

STUCZNA INTELIGENCJA, INTELIGENCJA OBLICZENIOWA I METODY INŻYNIERII WIEDZY

Klasyfikatory K Najbliższych Sąsiadów oraz ich wariacje



Adrian Horzyk
horzyk@agh.edu.pl



AGH

**Akademia Górniczo-
Hutnicza w Krakowie**



Klasyfikatory K Najbliższych Sąsiadów (KNN)

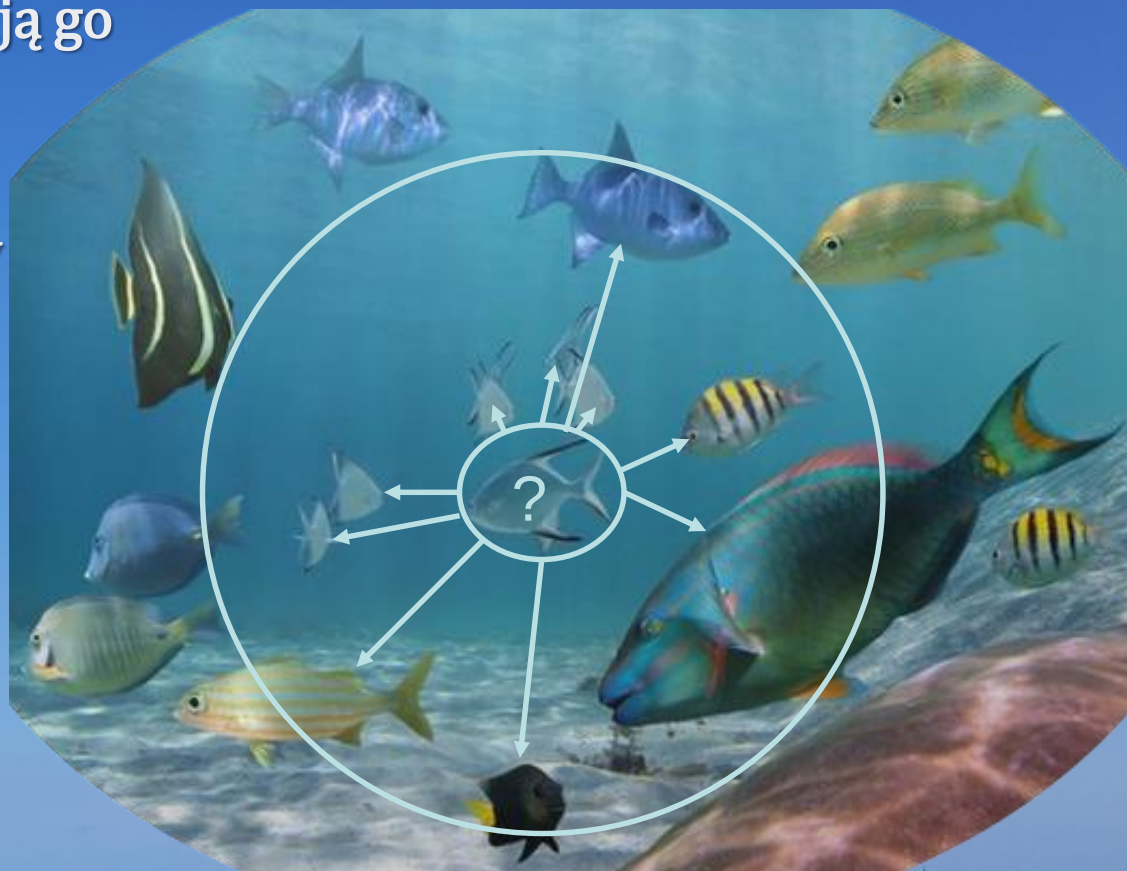


Klasyfikatory K-Najbliższych Sąsiadów (KNN - K-Nearest Neighbors) należą do grupy algorytmów **leniwych** (*lazy*), tj. takich, które nie tworzą wewnętrznej reprezentacji wiedzy, lecz poszukają rozwiązania po prezentacji każdego wzorca testowego przeszukując dane uczące. Metoda KNN wymaga przechowywania wszystkich wzorców uczących, dla których określana jest odległość do wzorca testowego.

Istnieją również inne algorytmy uczenia się, zwane **gorliwymi** (*eager*), które najpierw rozwijają model, a następnie wykorzystują go do określonego zadania obliczeniowego, np. klasyfikacji.

Ta grupa metod obejmuje wszystkie typy sieci neuronowych, systemy rozmyte, drzewa decyzyjne, SVM i wiele innych.

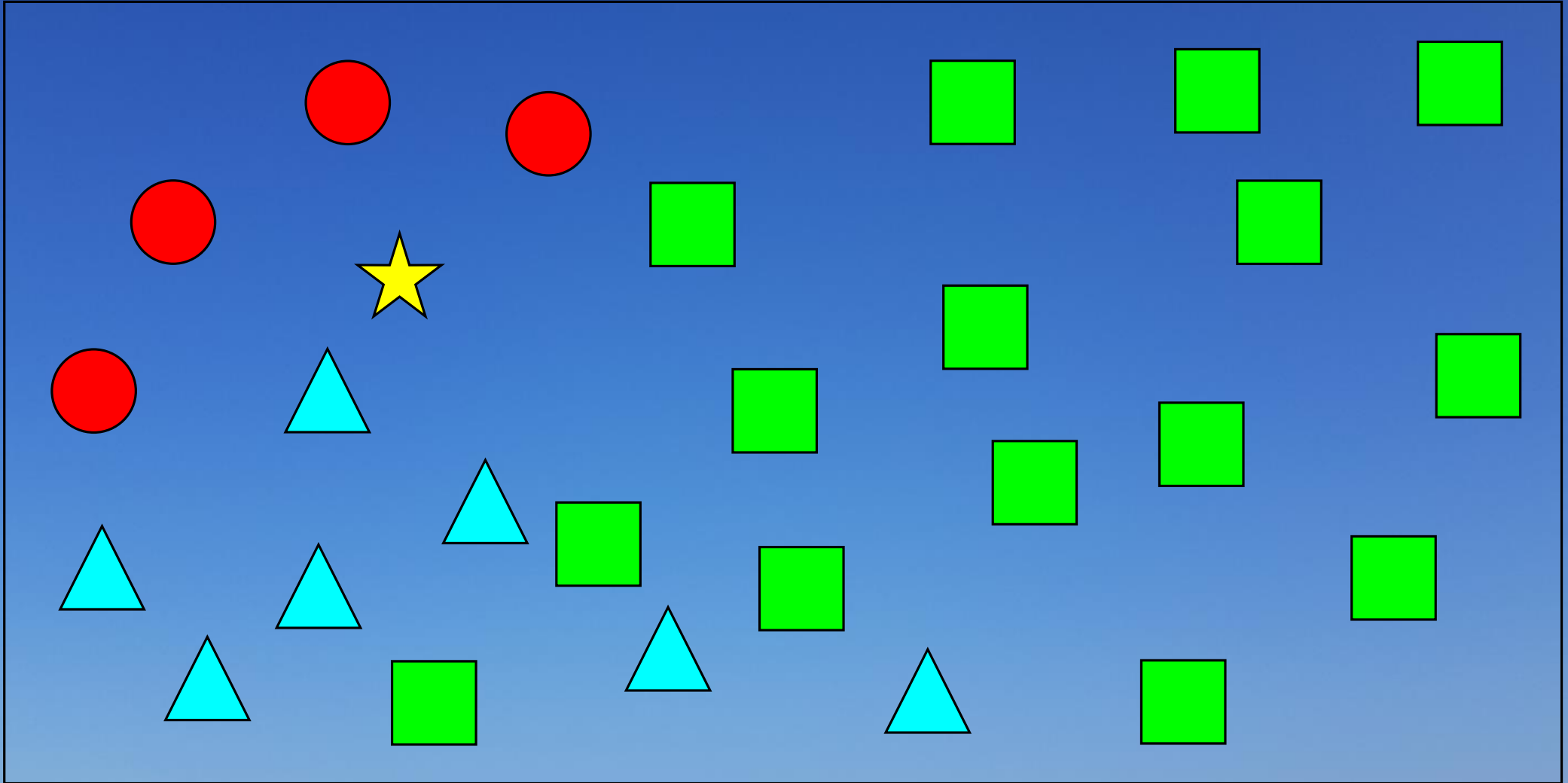
Po uczeniu (adaptacji) takich modeli dane uczące mogą zostać usunięte, ponieważ proces klasyfikacji wykorzystuje tylko utworzony model.



Klasyfikatory K Najbliższych Sąsiadów (KNN)



Do której klasy należy gwiazdka: **kółek**, **trójkątów** lub **kwadratów**?



Klasyfikatory K Najbliższych Sąsiadów (KNN)



Zestaw wzorców uczących (treningowych) składa się z zestawu par $\langle x^i, y^i \rangle$, gdzie:

$x^i = [x_1^i, \dots, x_n^i]$ jest wektorem wejściowym definiującym obiekty (zwykle w postaci wektora lub macierzy), y^i to przewidywana / pożądana wartość wyjściowa, np. indeks lub nazwa klasy, do której należy x^i .

Na rysunku mamy obiekty należące do **trzech klas**: **kółek**, **trójkątów** i **kwadratów**.

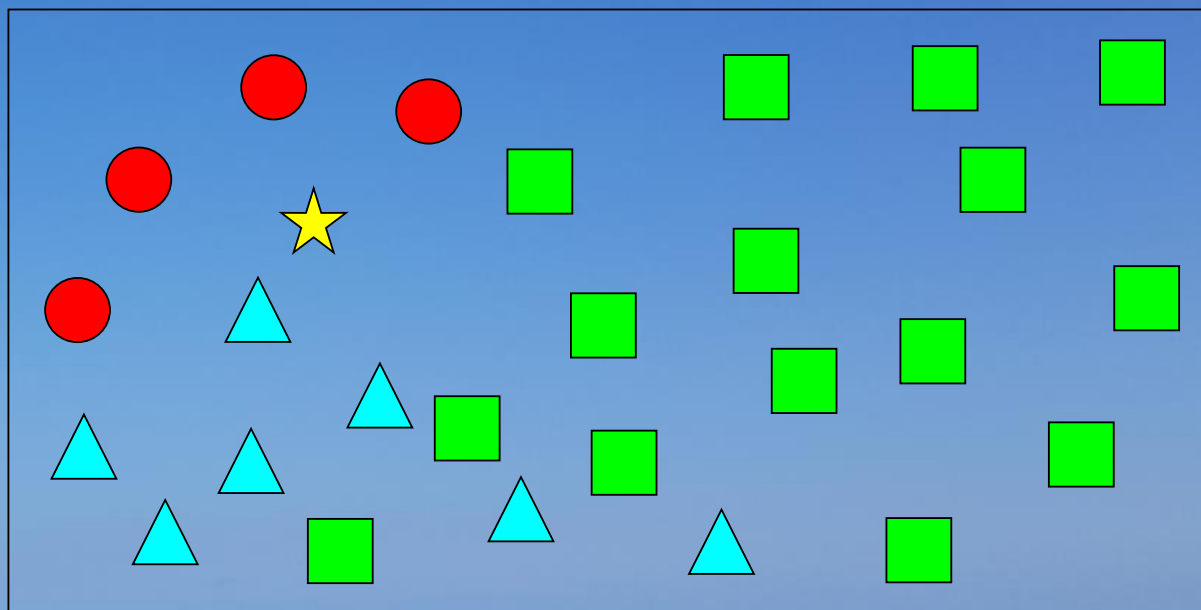
Gwiazdka to obiekt, który chcemy sklasyfikować, tj. przypisać go do jednej z istniejących klas.

