Laboratorium 2

Klasyfikacja

Wstęp

Chcemy wyręczyć botaników w rozpoznawaniu kwiatów, więc piszemy program ekspercki, który na podstawie 4 parametrów numerycznych kwiatu odgadnie jego gatunek.

Sepal.length	Sepal.width	Petal.length	Petal.width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
5.6	3.0	4.5	1.5	versicolor
5.8	2.7	4.1	1.0	???

Program musi na podstawie parametrów ustalić gatunek irysa, czyli go sklasyfikować. Dlatego cały algorytm nazywamy klasyfikatorem, procedurę zwiemy klasyfikacją. Przyjmuje się, że ostatnia kolumna/parametr to klasa (u nas to gatunek, Species), a wcześniejsze kolumny/parametry to „dane wejściowe".

Załóżmy, że napisaliśmy jakiś program klasyfikujący. Jak sprawdzić czy klasyfikator dobrze działa (poprawnie odgaduje odpowiedzi)? Odpowiedź jest prosta: sprawdzić jakich odpowiedzi udzieli, dla rekordów, dla których już znamy odpowiedzi. Następnie weryfikujemy ile odpowiedzi się już zgadza. Taki zabiega nazywamy ewaluacją klasyfikatora.

Uwaga! Zbiór rekordów tabeli, na podstawie której tworzymy algorytm nazywamy zbiorem treningowym. Zbiór rekordów bazy, na których dokonujemy ewaluacji nazywamy zbiorem testowym. Często przyjmuje się, że zbiór treningowy i testowy nie nachodzą na siebie, jednak w zadaniu 1 przyjmiemy, że zbiór treningowy i testowy to cała baza danych.

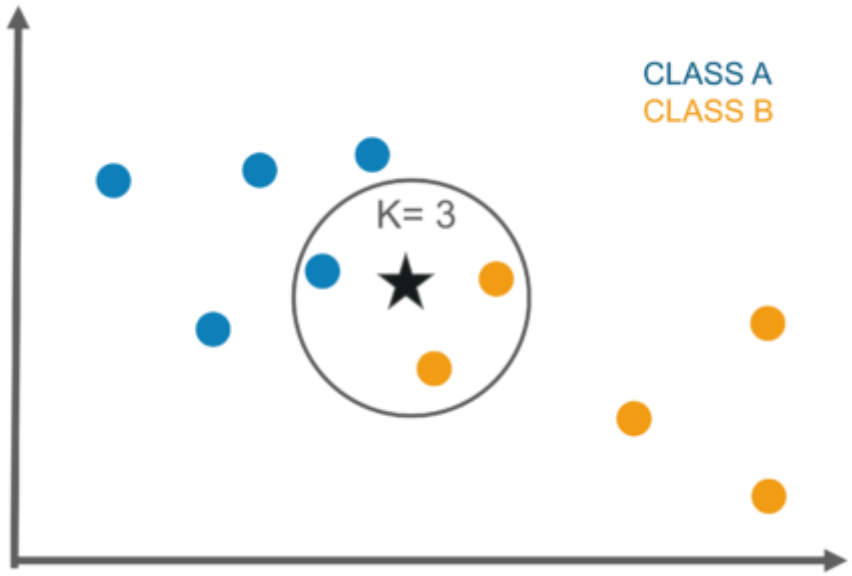
Małe binarne drzewo decyzyjne możemy wyobrazić sobie następująco:

Są algorytmy, takie jak ID3 czy C4.5, które tworzą takie drzewa automatycznie i to z o wiele większą precyzją niż człowiek. Zarówno w języku R, jak i w Pythonie są paczki, które oferują owe algorytmy.

- Język R:
  - Party https://rpubs.com/njvijay/14899
  - Tree lub Rpart: https://davetang.org/muse/2013/03/12/building-a-classification-tree-in-r/
- Python:
  - Sklearn (tree) https://scikit-learn.org/stable/modules/tree.html, lub https://medium.com/@haydar_ai/learning-data-science-day-21-decision-tree-on-iris-dataset-267f3219a7fa

Istnieje wiele innych klasyfikatorów. Poniżej kolejne dwa.

- Klasyfikator *k* najbliższych sąsiadów (k-nearest neighbors, k-NN). Zasada działania jest prosta. Klasyfikatorowi ustalamy *k*, np. *k*=3. Następnie dla każdego rekordu, który chcemy sklasyfikować, wyszukamy w bazie 3 najbardziej podobne rekordy (np. najbardziej zbliżone pod względem parametrów, metryka euklidesowa lub inna). Nasz klasyfikowany rekord przyjmuje taką klasę, jak większość z tych trzech rekordów.



