P4GL
Zadanie 1 - Laboratorium

Joanna Klinkiewicz
Weronika Walczak
Szymon Malański

Informatyka 2. stopień

Niniejszy dokument przedstawia 3 algorytmy klasyfikacji danych:
1. k-NN
2. klasyfikator bayesowski
4. SVM

Wykorzystany zbiór danych dotyczy nowotworu piersi. Zbiór składający się z 569 elementów zawiera 32 różne atrybuty ilościowe, z których tylko jeden nie przynosi żadnych wartości poznawczych (id). Natomiast klasyfikatorem - atrybutem decyzyjnym, który program ma za zadanie poprawnie określić jest pole jakościowe “diagnoza”, mające dwie możliwe wartości: M(od malignant - złośliwy) lub B (od benign - łagodny). Żaden rekord zbioru nie zawiera pustych elementów.

Zbiór danych podzielony został na 469 elementów do treningu oraz 100 elementów testowych.

1. Algorytm k-NN

Do napisania tego algorytmu zastosowane zostały dwa pakiety:

- class (zawiera funkcje knn),
- gmodels (zawiera funkcje CrossTable do wyświetlania wyników).

Wartości niektórych atrybutów są dużo większe (nawet 1000 razy) niż inne. Jest to problem, gdyż algorytm KNN opiera się na obliczaniu odległości, toteż atrybuty o większej wartości miałyby większa wagę przy obliczaniu atrybutu decyzyjnego. Aby tego uniknąć, kod zawiera funkcję normalizującą - “normalize” która skaluje wartości atrybutów do rangi 0 - 1.

Wbudowana funkcja knn() użyta do przeprowadzenia algorytmu składa się z 4 parametrów:

- train - zawiera zbiór elementów treningowych
- test - zawiera zbiór elementów testowych
- class - to wartości atrybutu decyzyjnego “diagnosis” dla zbioru elementów treningowych
- k - określa ilość najbliższych sąsiadów analizowanych by przewidzieć wartość atrybutu decyzyjnego.

Funkcja zwraca wartości wyliczonych atrybutów decyzyjnych dla zbioru testowego. Do wyliczeń stosuje ona wzór Euklidesa na miarę odległości. Aby łatwo je zwizualizować i porównać, użyta została funkcja CrossTable() przedstawiona poniżej.

Na początku do funkcja knn() przekazana została wartość k równa pierwiastkowi z ilości elementów treningowych: $k=\sqrt{469}$ czyli $k\approx 21$.

mydata_test_labels	mydata_test_pred_21 Benign	Malignant	Row Total
Benign	77	0	77
	1.000	0.000	0.770
	0.975	0.000	
	0.770	0.000	
Malignant	2	21	23
	0.087	0.913	0.230
	0.025	1.000	
	0.020	0.210	
Column Total	79	21	100
	0.790	0.210	

Fig. 1 Wizualizacja funkcji CrossTable() dla wyliczonego KNN dla k=21.

Tabela przedstawiona powyżej pokazuje wynik w następujący sposób:

- lewy górny róg - liczba łagodnych nowotworów poprawnie wykrytych jako łagodne,
- prawy górny róg - liczba łagodnych nowotworów źle wykrytych jako złośliwe,
- prawy dolny róg - liczba nowotworów złośliwych poprawnie wykrytych jako złośliwe,
- lewy dolny róg - liczba nowotworów złośliwych źle wykrytych jako łagodne.

Funkcja knn() została policzona dla innych wartości k, aby zoptymalizować i wybrać najlepsze k. Wartości to kolejne liczby nieparzyste, aby uniknąć sytuacji remisu. Poniżej przedstawione są dwa wykresy pokazujące liczbę błędnych szacunków programu dla różnych wartości k.