Do której klasy należy **gwiazdka**: **kółek**, **trójkątów** czy **kwadratów**? Co nam podpowiada intuicja?

Na przykład, można zbadać odległość **gwiazdki** od innych obiektów, dla których klasa jest znana, używając jednej ze znanych metryk, np. odległości Euklidesa:

$$\|x - x^k\|_2 = \sqrt{\sum_{j=1}^J (x_j - x_j^k)^2}$$

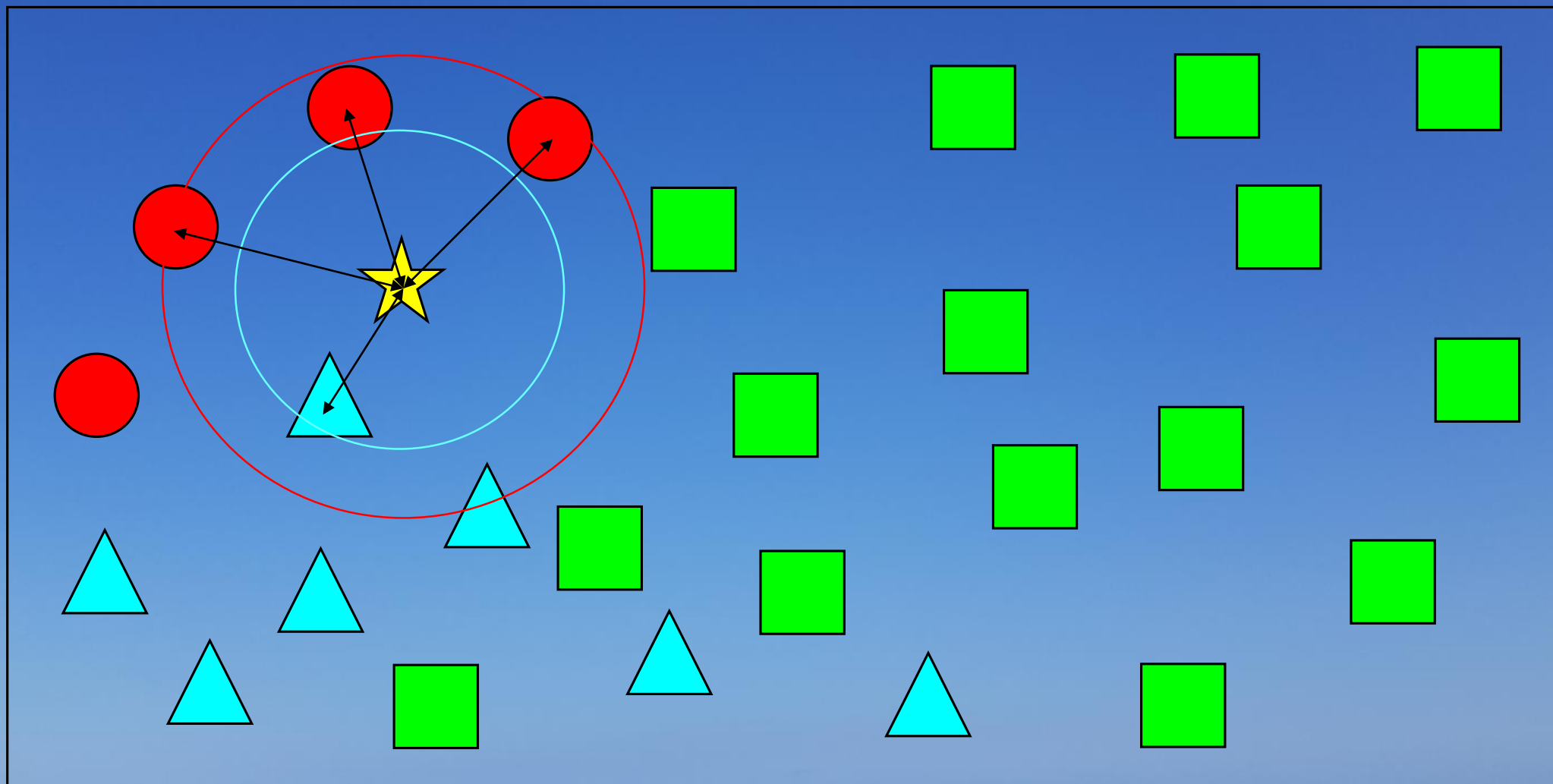
i na tej podstawie określ klasę **gwiazdki**.



Klasyfikatory K Najbliższych Sąsiadów (KNN)



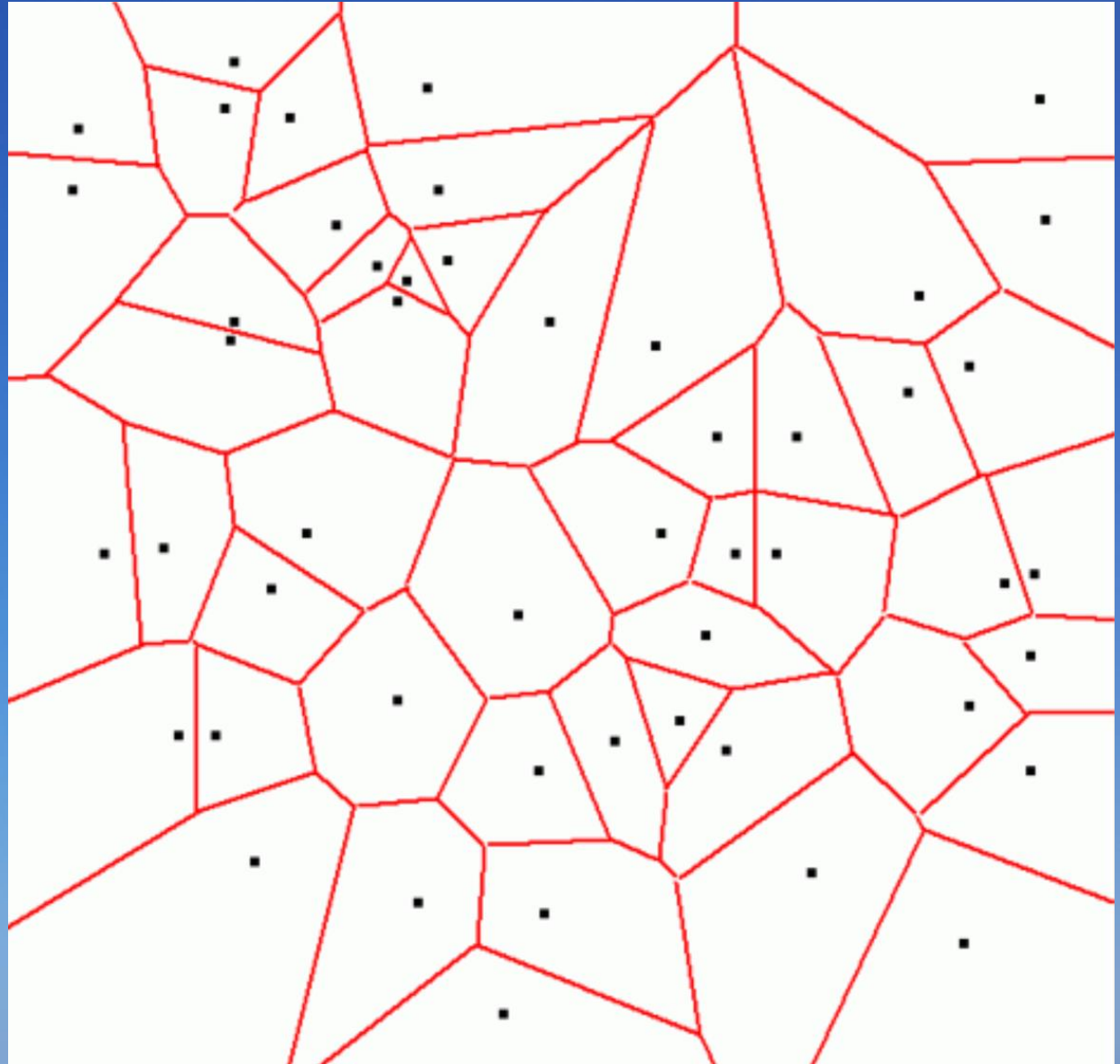
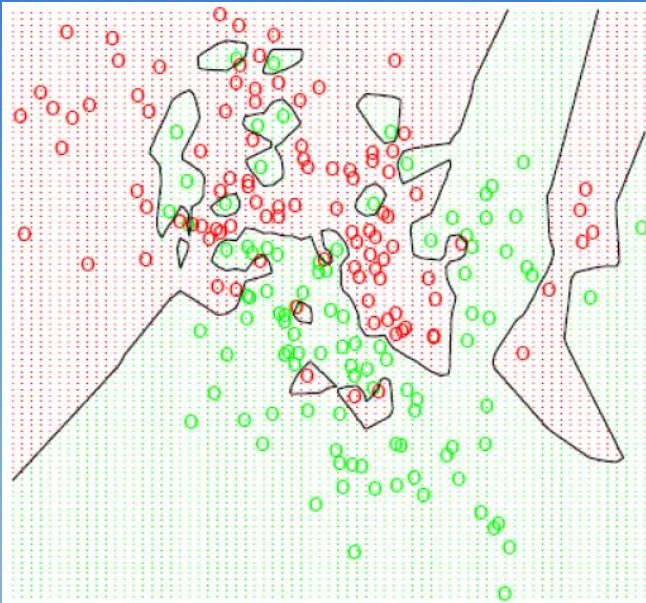
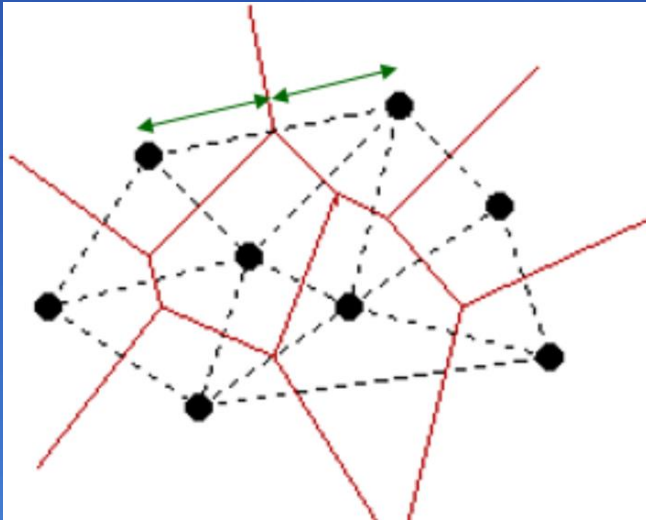
Klasyczny algorytm K Najbliższych Sąsiadów ma za zadanie znaleźć k sąsiadów, do których klasyfikowane obiekty są najbliższe dla wybranej metryki (np. Euklidesa), a następnie określa wynik klasyfikacji na podstawie większości głosów tych najbliższych k sąsiadów, biorąc pod uwagę, która klasa jest reprezentowana największą liczbą razy w grupie k najbliższych sąsiadów.



