie 25

Przykład wprowadzający

Aplikacja WEKA KnowledgeFlow pozwala nam w graficzny sposób budować modele analityczne. Wybierając w Load a template layout opcję CrossValidation możemy zapoznać się z zastosowaniem walidacji krzyżowej na zbiorze danych o irysach. Po jego uruchomieniu widzimy, że skuteczność użytego algorytmu J48 wynosi 96%, natomiast wartość Kappa to 0,94 zatem możemy uznać klasyfikację za wysoce skuteczną.

Takie same analizy zostały przeprowadzone dla zbioru danych Fishers Iris Dataset. Wynik przeprowadzonej klasyfikacji był nieco gorszy ponieważ źle zaklasyfikowany został dodatkowo jeden argument co sprawiło, że procent poprawności wyniósł 95,3%, a wartość Kappa 0,93. W dalszym ciągu pozwala nam to uznać, klasyfikację za doskonałą.

Dodatkowo zostały dodane bloki ImageSave oraz TextSave, które mają za zadanie zapisać wyniki klasyfikacji do odpowiednich plików (Fishers Iris Dataset_1 oraz Fishers Iris Dataset_2).

Ocena klasyfikatorów dla danych z zadania 1.1. i 2.4.

Dla danych z zadania 1.1 oraz 2.4 chcemy ustalić, który z algorytmów klasyfikacyjnych jest najbardziej skuteczny. W tym celu zostały przeprowadzone analizy a jej wyniki zostały zaprezentowane w Tabeli 1 i Tabeli 2.

Algorytm \ Plik	238359_klienci6	238359_bank6	GlassData	AutoMPG
J48	45,8%	69,38%	97,66%	43,00%
NaiveBayes	45,2%	66,98%	71,03%	42,24%
JRip	45,9%	70,17%	94,86%	44,02%
AdaBoostM1	46,6%	69,42%	54,21%	46,06%
RandomForest	43,7%	66,02%	99,07%	30,28%
Logistic	43,2%	70,43%	80,37 %	45,55%

Tabela 1 - Porównanie wyników klasyfikacji dla danych z zadania 1.1 i 2.4 - % poprawność klasyfikacji

Jak widać powyżej najwyższa procentowa poprawność klasyfikacji jest różna dla różnych algorytmów i różnych zbiorów danych zatem nie można uznać, że któryś z zastosowanych algorytmów jest najlepszy. Dla zbioru danych o klientach najlepszy okazał się AdaBoostM1, jednak różnica między pozostałymi klasyfikatorami była bardzo niewielka. Dane bankowe zostały najlepiej sklasyfikowane algorytmem Logistic -70,43%, natomiast GlassData uzyskał niemal doskonałą klasyfikację dzięki zastosowaniu RandomForest. AdaBoostM1 okazał się również najbardziej skuteczny dla danych AutoMPG, jednak procent poprawności wyniósł zaledwie 46,06%, zatem nie możemy tutaj mówić o dobrym wyniku klasyfikacji.

Dla tych samych zbiorów danych w następujący sposób przedstawiają się wartości wskaźnika Kappa:

Algorytm \ Plik	238359_klienci6	238359_bank6	GlassData	AutoMPG
J48	0,0231	0,3151	0,9659	0,3359
NaiveBayes	0,0402	0,2989	0,5661	0,3346
JRip	-0,0068	0,339	0,9243	0,3456
AdaBoostM1	-0,0131	0,2944	0,3027	0,1694
RandomForest	0,0241	0,264	0,9346	0,373
Logistic	-0,0174	0,3344	0,7151	0,3738

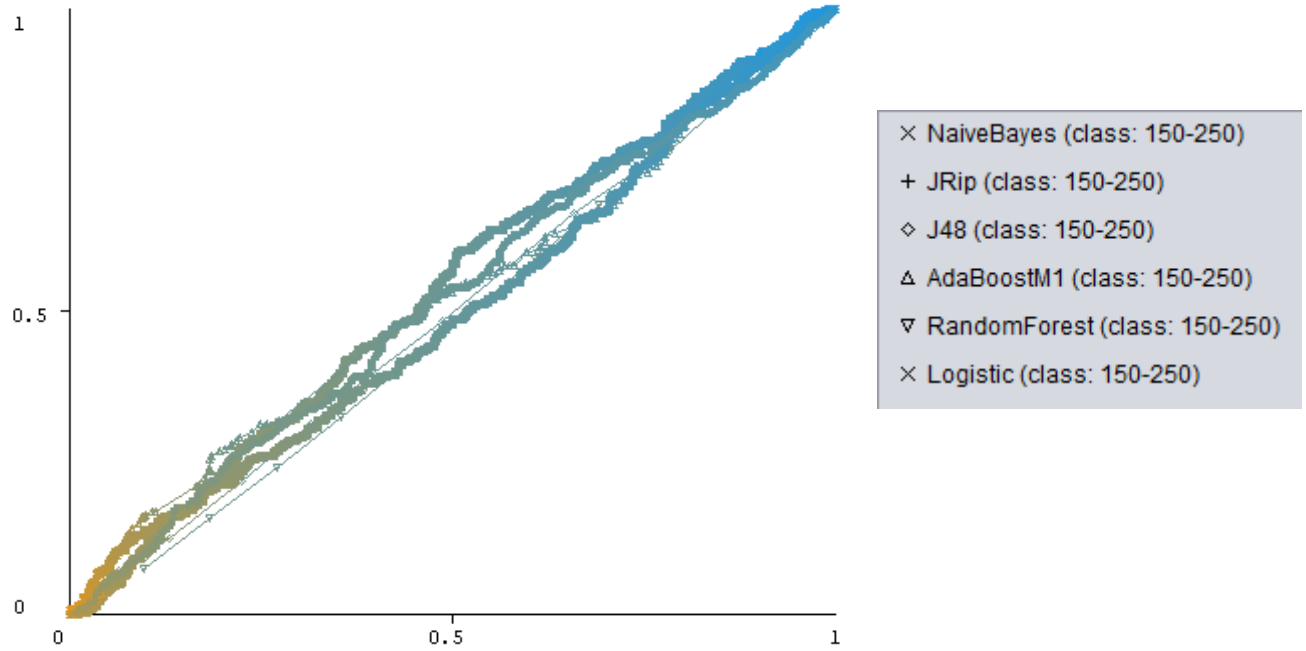
Tabela 2 - Porównanie wyników klasyfikacji dla danych z zadania 1.1 i 2.4 – wartość Kappa