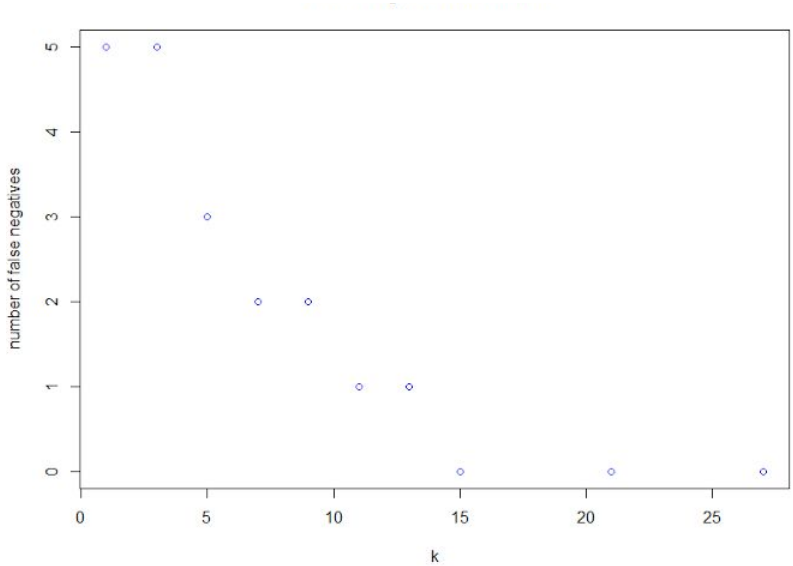

Fig. 2 Liczba łagodnych nowotworów źle wykrytych jako złośliwe dla danego k.

Z powyższego wykresu można zauważyć, że najmniejsza liczba błędów równa 0 występuje dla k=15, 21 lub 27.

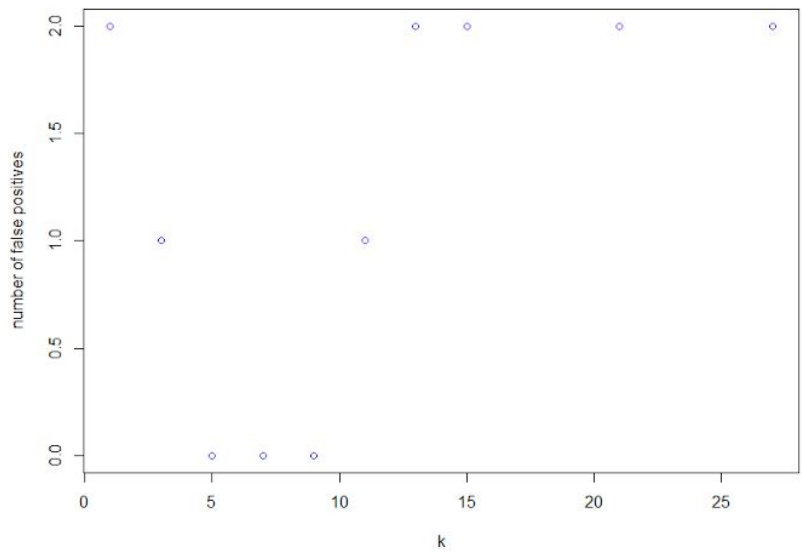

Fig. 3 Liczba złośliwych nowotworów źle wykrytych jako łagodne dla danego k.

Z powyższego wykresu można zauważyć, że najmniejsza liczba błędów równa 0 występuje dla k=5, 7 lub 9. Jest to informacja ważniejsza z punktu widzenia dobra pacjenta. Jeśli algorytm uzna nowotwór za złośliwy, gdy w rzeczywistości jest on tylko łagodny, pacjent dowie się o tym w toku kolejnych specjalistycznych badań. Natomiast, jeśli złośliwy nowotwór zostanie uznany za łagodny, lekarze mogliby odpuścić dalsze kroki lecznicze, jednocześnie bardzo zagrażając zdrowiu oraz życiu pacjenta.

W związku z tym analizując ponownie wykres z Fig. 2 można zdecydować iż k = 7 oraz 9 jest najoptymalniejszą wartością, gdyż pozwala uzyskać minimalną liczbę błędu (2%) przy jednoczesnej zerowej omyłce w diagnozie nowotworu złośliwego.