Regiony Przyciągania Klasyfikatorów KNN



Diagramy Voronoi są stosowane do ilustracji obszarów przyciągania do najbliższych obiektów w przestrzeni danych wejściowych:





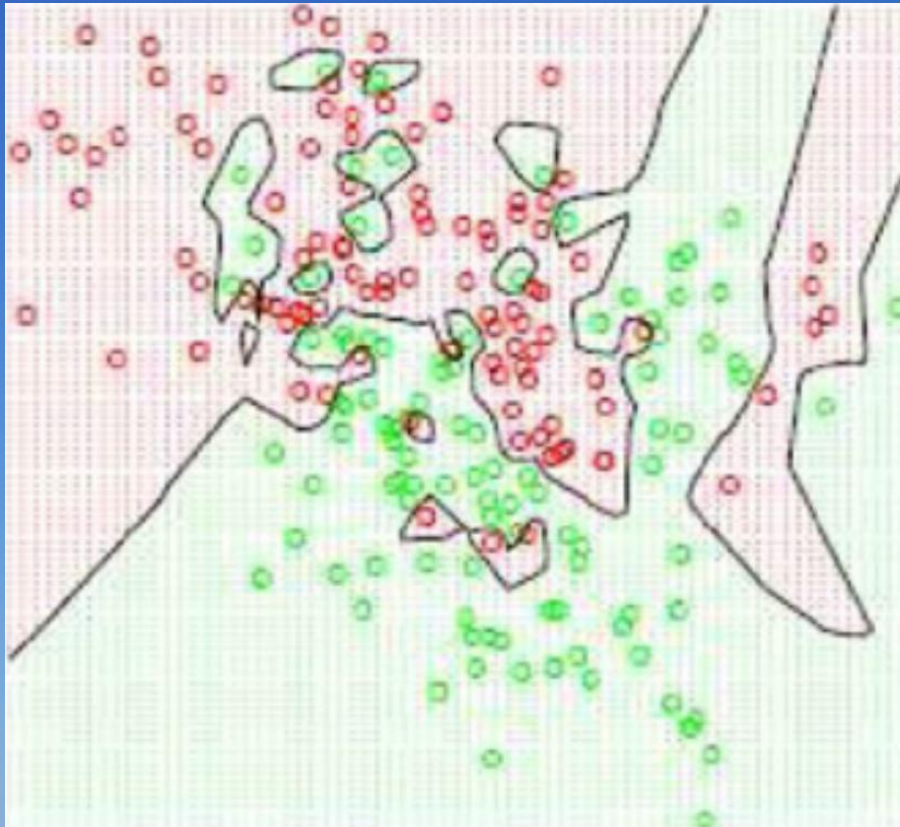
Regiony Przyciągania Klasyfikatorów KNN

Algorytm K Najbliższych Sąsiadów daje różne wyniki w zależności od wyboru parametru **K**.

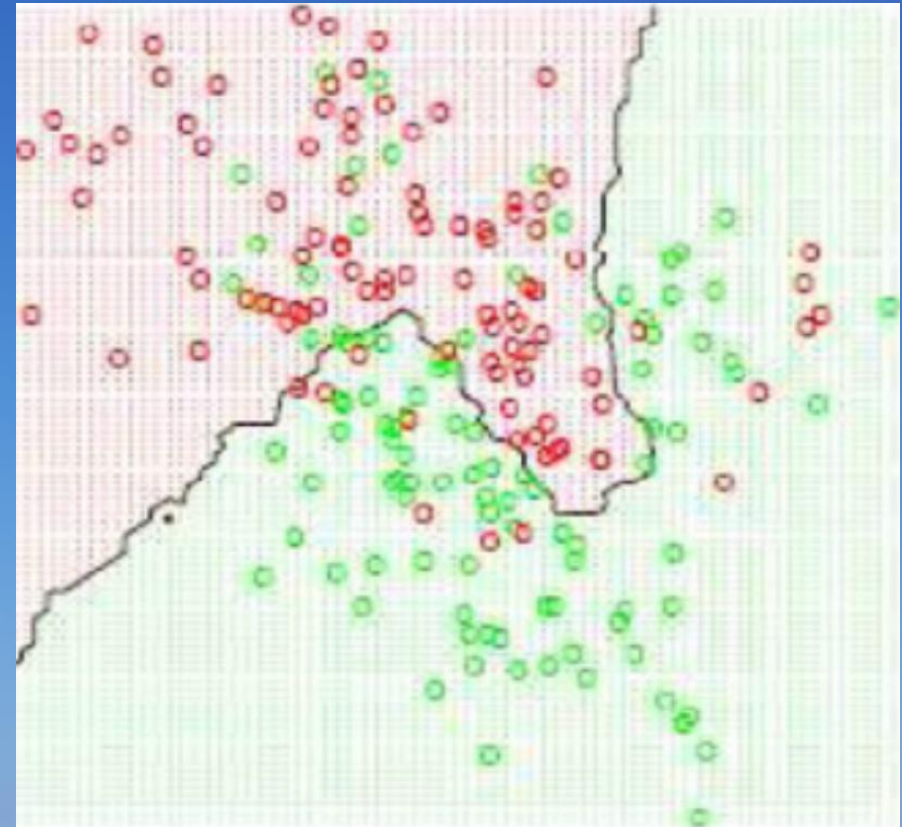
Obszary przyciągania determinują wyniki klasyfikacji.

Większe wartości **K** pozwalają na otrzymanie **bardziej wygładzonych obszarów przyciągania**, **usuwają szum i artefakty** (mogą zapewnić lepsze uogólnienie (generalizację)), ale również prowadzą do błędnej klasyfikacji odosobnionych poprawnych wzorców.

$K = 1$



$K = 15$



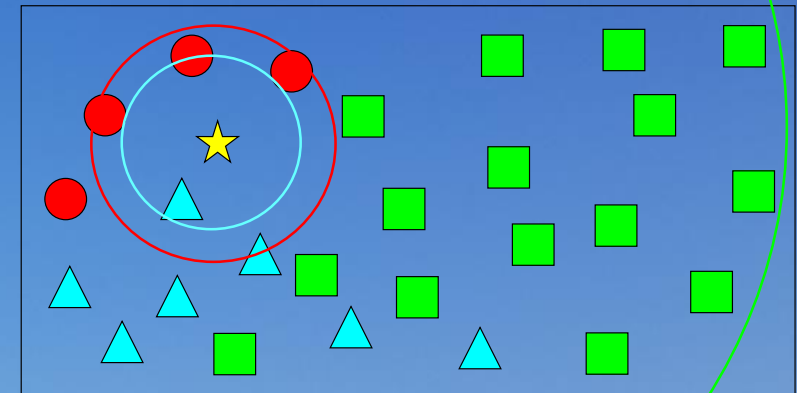
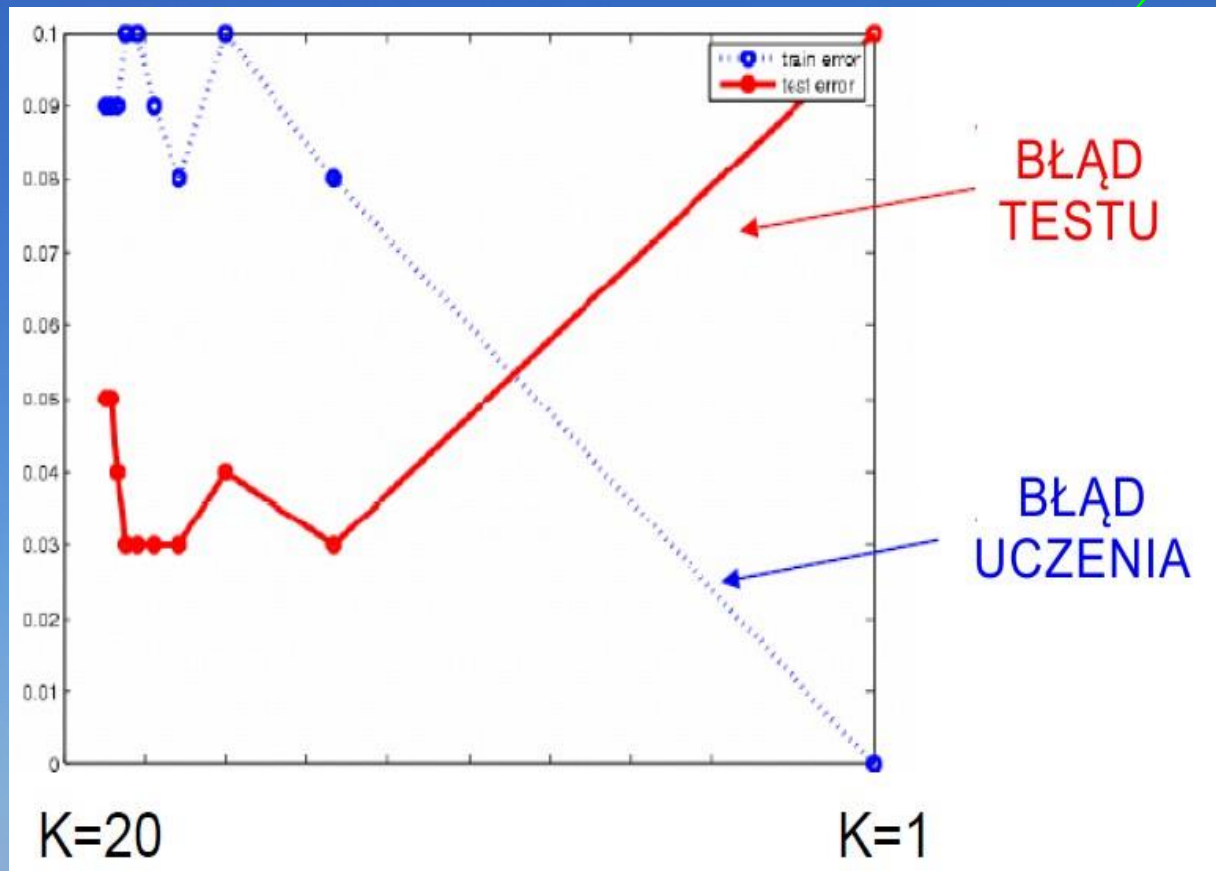
(źródło: Hastie, Tibshirani and Friedman (Elements of Statistical Learning))



Jak wybrać K?

Jeśli wybierzemy $K = N$, gdzie N jest ilością wszystkich wzorców (obiektów) uczących, wtedy wynik **klasyfikacji** będzie tylko **zależny od najbardziej licznie reprezentowanej klasy**:

