MOW - dokumentacja końcowa

Katarzyna Dziewulska, Łukasz Szymczyk

14 stycznia 2019

1 Treść zadania

Przewidywanie spożycia alkoholu przez studentów. Połączenie grupowania (z 2 atrybutów dyskretnych (porządkowych) chcemy uzyskać 2 klasy) oraz klasyfikacji (budowa i ocena drzewa decyzyjnego).

2 Interpretacja tematu projektu

Dostępny jest zbiór danych zawierający informacje o grupie studentów takie jak wiek, szkoła itp. będące atrybutami o wartościach dyskretnych. Informacja o spożyciu alkoholu zawiera się w dwóch atrybutach porządkowych: `Walc` - weekendowe spożycie alkoholu przez studentów, oraz `Dalc` - spożycie alkoholu w dniach pracujących. Na podstawie tych dwóch atrybutów, za pomocą wybranego algorytmu grupowania zbiór danych zostanie podzielony na dwie grupy: studentów spożywających dużo alkoholu i studentów spożywających go mniej. Celem zadania jest przewidywanie spożycia alkoholu przez studentów, zatem w dalszej części zadania ze zbioru danych wykluczone zostaną atrybuty `Dalc` i `Walc`, a zamiast nich informację o spożyciu alkoholu przez studenta nieść będzie jego przynależność do jednej z dwóch, wcześniej utworzonych na podstawie tych atrybutów grup. Zbiór danych zostanie podzielony na zbiór trenujący i testowy. Na podstawie zbioru trenującego utworzony zostanie model klasyfikatora gdzie klasami są dwie grupy: studenci spożywający dużo alkoholu i studenci spożywający go mniej. Następnie klasyfikator zostanie sprawdzony na zbiorze testowym przydzielając studenta do jednej z klas - przewidując czy jest on studentem spożywającym dużo alkoholu, czy mało. Ponieważ wiemy jaki był przydział studenta do grupy przez algorytm grupujący możemy zweryfikować dokładność naszego klasyfikatora tworząc macierz pomyłek (*ang. confusion matrix*).

Wykorzystywane w zadaniu dane pochodzą ze strony `www.kaggle.com`. Badania zostaną przeprowadzone z wykorzystaniem pakietu R i gotowych algorytmów dostępnych w tym pakiecie.

3 Opis algorytmów

Do wykonania zadania zostaną wykorzystane trzy różne algorytmy grupowania dostępne w pakiecie R, należące do trzech różnych metod grupowania. Z grupy metod opartych na podziałach wykorzystany zostanie algorytm k-średnich (*ang. k-means*), z grupy metod hierarchicznych algorytm aglomeracyjny agnes (*ang. aglomerative nesting*), zaś z grupy metod opatych na gęstościach

algorytm dbscan *(ang. density based spatial clustering)*. Do klasyfikacji zostanie wykorzystana metoda oparta na drzewach decyzyjnych. Poniżej przedstawiono krótki opis każdego z algorytmów.

3.1 Algorytm k-średnich

Algorytm przyporządkowuje zbiór przykładów do przyjętej a priori liczby grup. Każda grupa jest reprezentowana przez środek ciężkości obiektów w grupie - centroid. Kolejne kroki algorytmu przedstawiono poniżej.

Algorithm 1: Algorytm k-średnich

1. Wyznacz położenie początkowe centroidów (najczęściej losowe, centroidów jest tyle, na ile grup chcemy pogrupować zbiór danych).
2. Przyporządkuj każdej obserwacji najbliższy jej centroid (najczęściej odległość euklidesowa).
3. Kiedy wszystkie obserwacje zostaną przyporządkowane, wyznacz ponownie położenie centroidów. Nowe położenie centroidów to środek ciężkości grupy.
4. Powtarzaj kroki 2 i 3 dopóki centroidy zmieniają swoje położenie lub nie zostanie osiągnięta maksymalna liczba iteracji.

Zaletą algorytmu jest niska złożoność, a więc wysoka wydajność. Natomiast wadą jest konieczność z góry określenia liczby grup, jednak w przypadku podanego zadania nie ma to znaczenia, bo wiadomo, że dzielimy zbiór na dwie grupy. Wadą również jest to, że wynik grupowania zmienia się w zależności od różnego położenia początkowego centroidów. Algorytm nie sprawdza się kiedy grupy nie są wypukłe.