2. Algorytm klasyfikator bayesowski

Do napisania tego algorytmu zastosowane zostały pakiety:

- e1071 (zawiera funkcje naiveBayes()),
- caret (zawiera funkcje confusionMatrix())

Tym razem zbiór danych nie wymaga normalizacji, gdyż atrybuty są porównywane do siebie, natomiast po normalizacji jedynie pokaże je inaczej przeskalowane, jednakże prawdopodobieństwo się nie zmieni.

Wbudowana funkcja naiveBayes() użyta do przeprowadzenia algorytmu składa się z 3 parametrów:

- train- zawiera elementy treningowe zbioru,
- class- zawiera elementy atrybutu decyzyjnego będącego klasyfikatorem,
- laplace - określa estymację Laplace.

Funkcja zwraca model Bayes'a który dalej jest używany do "oszacowania" za pomocą funkcji predict() składającej się z 3 parametrów:

- model - będący modelem otrzymanym za pomocą naiveBayes(),
- test - będący próbką elementów przeznaczoną do testów,
- type - określający format oszacowań (wartość "class" zwraca rzeczywiste wartości atrybutu decyzyjnego, a nie stopień prawdopodobieństwa).

Estymator Laplaca jest przydatny, gdy wystąpienie danego atrybutu nigdy nie zaszło, a co za tym idzie prawdopodobieństwo zostanie wtedy wyliczone na 0. Estymator dodaje wartości do tabeli częstotliwości, tak by dla żadnego atrybutu prawdopodobieństwo nie było równe 0. Defaultowo estymator Laplaca jest równy 1 (tak więc każda kombinacja atrybut-klasa występuje przynajmniej raz). Dla zastosowanych danych jednak manipulowanie wartościami tego parametru nie przynosi lepszych efektów.

Ponieważ badany zbiór danych jest numeryczny (ciągły) klasyfikator Bayes'a dla każdego badanego atrybutu generuje dwa rozkłady normalne (tzw. rozkłady Gausa) - dla atrybutów decyzyjnych czyli nowotworu łagodnego oraz złośliwego. Rozkłady te można przedstawić za pomocą wykresu. Poniżej przedstawiony jest rozkład normalny atrybutu "średniej konsystencji" nowotworu.

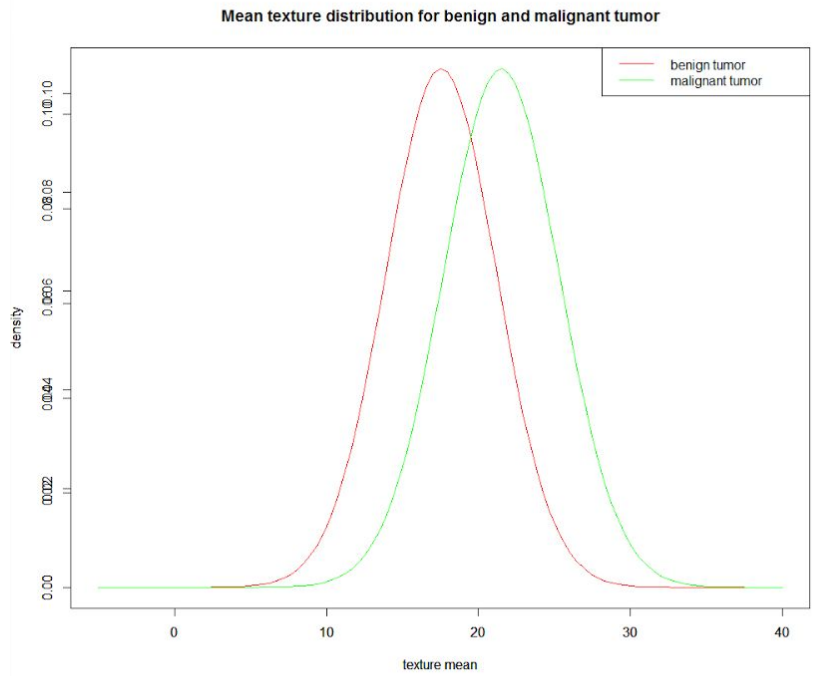

Fig. 5 Rozkład Gausa dla atrybutu "średniej konsystencji" nowotworu.