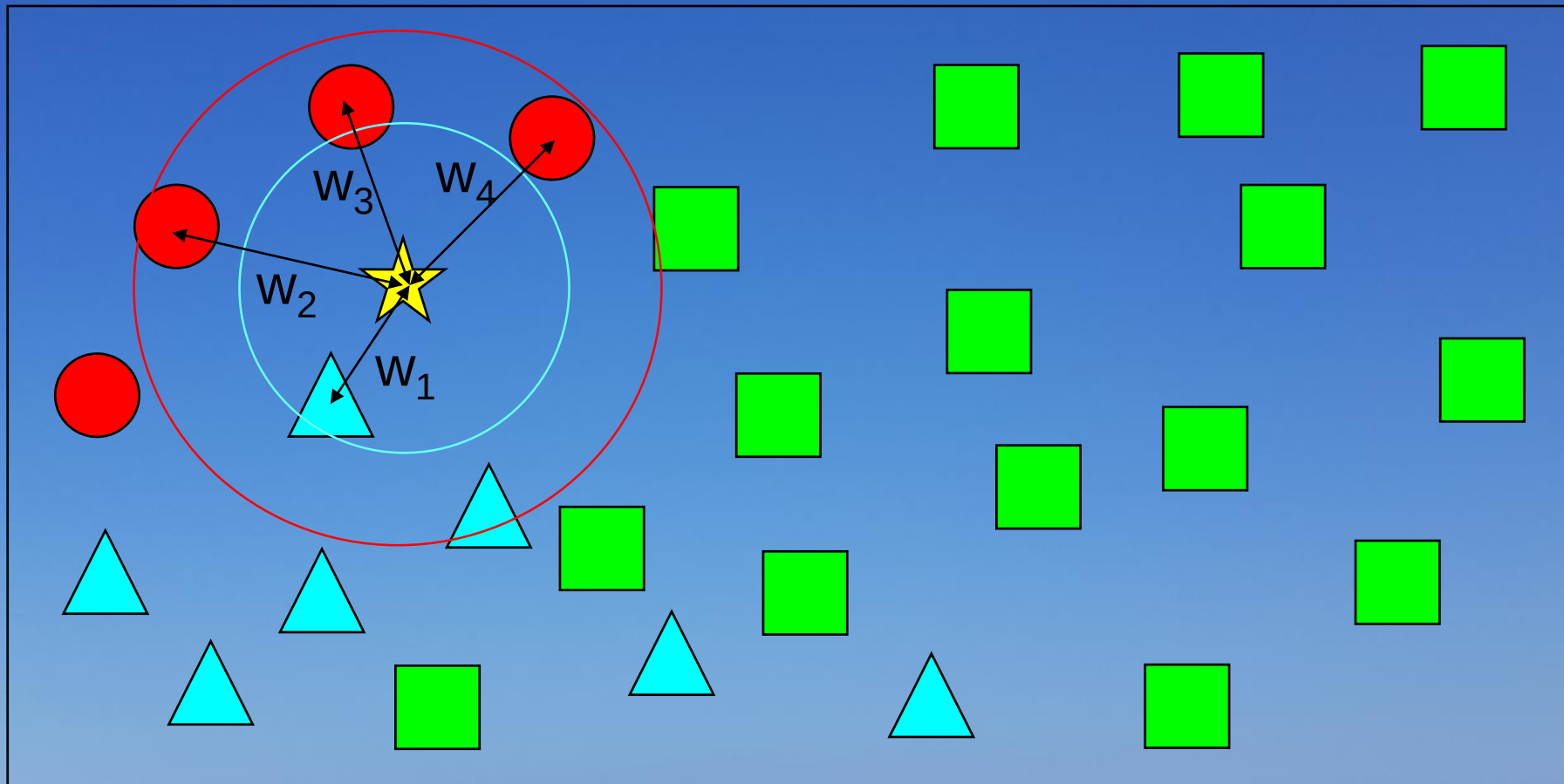
Błąd uczenia jest zawsze równy 0 dla $K = 1$,
nie oznacza to jednak zawsze najlepszego
uogólnienia (generalizacji).





Wariacje metody KNN

Metoda K Najbliższych Sąsiadów z ważeniem głosów w zależności od odległości sąsiadów bierze pod uwagę głosy dotyczące klasyfikowanego wzorca z wagą zależną od odległości każdego z najbliższych sąsiadów do tego wzorca. Odległość wyznaczamy zgodnie z przyjętą metryką. W związku z tym, najbliżsi sąsiedzi będą mieli mocniejszy głos (większy wpływ) na wynik klasyfikacji.



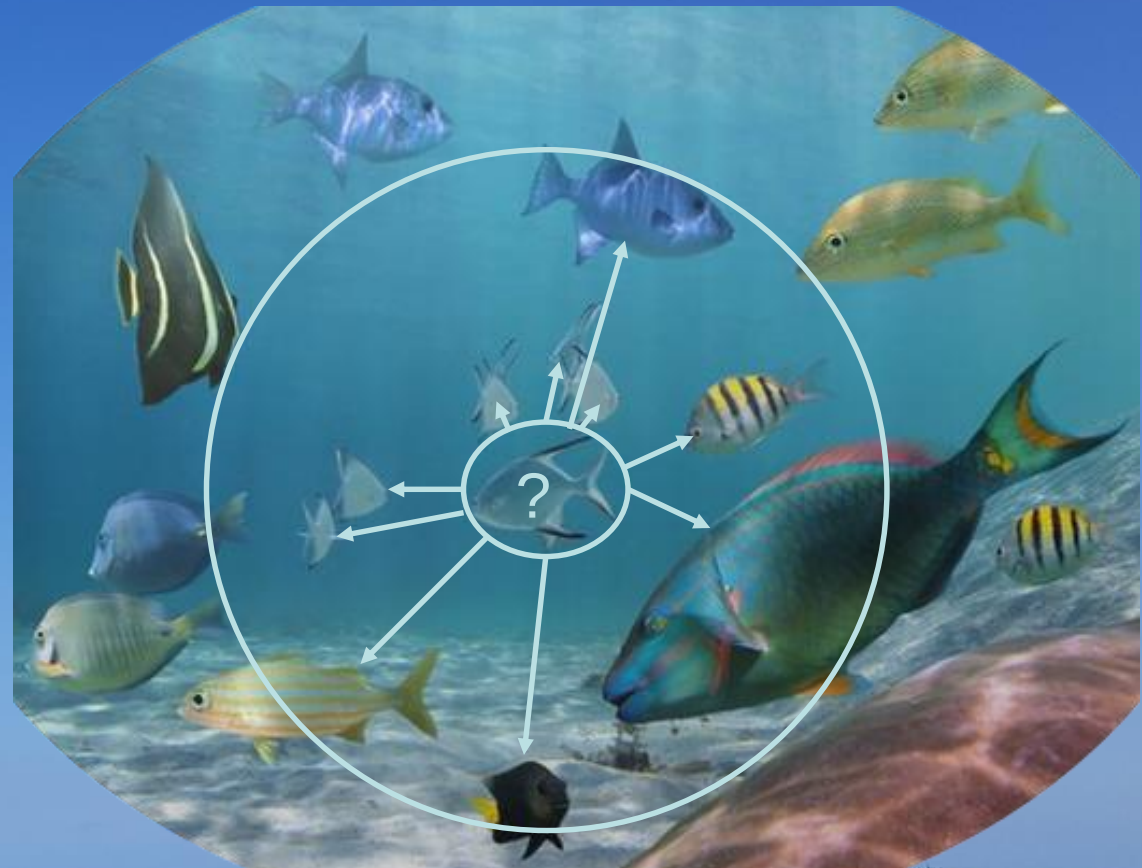


Wady klasyfikatorów KNN

Klasfikatory KNN są odporne na zaszumione dane treningowe i bardzo łatwe do wdrożenia, ale są:

- » **Leniwe** ponieważ nie tworzą modelu obliczeniowego,
- » **Wysoki koszt obliczeniowy** ponieważ wymagają obliczenia odległości do wszystkich wzorców uczących (których może być bardzo dużo), co daje liniowy koszt obliczeniowy w stosunku do wielkości zbioru danych uczących.
- » Metoda **jest wrażliwa na zmienność** reprezentacji wzorów różnych klas.

Dlatego KNN zwykle nie mogą być stosowane do problemów typu Big Data!



Po co przechowywać dane w tabelach?



W informatyce zwykle korzystamy z tabel do przechowywania, organizacji i zarządzania danymi:

SAMPLE OBJECTS	ATTRIBUTES				
	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLASS LABEL
O1	5.4	3.0	4.5	1.5	Versicolor
O2	6.3	3.3	4.7	1.6	Versicolor
O3	6.0	2.7	5.1	1.6	Versicolor
O4	6.7	3.0	5.0	1.7	Versicolor
O5	6.0	2.2	5.0	1.5	Virginica
O6	5.9	3.2	4.8	1.8	Versicolor
O7	6.0	3.0	4.8	1.8	Virginica
O8	5.7	2.5	5.0	2.0	Virginica
O9	6.5	3.2	5.1	2.0	Virginica

Jednak często wykorzystywane relacje, takie jak minima, maksima, tożsamość, podobieństwo, sąsiedztwo, liczba duplikatów muszą być wyszukiwane w kosztownych czasowo pętlach obliczeniowych, które muszą ewaluować różne warunki. Im więcej danych mamy, tym dłuższe wymagania czasowe!

Co można zrobić, aby osiągnąć lepszą wydajność?





Skojarz!

Big Data...

Duży
problem?



Możliwość wykorzystania struktur asocjacyjnych



Asocjacyjne Grafowe Struktury Danych (AGDS) można łatwo i szybko utworzyć dla dowolnych danych i umożliwić:

- » Zwiększenie wydajności obliczeniowej klasyfikacji KNN zazwyczaj dziesiątki lub setki razy w porównaniu z klasycznymi podejściami KNN.
- » Przekształcanie **leniwych** klasyfikatorów KNN na **gorliwe** klasyfikatory KNN + AGDS.
- » Definiowanie wydajnego modelu obliczeniowego dla KNN.
- » Agregowanie duplikatów wartości definiujących wzorce uczące i ich inteligentne definiowanie wartości atrybutów, oszczędzanie czasu i pamięci.
- » Unikanie przeglądania wszystkich danych treningowych podczas klasyfikacji.
- » Znajdowanie k najbliższych sąsiadów **zawsze w stałym czasie**, ponieważ sąsiedzi są przeszukiwani lokalnie tylko w najbliższym sąsiedztwie.
- » Uczynienie KNN odpowiednim i wydajnym do klasyfikacji Big Data!





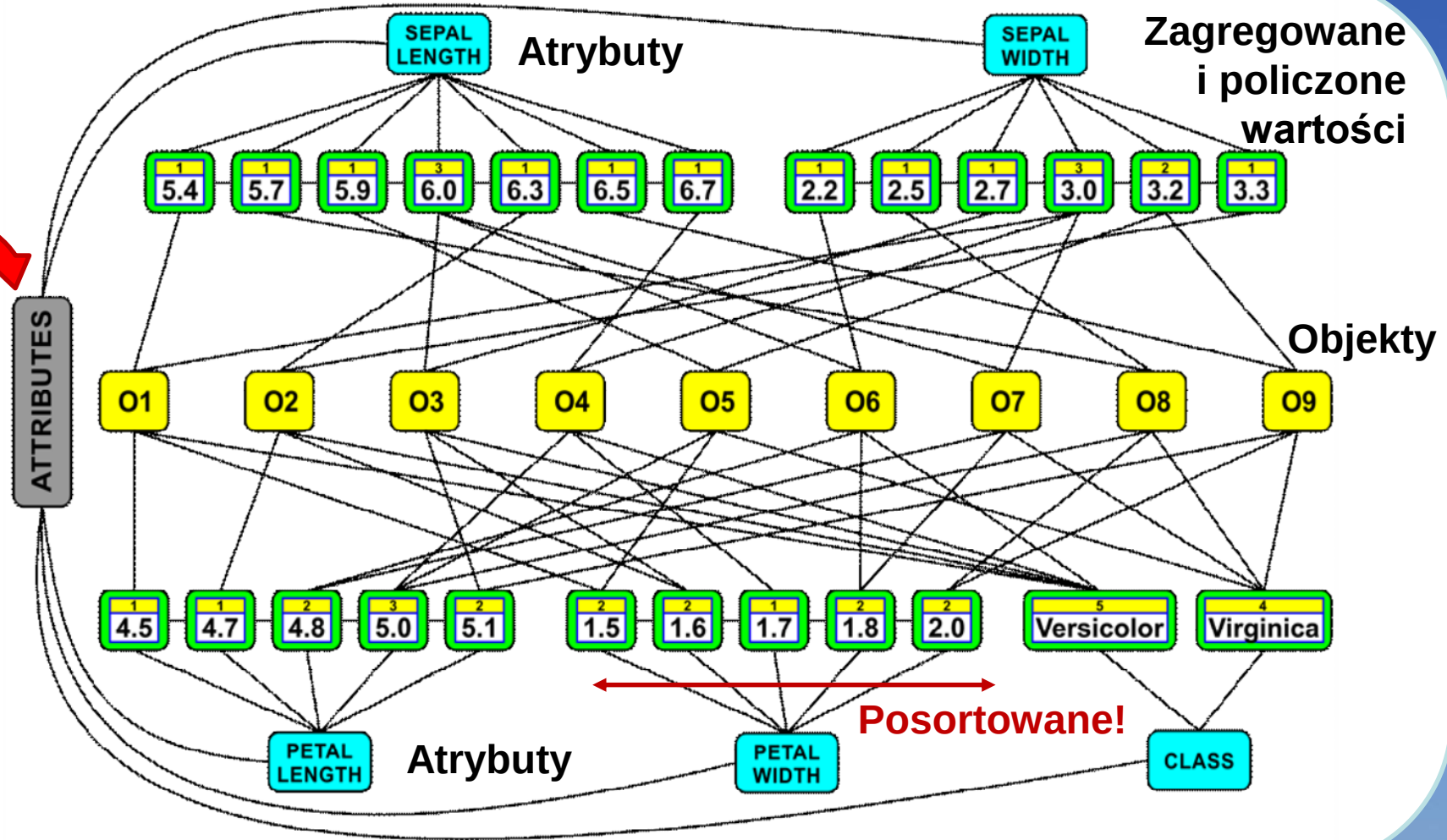
Asocjacyjne Graflowe Struktury Danych (AGDS)