Dla zbioru danych o klientach na podstawie wskaźnika Kappa nie możemy stwierdzić poprawności żadnego z zastosowanych algorytmów. Taka sama sytuacja występuje dla zbiorów danych bankowych, a także AutoMPG. Jedynie dane GlassData posiadają wartości wskaźnika Kappa dla algorytmów J48, JRip oraz RandomForest pozwalające uznać klasyfikację za doskonałą oraz średnią dla algorytmów NavieBayes i Logistic. Jedynie AdaBoostM1 posiada wartość pozwalającą określić klasyfikację jako niedostateczną.

Ocena klasyfikatorów z wykorzystaniem krzywych ROC

Model powstały w poprzednim etapie został rozszerzony o wizualizację krzywej ROC, a także moduł pozwalający określić klasę pozytywną zmiennej wyjściowej.

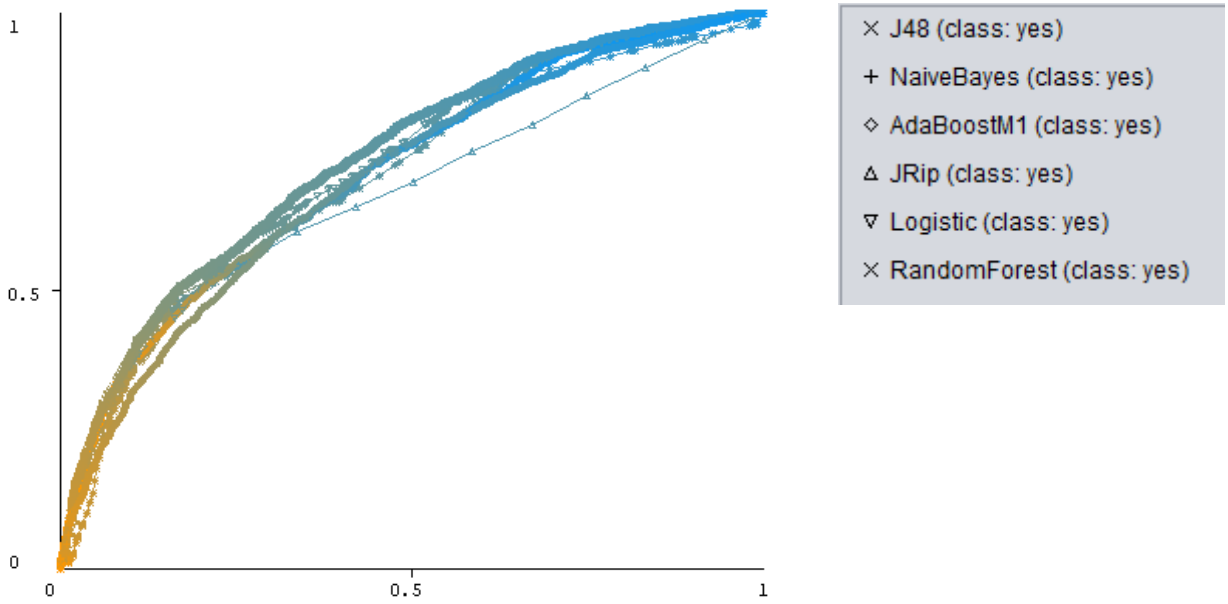
Krzywa ROC dla danych o klientach podając jako klasę pozytywną najwyższy przedział wartości zakupów czyli 150-250 została zaprezentowana na Rysunku 1.



Rysunek 1 - Krzywa ROC dla danych o klientach (klasa pozytywna - przedział kwoty zakupu 150-250)

Z powyższego rysunku ciężko jest zadecydować, który z użytych klasyfikatorów można uznać za najlepszy. Wszystkie AUC oscylują w okolicach 0,5 zatem możemy uznać, że mamy do czynienia z klasyfikatorami losowymi.