W tabeli poniżej widać skuteczność algorytmu. Pokazuje on, że 6 nowotworów łagodnych zostało ocenionych na złośliwy oraz 2 nowotwory złośliwe potraktowano jako łagodne.

mydata_test$diagnosis	predictor B	M	Row Total
B	71	6	77
	0.922	0.078	0.770
	0.973	0.222	
	0.710	0.060	
M	2	21	23
	0.087	0.913	0.230
	0.027	0.778	
	0.020	0.210	
Column Total	73	27	100
	0.730	0.270	

Fig. 4 Wizualizacja funkcji CrossTable() dla wyliczonego algorytmu.

3. SVM

Do napisania tego algorytmu zastosowany został pakiet kernlab (zawiera funkcje ksvm).

Wbudowana funkcja ksvm() użyta do przeprowadzenia algorytmu składa się z 4 parametrów:

- target - zawiera nazwę atrybutu decyzyjnego,
- data - zawiera zbiór elementów treningowych,
- kernel - określa sposób mapowania (liniowy, wielomianowy etc),
- c - określa koszt naruszenia granic przez funkcję.

Funkcja zwraca obiekt SVM który dalej jest używany do “oszacowania” za pomocą funkcji predict() składającej się z 3 parametrów:

- model - będący modelem SVM otrzymanym za pomocą ksvn(),
- test - będący próbką elementów przeznaczoną do testów,
- type - określający format oszacowań (wartość “response” zwraca rzeczywiste wartości atrybutu decyzyjnego, a nie stopień prawdopodobieństwa).

Tak potraktowane dane łatwo przeanalizować za pomocą funkcji Crosstable(), która prezentuje trafność algorytmu.

ydata_test$diagnosis	pred B	M	Row Total
B	76	1	77
	0.987	0.013	0.770
	0.987	0.043	
	0.760	0.010	
M	1	22	23
	0.043	0.957	0.230
	0.013	0.957	
	0.010	0.220	
Column Total	77	23	100
	0.770	0.230	

Fig. 6 Wizualizacja funkcji CrossTable() dla wyliczonego SVM dla liniowego mapowania.

Funkcja ksvn() została użyta dla 3 różnych wartością parametru kernel odpowiedzialnego za mapowanie elementów, aby znaleźć ten dające najlepsze rezultaty.

Zarówna dla mapowania liniowego jak i wielomianowego wynik był najlepszy (widoczny powyżej), natomiast dla bardziej zaawansowanych metod takich jak RBF (radialna funkcja bazowa) rezultaty były gorsze (aż 4, zamiast jak poprzednio 1, nowotwory złośliwe zostały uznane za łagodne).

Wnioski:

Tabela poniżej przedstawia porównanie najlepszych wyników trzech algorytmów. Najważniejszym czynnikiem jest tu liczba błędów tzw. false negatives, gdyż oznacza to, iż nowotwór złośliwy został zaklasyfikowany jako łagodny. W przypadku pacjenta, ma to ogromne znaczenia, bo lekarz może zaprzestać dalszego leczenie, które jest niezbędne aby pacjent mógł dalej żyć. W przypadku false positives, kiedy zdiagnozowany zostanie nowotwór złośliwy, podczas dalsze badania powinny wykazać iż jest on jednak łagodny, dlatego błędy w tym przypadku mają mniejsze znaczenie.

Dlatego też algorytm k-NN okazał się być najskuteczniejszym przy analizie tego zbioru danych, co potwierdza tabela poniżej.

algorytm	**liczba tzw. false positives**	**liczba tzw. false negatives**	**najlepszy wybór (1-najlepszy, 3 - najgorszy)**
k-NN	2	0	1
klasyfikator bayesowski	6	2	3
SVM	1	1	2