AGDS łączą powiązane dane różnego rodzaju poziomo i pionowo:



SAMPLE OBJECTS	ATTRIBUTES				CLASS LABEL
	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	
O1	5.4	3.0	4.5	1.5	Versicolor
O2	6.3	3.3	4.7	1.6	Versicolor
O3	6.0	2.7	5.1	1.6	Versicolor
O4	6.7	3.0	5.0	1.7	Versicolor
O5	6.0	2.2	5.0	1.5	Virginica
O6	5.9	3.2	4.8	1.8	Versicolor
O7	6.0	3.0	4.8	1.8	Virginica
O8	5.7	2.5	5.0	2.0	Virginica
O9	6.5	3.2	5.1	2.0	Virginica

Inspirowane
mózgiem
AGDS



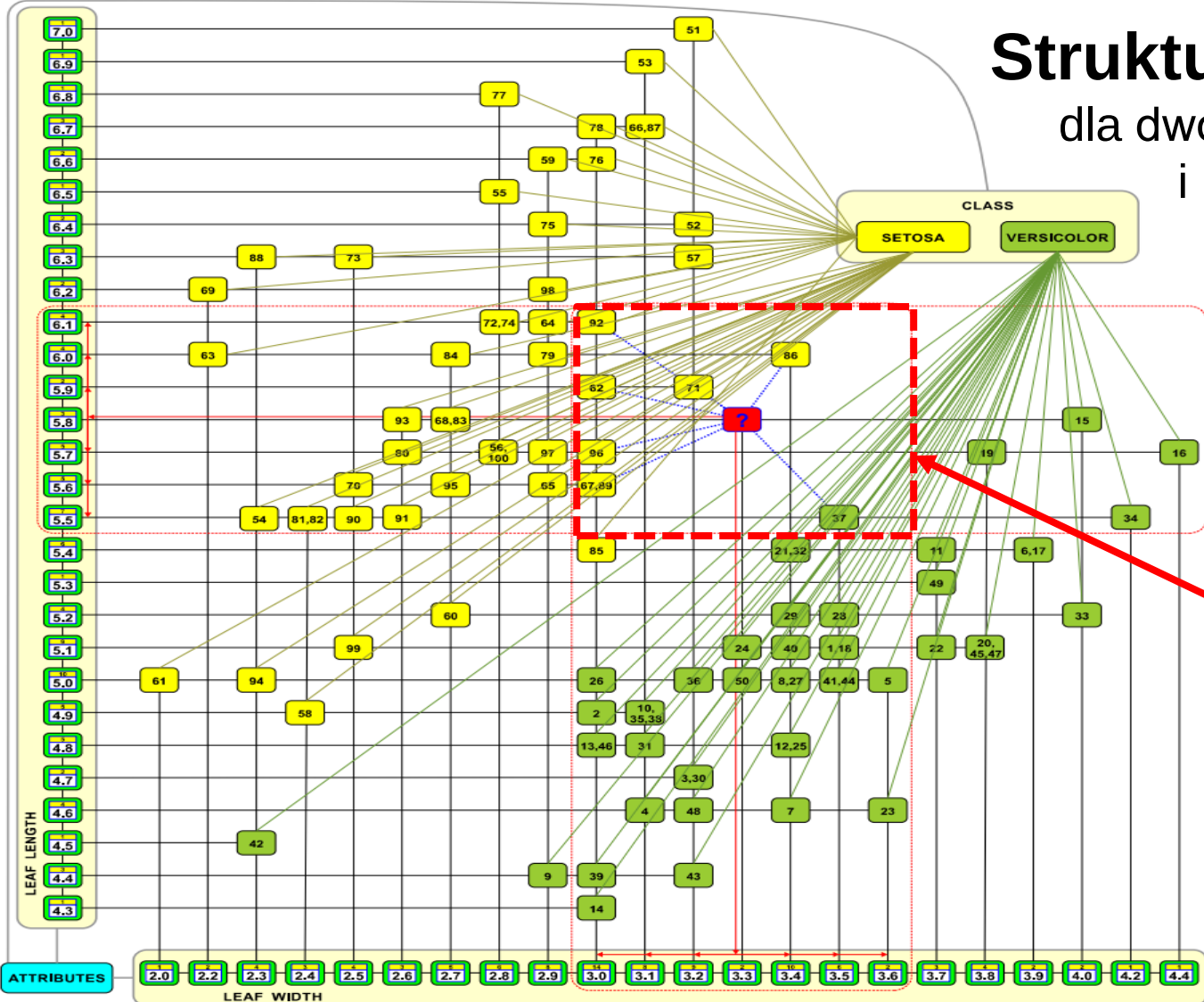
Połączenia reprezentują różne relacje między elementami AGDSów, takimi jak podobieństwo, bliskość, sąsiedztwo, porządek, następstwo, zawieranie itp.

KNNy wykorzystujące struktury AGDS



Wyszukiwanie jest ograniczone do niewielkiego regionu, w którym znajdują się sąsiedzi:

100 wartości reprezentowanych przez 28 węzłów wartości!



Struktura AGDS utworzona dla dwóch wybranych atrybutów i 100 próbek treningowych danych Iris

KNN wyszukuje najbliższych sąsiadów lokalnie w sąsiedztwie klasyfikowanej próbki

Możemy zaoszczędzić dużo czasu obliczeniowego dzięki stworzonym powiązaniom w AGDS!

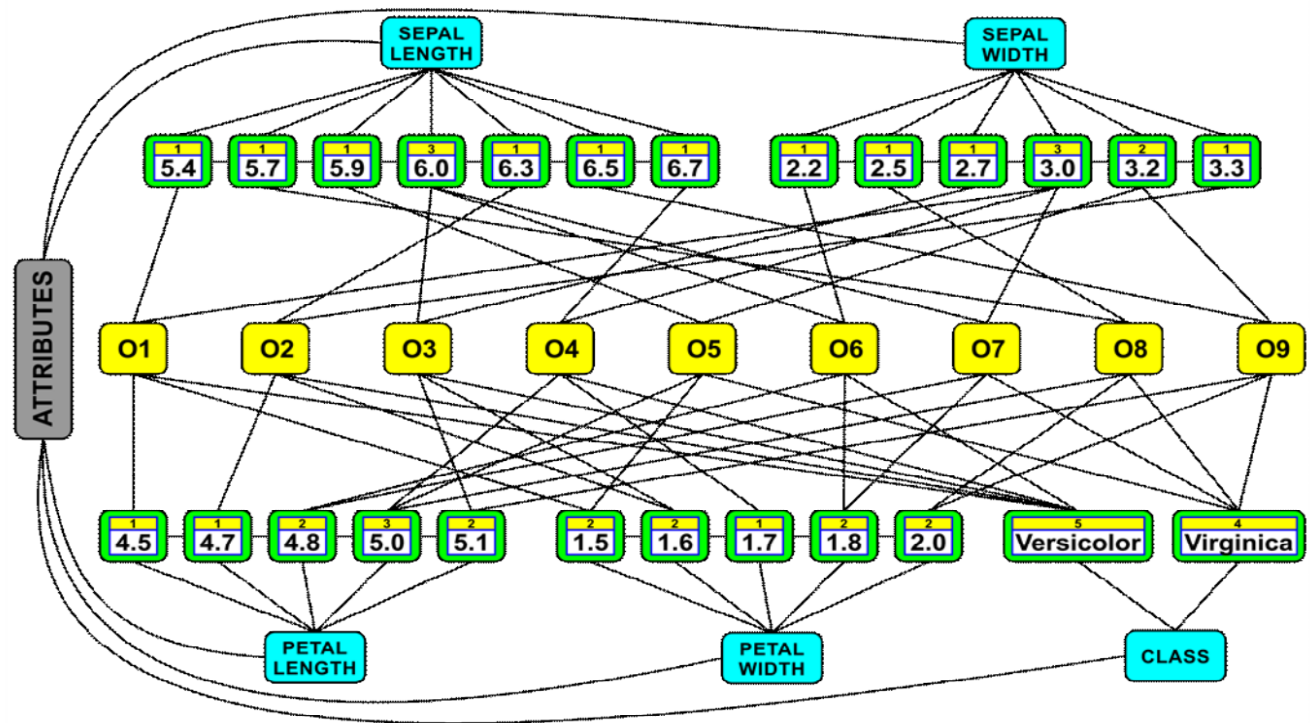
100 wartości reprezentowanych przez 22 węzły wartości!

Przyspieszenie klasyfikatora KNN dzięki zastosowaniu struktur AGDS



Rank table

1.		
2.		
3.		



1. Utwórz pustą k-wierszową tabelę rankingową, która będzie składała się ze wskaźników do k najbliższych sąsiadów i ich odległości do klasyfikowanego obiektu.

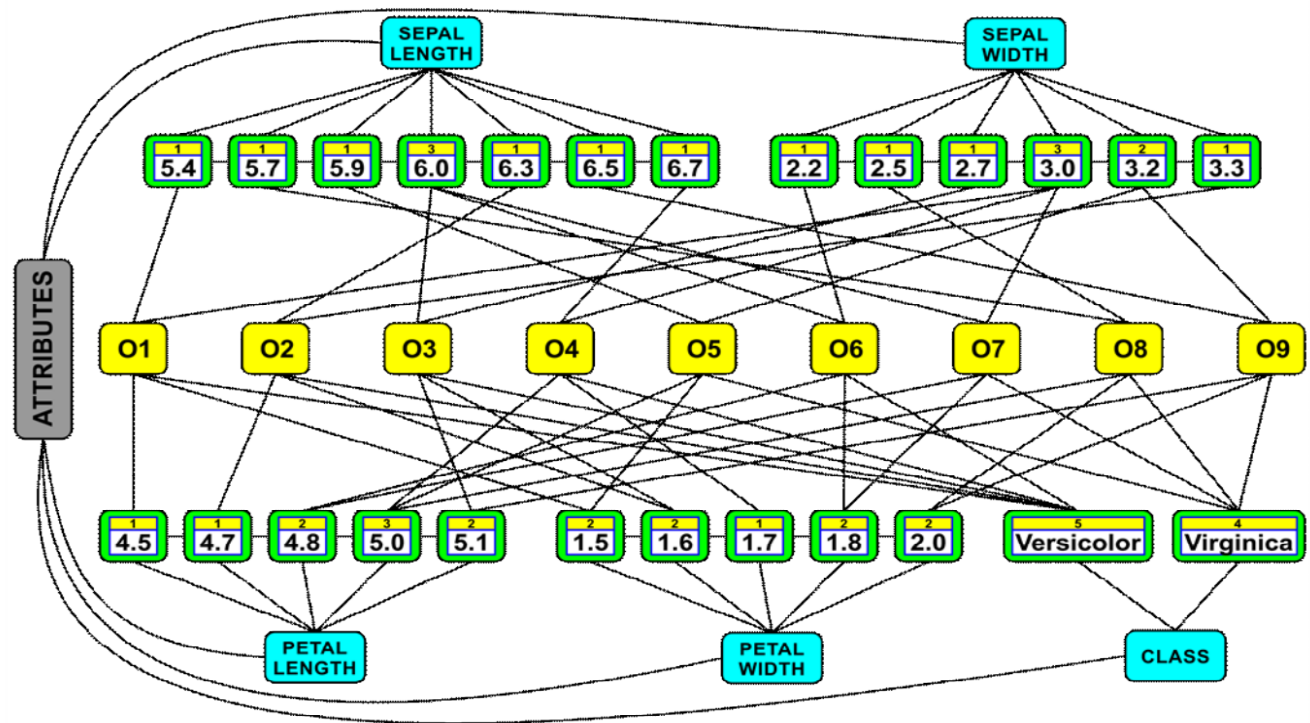
Przyspieszenie klasyfikatora KNN dzięki zastosowaniu struktur AGDS



Rank table

1.		
2.		
3.		