3.2 Algorytm hierarchiczny

Algorytmy hierarchiczne mają na celu zbudowanie hierarchii klastrów (graficzna reprezentacja hierarchii ma postać drzewa). Dzielą się na metody aglomeracyjne i rozdzielające. W metodach aglomeracyjnych każda obserwacja tworzy na początku jednoelementowy klaster. Następnie pary klastrów są scalane. W każdej iteracji algorytmu łączone są ze sobą dwa najbardziej zbliżone klastry. W metodach rozdzielających początkowo wszystkie obserwacje znajdują się w jednym klastrze. W następnych krokach klastry dzielone są na mniejsze i bardziej jednorodne. Do wykonania zadania wybrano algorytm agnes dostępny w pakiecie R, należący do metod aglomeracyjnych, którego kroki przedstawiono poniżej.

Algorithm 2: Algorytm aglomeracyjny

1. Utwórz tyle jednoelementowych grup ile jest obserwacji.
2. Znajdź dwie najbliższe (w sensie przyjętej miary np. euklidesowa) grupy.
3. Scal ze sobą powyższe grupy zmniejszając liczbę grup o jeden.
4. Powtarzaj kroki 2 i 3, tak długo aż utworzy się jedna grupa zawierająca wszystkie próbki (można na początku ustalić końcową większą liczbę grup i skończyć algorytm wcześniej po osiągnięciu ich liczby).

Zaleta algorytmów hierarchicznych jest brak konieczności początkowego określania liczby klas. Wadą są zróżnicowane wyniki w zależności od wybranych metod łączenia grup.

3.3 Algorytm DBSCAN

W algorytmie tym wykorzystywane są dwa parametry ϵ - promień definiujący otoczenie obiektu oraz $MinPts$ - minimalna liczba punktów w otoczeniu ϵ. Rdzeniem nazywamy obiekt, który ma co najmniej$MinPts$ w otoczeniu ϵ, a obiektem brzegowym nazywamy obiekt, który ma mniej niż $MinPts$ w otoczeniu ϵ. Mówimy, że obiekt x jest bezpośrednio gęstościowo osiągalny z obiektu y jeżeli y jest rdzeniem oraz odległość x od y jest mniejsza od ϵ. Mówimy, że obiekt x jest gęstościowo osiągalny z obiektu y, jeśli istnieje łańcuch obiektów $p_1, p_2, ..., p_n$ taki, że $p_1 = y$, $p_n = x$ oraz p_{i+1} jest bezpośrednio gęstościowo osiagalny z p_i $(1 <= i < n)$. Kolejne kroki algorytmu zostały przedstawione poniżej.

Algorithm 3: Algorytm DBSCAN

1. Rozpoczynamy od dowolnego obiektu x.
2. Jeżeli w otoczeniu ϵ obiektu x znajduje się co najmniej $MinPts$ obiektów to tworzony jest klaster C i wszystkie obiekty gęstościowo osiągalne z obiektu x są dołączane do klastra C. W przeciwnym razie obiekt x nie jest rdzeniem, wracamy do punktu 1 i wybieramy następny obiekt ze zbioru.
3. Proces grupowania jest kontynuowany tak długo, aż zostaną przetworzone wszystkie obiekty ze zbioru. Obiekty, które nie zostały zaklasyfikowane do żadnego z klastrów tworzą zbiór punktów osobliwych.

W przypadku algorytmu DBSCAN zaletą jest to, że można uzyskać bardziej skomplikowane kształty klastrów niż w przypadku algorytmu k-średnich. Wadą natomiast jest to, że wyniki zależą od kolejności w jakiej przeglądane są dane.

3.4 Drzewa decyzyjne

Jako algorytm klasyfikacji opartej na drzewach decyzyjnych zostanie wykorzystany algorytm $ID3$ lub jego ulepszona wersja z przycinaniem $C4.5$. Poniżej przedstawiono ogólny algorytm $ID3$ uczenia się drzewa T ze zbioru uczącego S.

Algorithm 4: Buduj_Drzewo (P,d,S)

1. Wybierz najlepszy atrybut A (najlepiej różnicujący przykłady ze zbioru S).
2. Rozbuduj drzewo poprzez dodanie do węzła nowych gałęzi odpowiadającym poszczególnym wartościom $v_1, v_2, ..., v_n$ atrybutu A.
3. Podziel S na podzbiory $S_1, S_2, ..., S_n$ odpowiadające wartościom v_1, v_2, v_n atrybutu A i przydziel te podzbiory do odpowiednich gałęzi rozbudowanego drzewa.
4. Jeżeli wszystkie przykłady w S_i należą do tej samej klasy C zakończ gałąź liściem wskazującym C. W przeciwnym przypadku rekurencyjnie buduj drzewo T_i dla S_i (tzn. powtórz kroki 1-4).