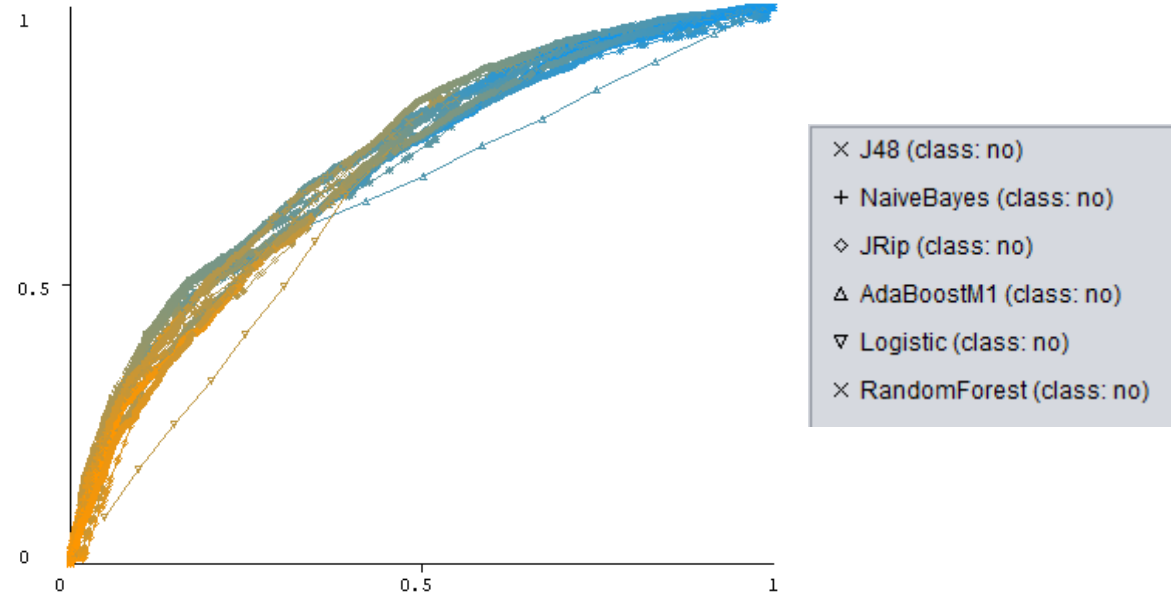
Dla danych bankowych również podjęto się zastosowania klasyfikatorów z wykorzystaniem krzywej ROC. W tym celu jako klasę pozytywną wybrano kolejno „yes” oraz „no” zmiennej wyjściowej „y”. Krzywa ROC dla „yes” została zaprezentowana na Rysunku 2.



Rysunek 2 - Krzywa ROC dla danych bankowych (klasa pozytywna – yes dla zmiennej y)

Jak widać na Rysunku 2 użyte przez nas algorytmy mają podobne wyniki AUC (wizualnie wyjątek stanowi RandomForest z powodu użycia mniejszej ilości drzew) oscylujące w okolicach 0,7. Taki wynik sprawia, że pole jest bardziej wypukłe, jednak w dalszym ciągu daleko mu do idealnego poziomu.

Dla wartości „no” zmiennej „y” wykres ROC został przedstawiony na Rysunku 3.



Rysunek 3 - Krzywa ROC dla danych bankowych (klasa pozytywna – no dla zmiennej y)

Dla wartości „no” zmiennej „y” również możemy zaobserwować wyniki oscylujące w okolocach 0,7, zatem skuteczność wykrywaniaobiektów dla klasy pozytywnej „yes” oraz „no” jest taka sama. W zwiazku z tym możemy stwierdzić, że dokładność rozpoznania klientów, którzy założą lokatę jest taka sama jak dla klientów, którzy jej nie założą.

Zapoznaj się z innymi przykładami funkcjonalności Knowledge Flow

Dla zboru danych bankowych zostało zastosowanie przetwarzanie strumieniowe z wykorzystaniem klasyfikatora IBk. Wyniki analizy zostały zapisane do pliku 238359_bank_e3.txt. Jeżeli chodzi o procentową poprawność klasyfikacji to kształtuje się ona na poziomie 63,47% co nie jest najlepszym wynikiem. Potwierdza to również wartość wskaźnika Kappa wynosząca zaledwie 0,2126.

Program Weka Knowledge Flow pozwala nam na zastosowanie parametryzacji procesu. W związku z tym zastosowano ją dla danych bankowych oraz danych o klientach wybierając wcześniej już użyte algorytmy klasyfikacji (J48, JRip, NaiveBayes, RandomForest, AdaBoostM1 oraz Logistic). Wyniki zostały zapisane w pliku 238359_zadanie_e4.txt. Takie rozwiązanie pozwala nam zobaczyć jakie reguły zostały zastosowane w danym algorytmie podczas klasyfikacji czyli jaka wartość danej zmiennej pozwala nam określić do jakiej grupy zostanie zaklasyfikowany dany obiekt.