Classify [5.7; 2.5; 4.8; 1.6]



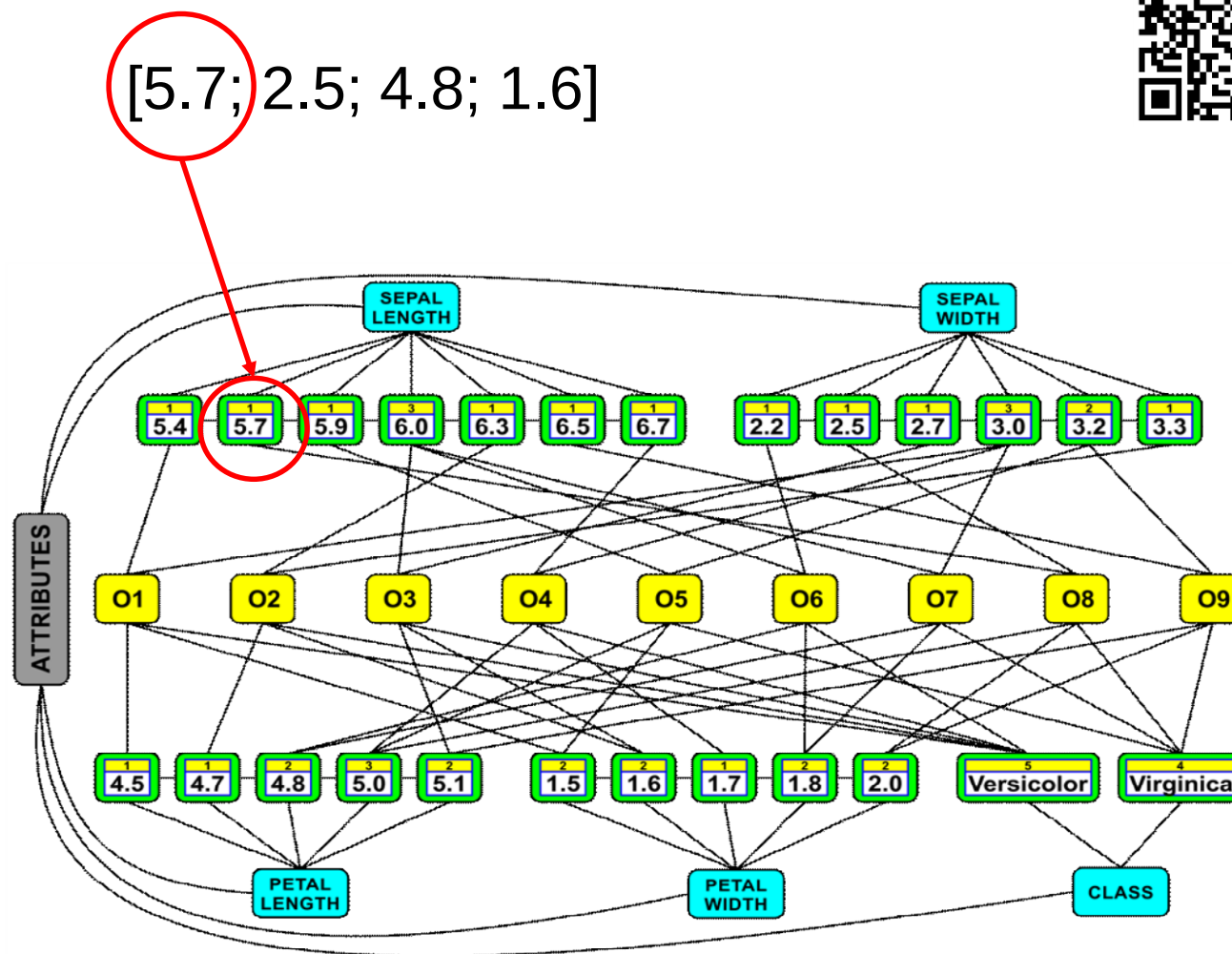
2. Dla pierwszej wartości atrybutu klasyfikowanego obiektu znajdź najbliższą wartość atrybutu w skonstruowanej strukturze AGDS.

Przyspieszenie klasyfikatora KNN dzięki zastosowaniu struktur AGDS



Rank table

1.		
2.		
3.		



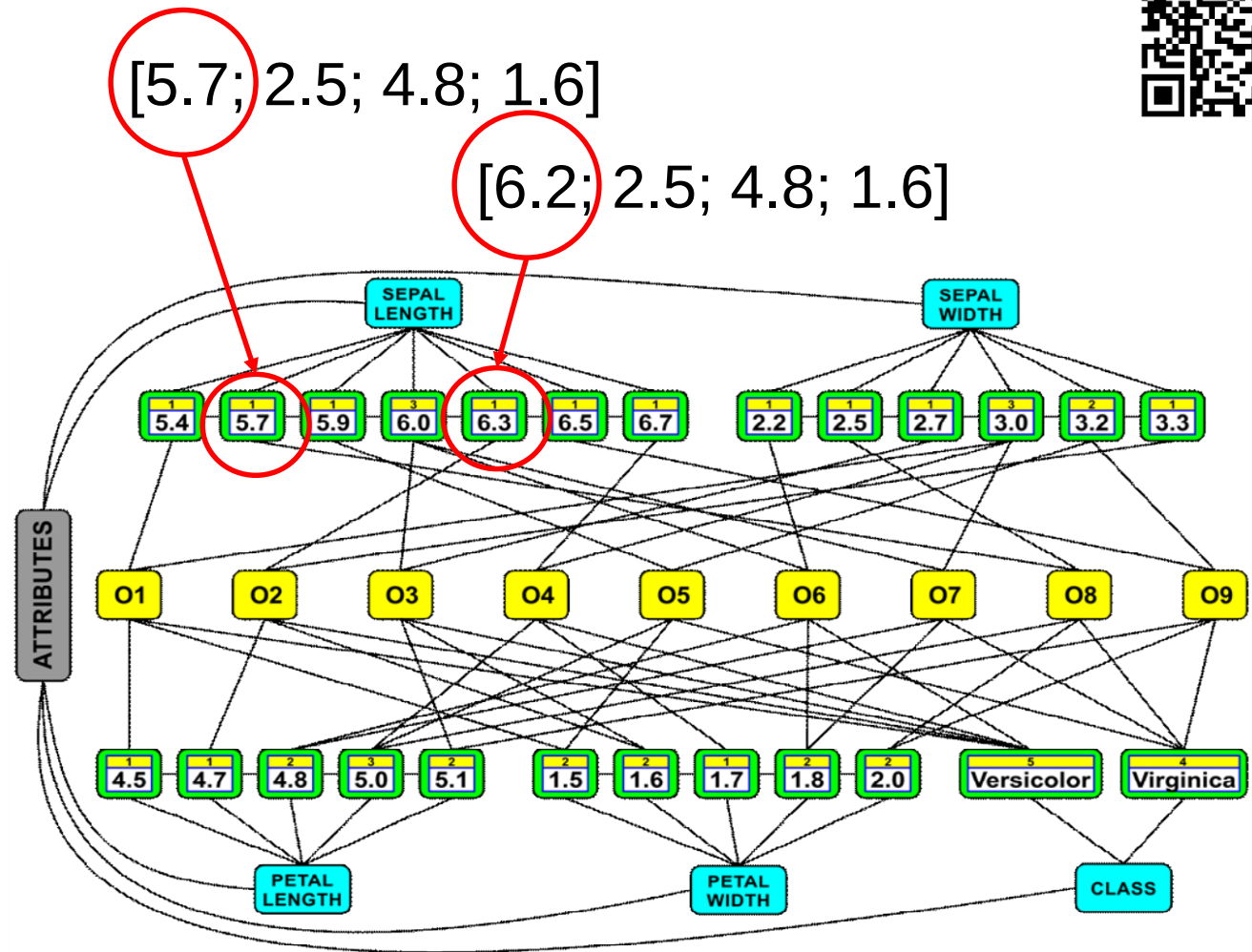
3. Gdy pierwsza wartość atrybutu sklasyfikowanego obiektu jest reprezentowana przez istniejący węzeł wartości struktury AGDS, przejdź do kroku 5, w przeciwnym razie przejdź do kroku 4.

Przyspieszenie klasyfikatora KNN dzięki zastosowaniu struktur AGDS



Rank table

1.		
2.		
3.		



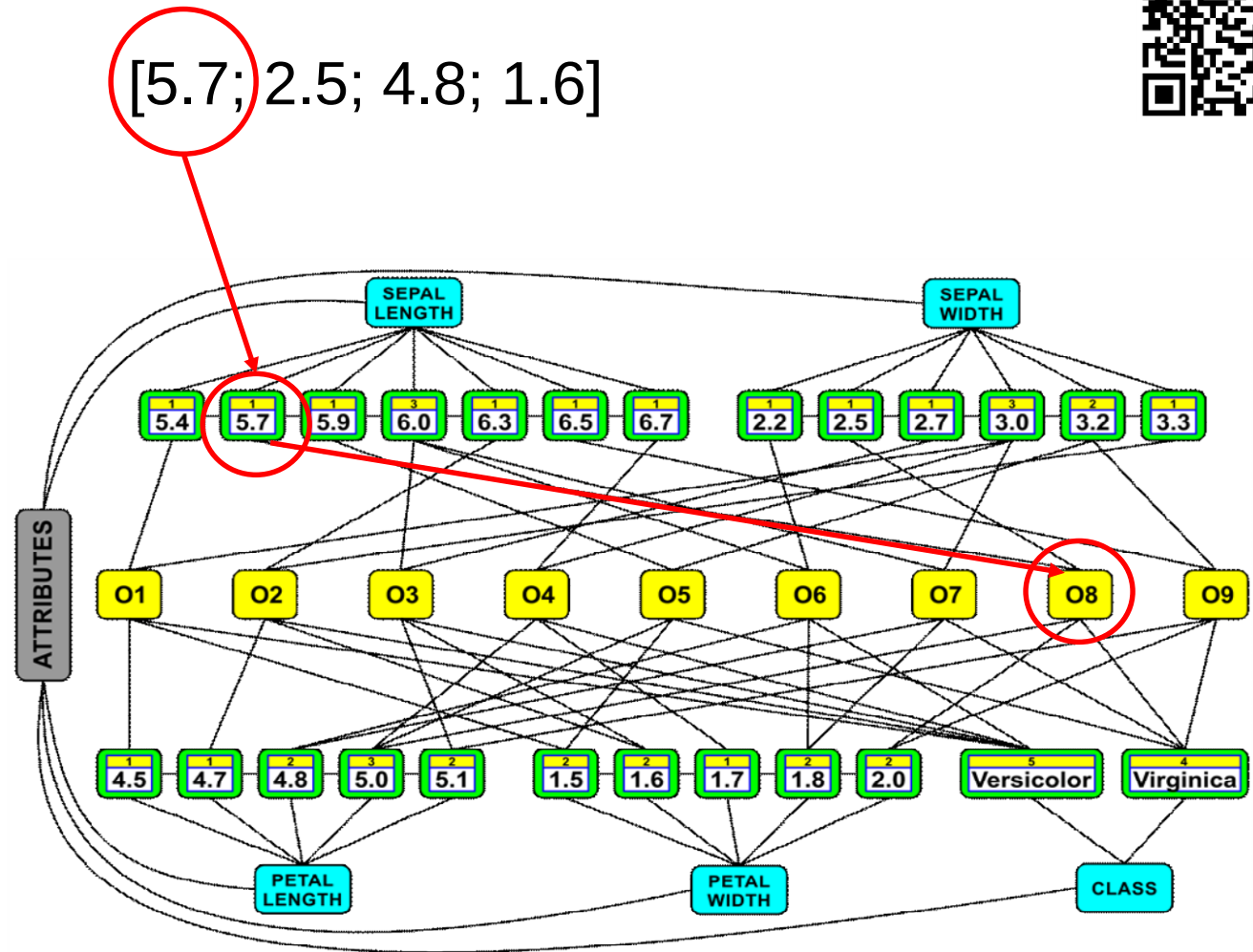
4. Gdy pierwsza wartość atrybutu sklasyfikowanego obiektu nie jest reprezentowana przez żaden węzeł wartości tego pierwszego atrybutu, to najbliższa wartość jest reprezentowana przez węzeł wartości reprezentujący najbliższą niższą lub najbliższą większą wartość lub oba. Wybierz najbliższą wartość lub jedną z najbliższych wartości i przejdź do kroku 5.