Zaletą drzew decyzyjnych jest możliwość reprezentowania dowolnie złożonych pojęć oraz efektywność pamięciowa. Wadą jest natomiast to, że dla pewnych zadań i miar drzewa mogą rosnąć eksponencjalnie.

4 Opis i wyniki badań

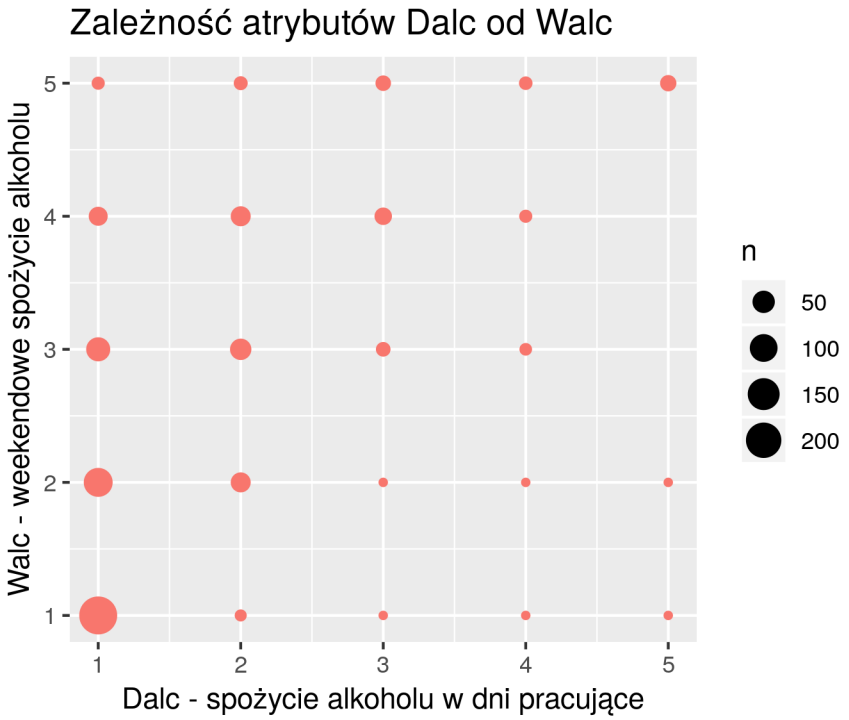
4.1 Zbiór danych

Badania przeprowadzone zostały na danych zebranych wśród studentów uczęszczających na zajęcia z języka portugalskiego, znajdujące się w pliku `student_por.csv`. Plik zawiera 33 atrybuty dyskretne opisujące 649 studentów (rekordów). Atrybuty te to między innymi wiek, płeć, czas nauki, itd. Dane pochodzą z platformy `www.kaggle.com` i zostały zebrane wśród studentów uczących się języka portugalskiego oraz matematyki. Zdecydowaliśmy się wykorzystać tylko plik zawierający studentów uczęszczających na portugalski, gdyż zawiera on prawie dwa razy więcej rekordów, a według informacji zamieszczonych na stronie 382 z 395 studentów uczęszczających na matematykę uczęszcza również na portugalski. Atrybuty w obu plikach są takie same, za wyjątkiem 3 atrybutów dotyczących ocen z danego przedmiotu w ciągu odbytych lat nauki, ponieważ są to oceny odpowiednio z portugalskiego lub matematyki.

4.2 Przeprowadzone eksperymenty

Pierwszym etapem realizacji projektu było pogrupowanie studentów z wykorzystaniem trzech wybranych wcześniej algorytmów grupowania dostępnych w pakiecie R. Klasteryzacja wykonana została na podstawie dwóch atrybutów: `Walc` - weekendowe spożycie alkoholu przez studentów i `Dalc` - dzienne spożycie alkoholu przez studentów. Atrybuty te przyjmują wartości w skali: 1-niskie spożycie alkoholu do 5-wysokie spożycie alkoholu. W wyniku grupowania studenci podzieleni zostali na dwie grupy: studentów pijących więcej alkoholu - "pijaków" oraz studentów pijących umiarkowane ilości alkoholu - "nie pijaków". Parametry algorytmów były dobierane w taki sposób aby podział na grupy był jak najbardziej logiczny. Następnie atrybuty wykorzystywane w grupowaniu zostały wyłączone ze zbioru danych, a zamiast nich każdy student przyporządkowany został do jednej z klas: "pijak" - klasa 1, "nie pijak" - klasa 0. Zbiór danych podzielony został na trenujący i testowy za pomocą walidacji krzyżowej. Następnie zbudowano model klasyfikatora z wykorzystaniem drzewa decyzyjnego i sprawdzono jego jakość.