- Klasyfikator naiwny bayesowski (Naive Bayes) to klasyfikator probabilistyczny. Bierzemy wartość pierwszej danej z badanego rekordu, następnie sprawdzamy czy inne rekordy z tą samą wartością „siedziały" w jednej klasie, czy drugiej. Robimy tak dla wszystkich kolejnych wartości. Następnie wykorzystując wzór Bayesa na prawdopodobieństwo warunkowe oraz zakładając, że dane z kolumn są niezależne (stąd ta niemądra naiwność) szacujemy, w której klasie znajduje się najprawdopodobniej nasz rekord.
  Przykład: rekord ma trzy dane $[x_1\ x_2\ x_3]$ i chcemy sprawdzić czy przyjmuję klasę *yes* czy *no*. Obliczamy poniższe prawdopodobieństwa i wybieramy większe z nich.

$$P(yes \mid x_1\ x_2\ x_3) = P(x_1 \mid yes) \cdot P(x_2 \mid yes) \cdot P(x_3 \mid yes) \cdot P(yes)$$
$$P(no \mid x_1\ x_2\ x_3) = P(x_1 \mid no) \cdot P(x_2 \mid no) \cdot P(x_3 \mid no) \cdot P(no)$$

Zadanie 1

Załóżmy, że mamy małą bazę danych osób, które decydują się (lub nie) na kupno komputera. Parametry tych osób to wiek, dochód, bycie studentem, zdolność kredytowa. Klasa „buys" odpowiada na pytanie: „czy osoba kupuje komputer?".

age	income	student	credit.rating	buys
31..40	high	no	fair	yes
>40	medium	no	fair	yes
>40	high	yes	excellent	yes
>40	low	yes	excellent	no
31..40	low	no	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	no

Dokonaj klasyfikacji naiwnej bayesowskiej na podanym niżej rekordzie Y. Dla ułatwienia po lewej stronie podano obliczenia na rekordzie X.

Rekord X

>40	medium	no	excellent	???

Krok.1 Obliczamy prawdopodobieństwo obu klas w bazie danych:

P(buys=yes)=4/7 P(buys=no)=3/7

Krok.2 Obliczamy prawdopodobieństwa warunkowe biorąc pod uwagę dane z niewiadomego rekordu.

P(age>40|buys=yes)=2/4 (wśród osób kupujących komputer liczymy osoby starsze niż 40)
P(age>40|buys=no)=1/3 (jest jedna osoba 40+ wśród 3 osób niekupujących komputera)
P(income=medium|buys=yes)=1/4
P(income=medium|buys=no)=1/3
P(student=no|buys=yes)=3/4
P(student=no|buys=no)=1/3
P(credit.rating=excellent|buys=yes)=2/4
P(credit.rating=excellent|buys=no)=1/3

Krok.3 Korzystamy z naiwnego wzoru Bayesa. Mnożymy prawdopodobieństwa warunkowe z prawdopodobieństwami klas. Osobno dla każdej klasy.

P(X|buys=yes) = (2/4) * (1/4) * (3/4) * (2/4) = 3/64
P(X|buys=no) = (1/3) * (1/3) * (1/3) * (1/3) = 1/81
P(buys=yes|X)=P(X|buys=yes)*P(buys=yes) = (3/64)*(4/7) = 0.02679
P(buys=no|X)=P(X|buys=no)*P(buys=no) =(1/81)*(3/7) = 0.00529

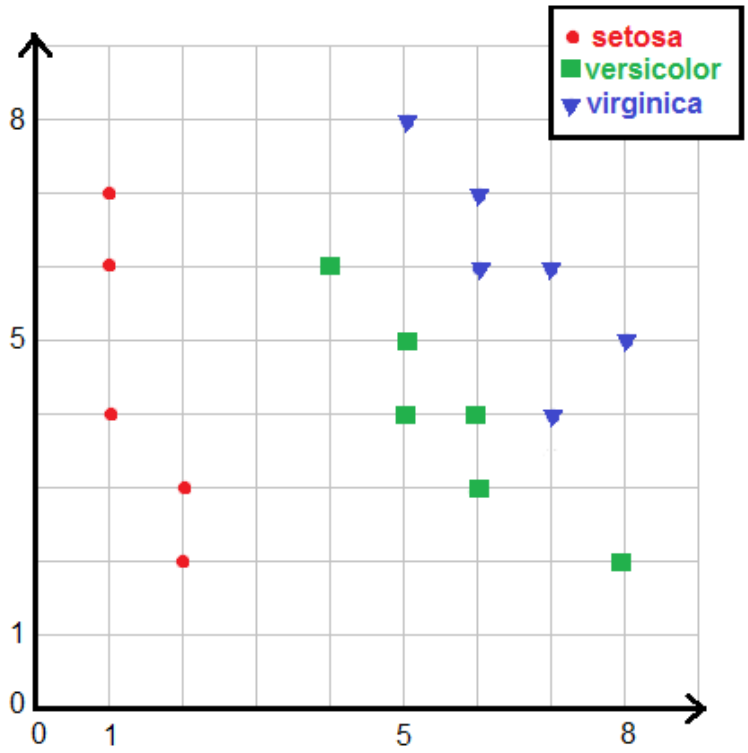
Z obu wartości większa jest 0.02679, więc nasz rekord przyjmuje klasę yes.