Przyspieszenie klasyfikatora KNN dzięki zastosowaniu struktur AGDS



Rank table

1.		
2.		
3.		



5. Przejdź wzdłuż wszystkich krawędzi wybranego węzła wartości do wszystkich połączonych węzłów obiektów i wykonaj krok 6 dla wszystkich tych węzłów obiektów.

Przyspieszenie klasyfikatora KNN dzięki zastosowaniu struktur AGDS



Rank table

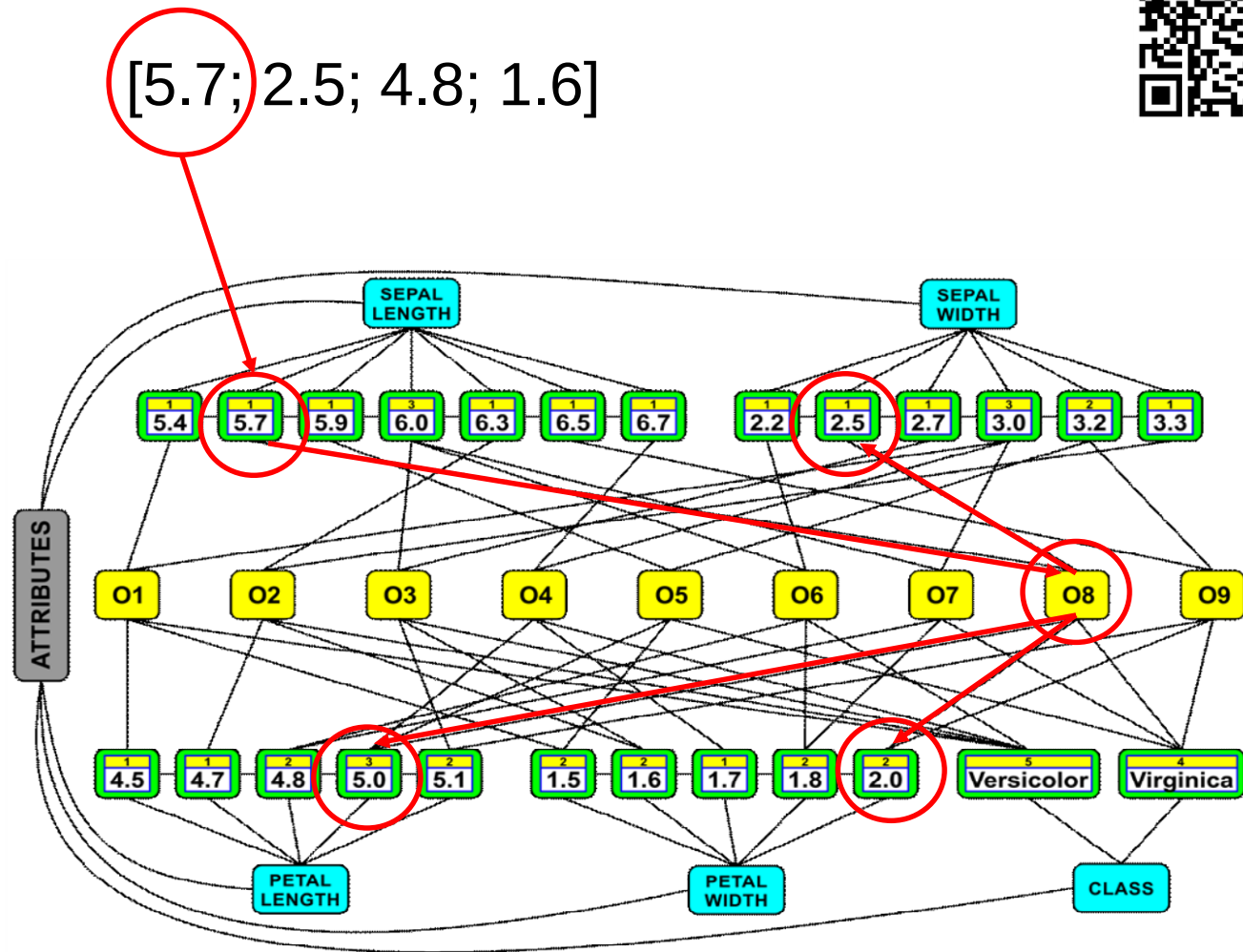
1.		
2.		
3.		

$$d_e(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

$$d_m(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (2)$$

$$d_e(x, y) = 0.45$$

$$d_m(x, y) = 0.60$$



6. W węźle osiągniętego obiektu przejdź do wszystkich połączonych węzłów wartości, z wyjątkiem węzła wartości, z którego został osiągnięty węzeł obiektu, i oblicz odległość zgodnie z (1) lub (2). Następnie spróbuj wstawić ten węzeł obiektu do tabeli rankingowej w kroku 7.

Przyspieszenie klasyfikatora KNN dzięki zastosowaniu struktur AGDS



Rank table

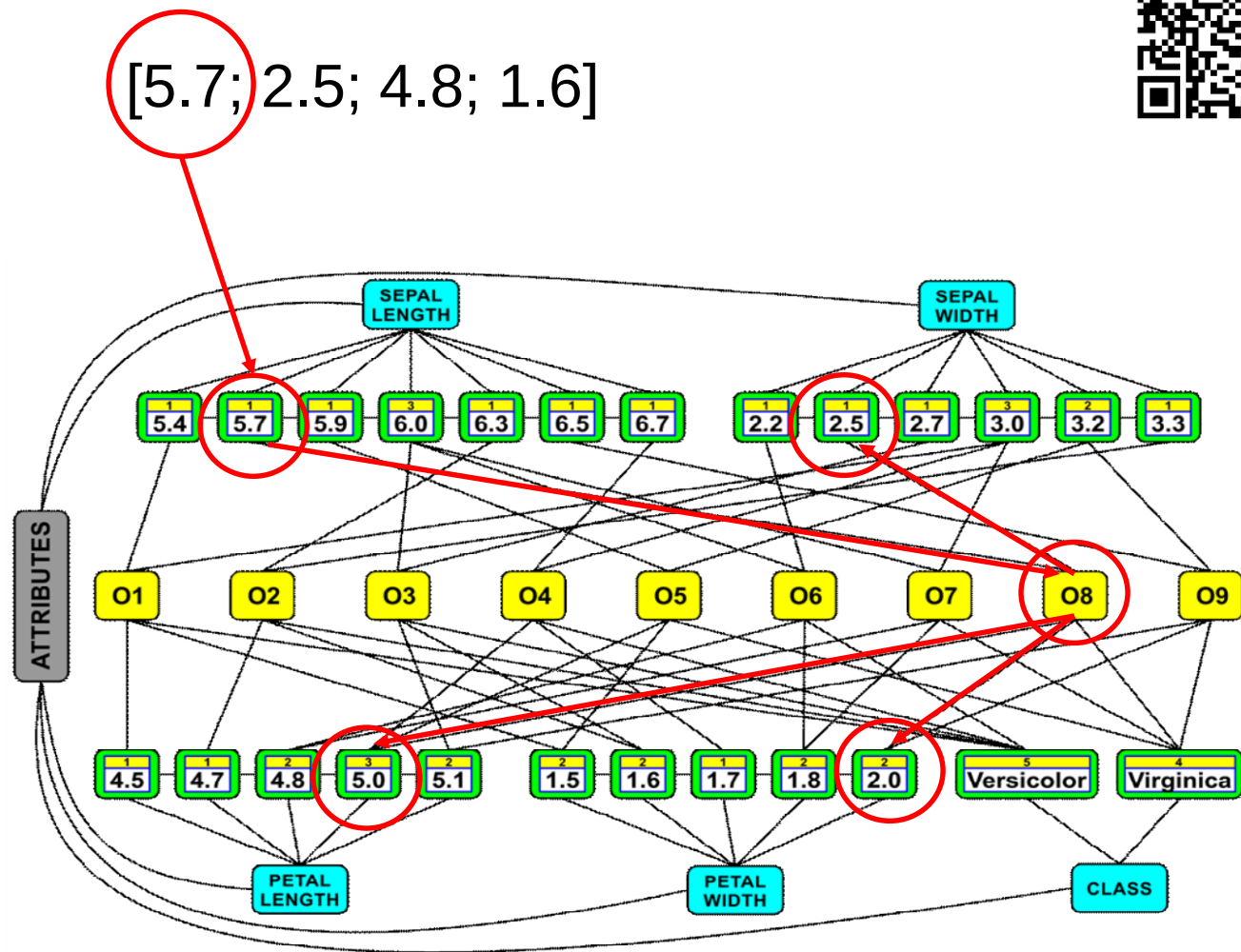
1.		
2.		
3.		

$$d_e(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

$$d_m(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (2)$$

$$d_e(x, y) = 0.45$$

$$d_m(x, y) = 0.60$$



7. Jeśli k-ty wiersz tabeli rankingowej jest pusty lub obliczona odległość jest krótsza niż odległość do węzła obiektu przechowywanego w ostatnim (k-tym) wierszu tabeli rankingowej, przejdź do kroku 8, w przeciwnym razie przejdź do krok 9.

