Usuwanie wartości brakujących

Wstęp

Jednym z poważnych problemów budowy systemów predykcyjnych jest możliwość występowania wartości brakujących. Wartości brakujące są to nieznane wartości niektórych atrybutów w zbiorze danych używanym zarówno dla treningu jak i w trakcie predykcji. Źródłem wartości brakujących mogą być np. awarie pojedynczych czujników w systemie pomiarowych, nietypowe wartości pojawiające się na wejściu czujników pomiarowych, duży koszt wykonania danych pomiarów, co powoduje, że nie dla każdej próbki są one wykonywane.

Lp.	label	att1	att2	att3	att4
1	c1	5.1	3.5	1.4	0.2
2	c1	N1	3	1.4	0.2
3	c2	4.7	3.2	1.3	0.2
4	c2	4.6	3.1	1.5	0.2
5	c1	5	3.6	N2	0.2
6	c2	5.4	3.9	1.7	0.4
7	c2	4.6	N3	1.4	0.3

Najprostszym rozwiązaniem wartości brakujących jest zastąpienie ich przez wartość stałą np. 0 lub inną zdefiniowaną przez użytkownika. W tym przypadku każde wystąpienie wartości brakującej spowoduje zamienienie jej na wspomnianą wyżej wartość. Bardziej zaawansowaną metodą jest wykorzystanie średniej arytmetycznej lub mediany. W przypadku wykorzystania wartości średniej konieczne jest policzenie średniej arytmetycznej danego atrybutu :

N1 = średnia(5.1;4.7;4.6;5;5.4;4.6) => 4.9

Lub poprzez medianę

N1 = mediana(5.1;4.7;4.6;5;5.4;4.6) => 4.85

Lub tez odpowiednio inny typ średniej (np. średnią geometryczną)

N1 = średnia.geometryczna(5.1;4.7;4.6;5;5.4;4.6) => 4.891303

I wpisanie jej w każde brakujące miejsce dla danego atrybutut.

Jak widać w zależności od typu wybranej średniej uzyskiwane będą różne wartości

Poważną wadą powyżej opisanych metod jest fakt iż wartość brakującego atrybutu jest zawsze stała, niezależnie od położenia w przestrzeni. Może to powodować bardzo niebezpieczne zachowania się systemów predykcyjnych prowadząc do błędnej klasyfikacji.

Lp.	label	att1	att2	att3	att4
1	c1	5.1	3.5	1.4	0.2
2	c1	4.8	3	1.4	0.2

3	c2	4.7	3.2	1.3	0.2
4	c2	4.8	3.1	1.5	0.2
5	c1	4.8	3.6		0.2
6	c2	4.8	3.9	1.7	0.4
7	c2	4.6		1.4	0.3

Innym, bardziej zaawansowanym sposobem radzenia sobie z wartościami brakującymi jest wykorzystanie jakiegoś innego systemu predykcyjnego, którego zadaniem byłoby przewidzenie, jaką powinien mieć wartość brakujący atrybut.

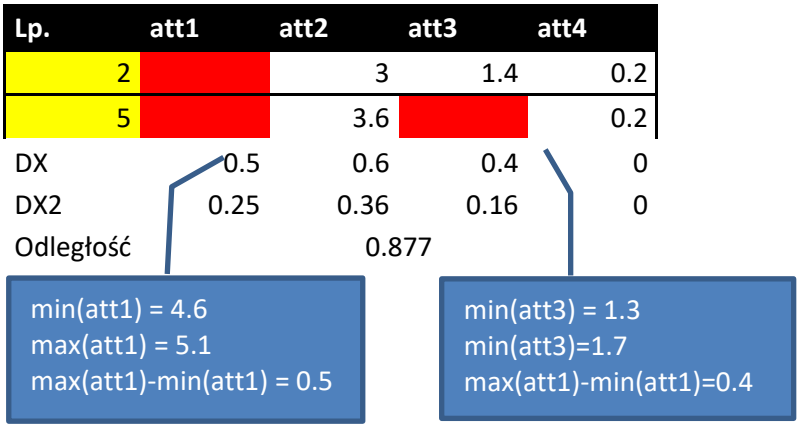
Innymi słowy wartość brakującego atrybutu powinna zostać zastąpiona wartością, która jest najbardziej prawdopodobne ze uwzględnieniem wartości pozostałych atrybutów (czyli wartością która jest lokalnie najbardziej prawdopodobna). Jednym z prostszych rozwiązań jest tutaj wykorzystanie estymacji najbliższego sąsiada.