Na poniższym wykresie zamieszczono zależność atrybutu `Walc` od `Dalc` wraz z licznością studentów.



Rysunek 1: Zależność atrybutów Dalc od Walc wraz z liczbą studentów.

4.3 Grupowanie algorytmem k-means

Do wykonania grupowania algorytmem k-means wykorzystana została funkcja z pakietu R: `kmeans(x, centers, iter.max, nstart, algorithm, trace=FALSE)`, gdzie:

- `x` - macierz z danymi do grupowania,
- `centers` - liczba skupień które chcemy wyróżnić w danych, albo podane początkowe centra skupień (jeśli podana zostanie liczba skupień wtedy ich początkowe centra wybierane są losowo),
- `iter.max` - maksymalna liczba iteracji,
- `nstarts` - jeśli w centers podano liczbę grup to parametr ten określa liczbę różnych losowych środków branych pod uwagę w grupowaniu,
- `algorithm` - jaki algorytm spośród ("Hartigan-Wong", "Lloyd", "Forgy", "MacQueen") wykorzystany zostanie w grupowaniu.

Badania przeprowadzone zostały dla następujących wartości parametrów:

- jako środki najpierw przyjęte zostały punkty: (1,1) - dla studentów "nie pijaków" oraz (3,4) - dla studentów "pijaków". Obserwując dane na Rys.1 takie rozłożenie środków ciężkości grup

sensownie dzieliłoby zbiór danych na te dwie grupy. Następnie sprawdzone zostało działanie algorytmu dla losowych środków - wartość parametru to liczba grup - 2.

- maksymalna liczba iteracji: 1, 10, 100, 500, 1000, 5000
- kiedy parametr centers przyjęty był 2, to nstarts przyjmowano kolejno: 1, 3, 5, 10
- sprawdzono działanie algorytmu dla każdej z możliwych wartości parametru algorithm

4.3.1 Walidacja krzyżowa

K-krotna walidacja krzyżowa (ang. *k-fold cross-validation*) polega na losowym podziale zbioru na k podzbiorów, gdzie jeden z podzbiorów wykorzystany jest jako zbiór testowy, natomiast pozostałe k-1 tworzą zbiór treningowy. Procedura ta jest powtarzana k razy, dzięki temu każdy element zbioru jest użyty zarówno do testów jak i treningu (w różnych iteracjach). Na końcu k cykli uśrednia się błędy otrzymując miarę jakości modelu. Błąd klasyfikatora obliczany jest według wzoru :

$$\epsilon = \frac{N_\mathrm{t} - N_\mathrm{c}}{N_\mathrm{t}} \cdot 100\% \tag{1}$$

gdzie: N_c to liczba poprawnie sklasyfikowanych przykładów, a N_t to liczba wszystkich testowanych przykładów.

Miara jakości modelu (uśrednione błędy):

$$\epsilon_{\acute{\mathrm{s}}\mathrm{r}} = \frac{1}{k} \sum_{n=1}^{k} \epsilon_\mathrm{n} \tag{2}$$

4.4 Ocena jakości klasyfikatora

W celu oceny jakości klasyfikatora zbudowana zostanie macierz pomyłek (*ang. confussion matrix*).

Tabela 1: Macierz pomyłek.

	klasa rzeczywista	
klasa predykowana	Pozytywna	Negatywna
Pozytywna	TP	FN
Negatywna	FP	TN

gdzie :

- TP (ang. *true positive*) - liczba poprawnie sklasyfikowanych przykładów z wybranej klasy
- FN (ang. *false negative*) - liczba błędnie sklasyfikowanych przykładów z tej klasy tj. decyzja negatywna podczas gdy przykład w rzeczywistości jest pozytywny
- TN (ang. *true negative*) - liczba przykładów poprawnie nie przydzielonych do wybranej klasy (poprawnie odrzuconych)

- FP (ang. *false positive*) - liczba przykładów błędnie przydzielonych do wybranej klasy, podczas gdy w rzeczywistości do niej nie należą

wrażliwość/czułość - określa ułamek zidentyfikowanych wystąpień klasy pozytywnej w całym zbiorze (ignoruje fałszywe pozytywy, czyli wystąpienia negatywne, które zostały zidentyfikowane jako pozytywne) = $\frac{TP}{TP+FN}$

specyficzność - określa zdolność klasyfikatora do wykrywania klasy negatywnej = $\frac{TN}{FP+TN}$