Rekord Y

>40	low	no	fair	???

Zadanie 2

Chcemy sprawdzić jak klasyfikator k najbliższych sąsiadów poradzi sobie na małym zbiorze irysów opisanych tylko przez dwa parametry. Przyjmujemy, że miarą podobieństwa jest zwykła odległość euklidesowa. Zbiór treningowy klasyfikatora jest podany w formie graficznej na wykresie. Jak poradzą sobie klasyfikator 1 i 3 najbliższych sąsiadów? Który będzie lepszy? Dokonaj ewaluacji na poniższym zbiorze testowym.



Zbiór testowy:

X	Y	Klasa (prawdziwa)	Klasa przewidywana: 1 najbliższy sąsiad	Klasa przewidywana: 3 najbliższych sąsiadów
2.7	6	versicolor		
5	7	virginica		
7	3.5	versicolor		
9	3	virgnica		
2	5	setosa		

Ewaluacja klasyfikatorów:

Klasyfikator 1 najbliższy sąsiad

Dokładność: ..... %

Macierz błędu:

		prawdziwa		
		Ve	Vi	Se
przewidywana	Ve			
	Vi			
	Se			

Klasyfikator 3 najbliższych sąsiadów

Dokładność: ..... %

Macierz błędu:

		prawdziwa		
		Ve	Vi	Se
przewidywana	Ve			
	Vi			
	Se			

Zadanie 3

W załączonym zbiorze danych diabetes.csv znajdują się dane kobiet indiańskiego pochodzenia z USA, które zachorowały lub nie zachorowały na cukrzycę. Klasyfikator ma na celu diagnozowanie choroby na podstawie parametrów medycznych kobiety. Sprawdź jak działają poznane klasyfikatory na tej bazie danych. Dokonaj porównania:

- k-NN, k=3
- k-NN, k=5
- k-NN, k=11
- Naiwny bayesowski.
- Drzewa decyzyjne.

W rozwiązaniu zadania uwzględnij następujące punkty:

a) Podziel w losowy sposób bazę danych na zbiór treningowy (67%) i testowy (33%).

b) Uruchom każdy z klasyfikatorów wykorzystując paczki i dokonaj ewaluacji ma zbiorze testowym wyświetlając procentową dokładność i macierz błędu.

R	Python
Naive Bayes: http://ugrad.stat.ubc.ca/R/library/e1071/html/naiveBayes.html https://www.rdocumentation.org/packages/e1071/versions/1.7-2/topics/naiveBayes https://www.r-bloggers.com/understanding-naive-bayes-classifier-using-r/ k-NN: https://www.rdocumentation.org/packages/class/versions/7.3-15/topics/knn https://towardsdatascience.com/k-nearest-neighbors-algorithm-with-examples-in-r-simply-explained-knn-1f2c88da405c	Naive Bayes: https://www.datacamp.com/community/tutorials/naive-bayes-scikit-learn https://scikit-learn.org/stable/modules/naive_bayes.html k-NN: https://towardsdatascience.com/k-nearest-neighbor-python-2fccc47d2a55 https://scikit-learn.org/stable/modules/neighbors.html

c) Nanieś wszystkie dokładności klasyfikatorów na wykres słupkowy. Każdy słupek odpowiada jednemu klasyfikatorowi, a wysokość słupka to jego dokładność procentowa. Jeśli trzeba to dodaj legendę.

d) Pytanie dodatkowe:
Chcemy zminimalizować błędy, gdy klasyfikator chore osoby klasyfikuje jako zdrowe (i odsyła do domu bez leków). Który z klasyfikatorów najbardziej się do tego nadaje?