Metoda ta polega na tym, iż chcąc uzupełnić określoną wartość brakującą, szukamy k wektorów, które są najbardziej podobne do wektora zawierającego wartości brakujące, a następnie w miejsce wartości brakującej wstawiamy średnią arytmetyczną jego k-najbliższych sąsiadów.

Szukanie wektorów najbardziej podobnych odbywa się poprzez wyznaczenie odległości od wektora z wartością brakująca do wszystkich innych wektorów. Przy czym ewentualne występujące braki wartości atrybutów zastępowane są poprzez maksymalną odległość.

Np.:

Chcąc znaleźć wartość N1 z przykładu 1 wyznaczamy odległość pomiędzy wektorem nr 2 a pozostałymi wektorami np. Licząc odległość pomiędzy wektorami 2 i 5 otrzymujemy



Zadania

1) Pobierz paczkę danych ze strony prowadzącego. Paczka zawiera pomocnicze funkcje oraz zbiory danych potrzebne do realizacji zajęć
2) Do usuwania wartości brakujących w kolejnych zadaniach wykorzystywana będzie funkcja *removeMissing* (zajrzyj do kodu funkcji), która w sowim ciele wywołuje odpowiednią funkcję implementująca kolejne sposoby usuwania wartości

brakujących. Twoim zadaniem jest implementacja trzech wyżej opisanych metod usuwania wartości brakujących poprzez implementację brakujących funkcji.

3) Otwórz skrypt missingValuesZad1 – skrypt ten zawiera przykład, w którym zademonstrowano zachowanie się różnych metod usuwania wartości brakujących w tym: poprzez wartość stałą, średnią arytmetyczną i/lub medianę oraz algorytm najbliższego sąsiada, w zastosowaniu do predykcji sieci neuronowej typu MLP. Skrypt porównuje wyżej opisane metody z sytuacją w której mamy do dyspozycji cały zbiór danych bez wartości brakujących.
   Uruchom skrypt i zanotuj uzyskane wyniki dla różnych ustawień parametru S sieci neuronowej (różnej liczby neuronów i warstw)
   Która z metod jest najlepsza i dlaczego?
4) Metodę usuwania wartości brakujących można również wykorzystać do uczenia sieci neuronowych. W tym przypadku usuwanie wartości brakujących wywołujemy przed uczeniem sieci. Sprawdź jak działa funkcja usuwania wartości brakujących podczas uczenia sieci. W tym celu uruchom skrypt missingValuesZad2 i porównaj wyniki dla różnych metod. Obliczenia powtórz dla różnych konfiguracji sieci. Zanotuj wnioski.

Uwagi i wskazówki

1. Każda z funkcji posiada dwa argumenty *dataMiss* oraz *dataRef*. Pierwszy ze zbiorów to zbiór danych w którym należy usunąć wartości brakujące, natomiast drugi to zbiór referencyjny na podstawie którego mają zostać usunięte wartości brakujące.

   Innymi słowy np. uzupełniając wartości brakujące poprzez wartość średnią szukamy wartości średniej na zbiorze dataRef a wstawiamy wyznaczoną wartość dla zbioru dataMiss.

2. Dla metody kNN konieczne będzie wyznaczenie wartości odległości pomiędzy punktami. W tym celu możesz wykorzystać funkcję distance2NaN, która zwraca odległość pomiędzy danym wektorem (zawierającym wartość brakującą) a całym zbiorem danych (pierwszy argument to zbiór danych a drugi to wektor wzorcowy)

   W następnej kolejności aby wyznaczyć najbliższych sąsiadów należy wyznaczyć odlegości od danego wektora, a następnie je posortować od najmniejszej do największej (funkcja sort). Uwaga funkcja sort zwraca dwa argumenty – wartości posortowane oraz indeks kolejności tzn nr wektora który był najbliżej itd. Korzystając z indeksów wartości posortowanych łatwo można wyznaczyć numery najbliższych sąsiadów jako k pierwszych wartości gdzie k to liczba sąsiadów używanych do wyznaczenia średniej. Zwykle k=3;

   Uważaj bo może się zdarzyć że wśród k najbliższych wektorów mogą się zdarzyć takie które też zawierają wartość NaN dla badanego przypadku. Takie przypadki należy usunąć spośród najbliższych sąsoadów

3. Dla danych treningowego jako zbiór referencyjny należy użyć zbiór treningowy, natomiast dla zbioru testowego jako zbiór referencyjny należy użyć zbioru treningowego.