Przyspieszenie klasyfikatora KNN dzięki zastosowaniu struktur AGDS



Rank table

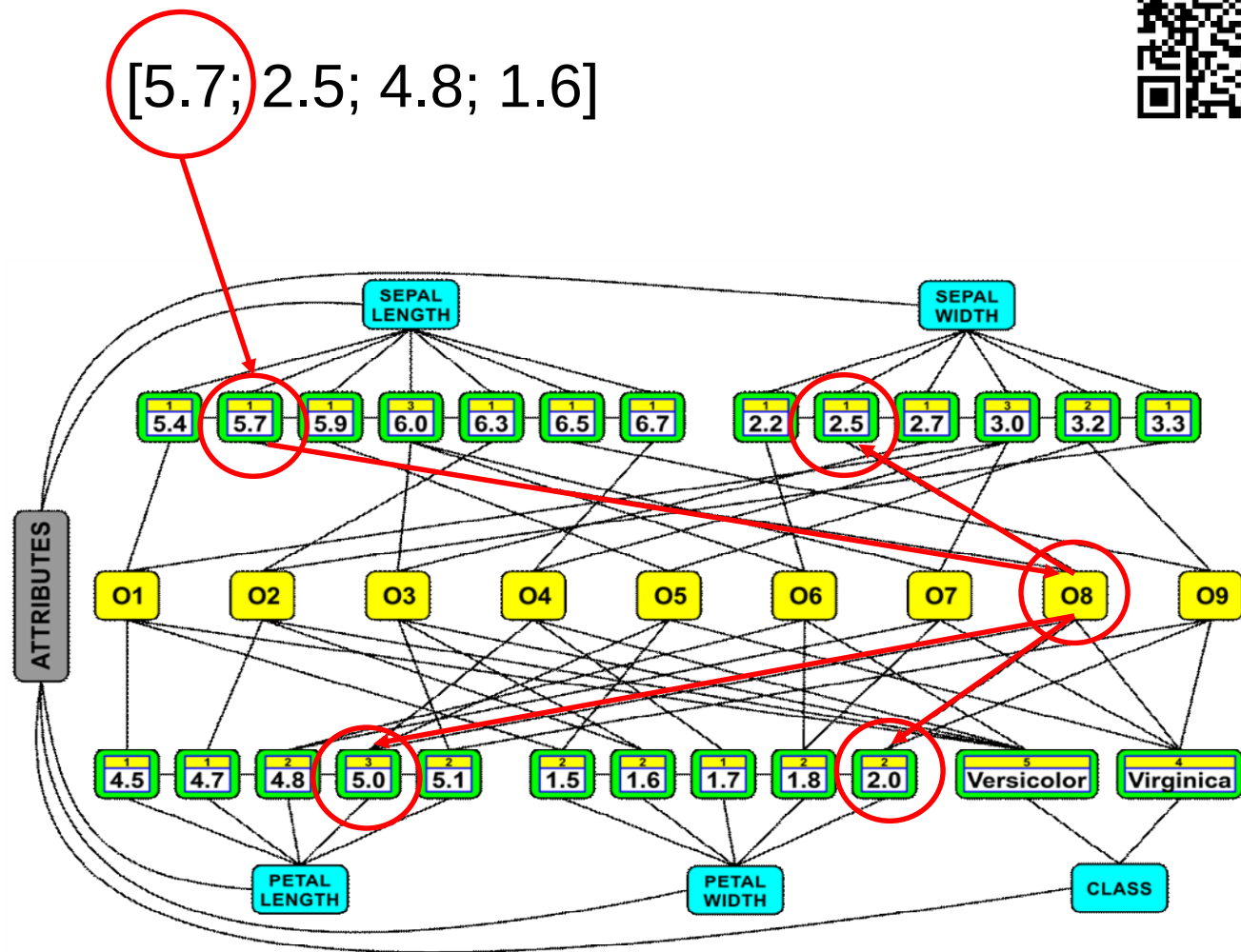
1.	O8	0.45
2.		
3.		

$$d_e(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

$$d_m(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (2)$$

$$d_e(x, y) = 0.45$$

$$d_m(x, y) = 0.60$$



8. Wstaw ten węzeł i jego odległość do tabeli rankingowej w kolejności rosnącej (przy użyciu algorytmu sortowania wstawiania połówkowego), a jeśli to konieczne (jeśli tabela jest przepełniona), usuń ostatni (tj. najbardziej odległy) węzeł obiektu wraz z jego odległością od tej tabeli.

Przyspieszenie klasyfikatora KNN dzięki zastosowaniu struktur AGDS

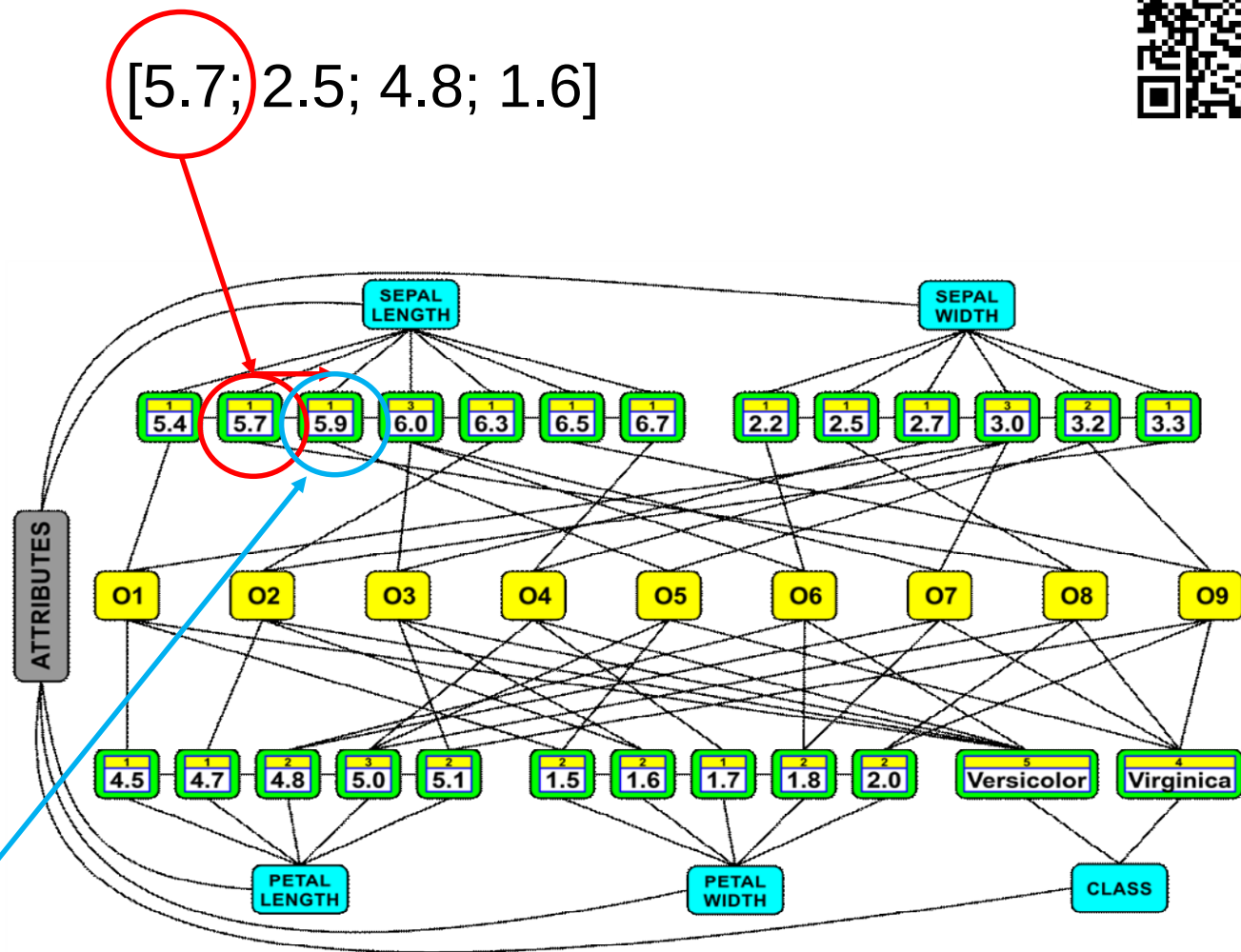


Rank table

1.	O8	0.45
2.		
3.		

$$d_e(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

$$d_m(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (2)$$



9. Po sprawdzeniu wszystkich węzłów obiektu połączonych z aktualnie wybranym węzłem wartości przedstawionym jako najbliższy pierwszej wartości atrybutu sklasyfikowanego obiektu z już nie przetworzonych węzłów wartości, **przejdź do następnego węzła najbliższej wartości** (reprezentującego niższą lub większą wartość do pierwsza wartość atrybutu sklasyfikowanego obiektu).

Przyspieszenie klasyfikatora KNN dzięki zastosowaniu struktur AGDS



Rank table

1.	O8	0.45
2.		
3.		

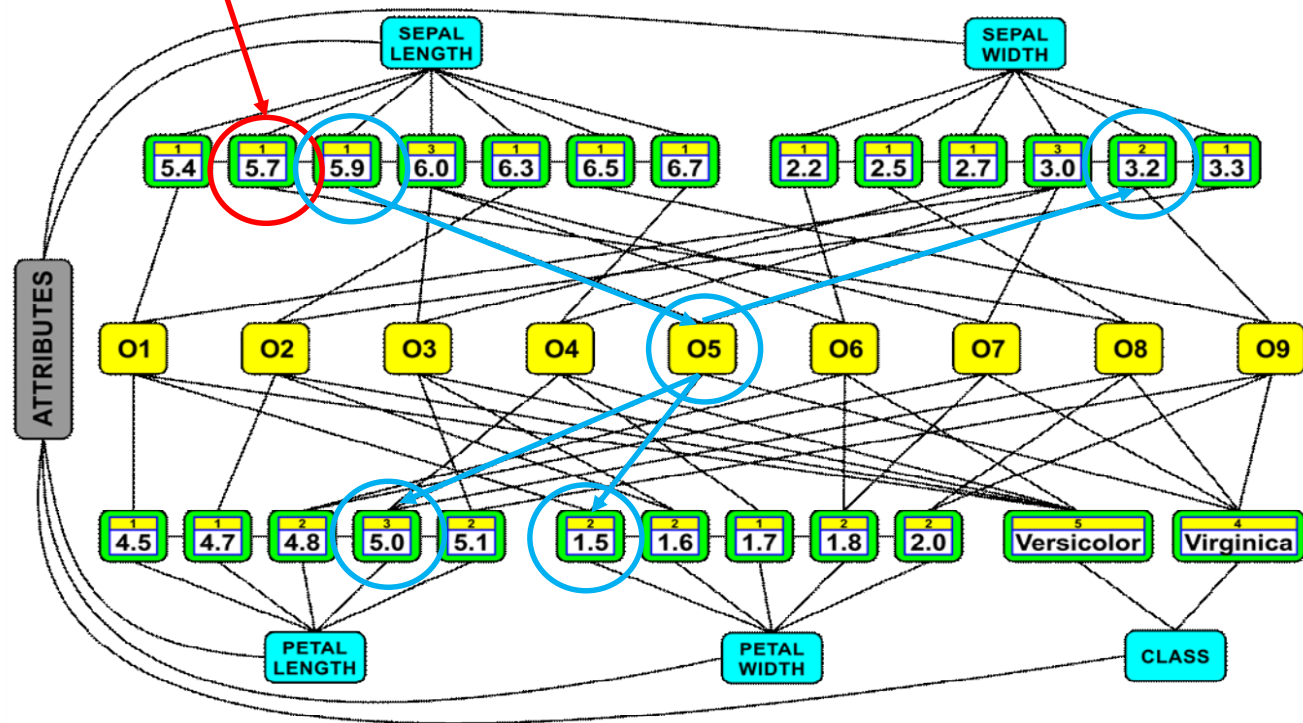
[5.7; 2.5; 4.8; 1.6]

$$d_e(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

$$d_m(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (2)$$

$$d_e(x, y) = 0.48$$

$$d_m(x, y) = 0.90$$



9. Po sprawdzeniu wszystkich węzłów obiektu połączonych z aktualnie wybranym węzłem wartości przedstawionym jako najbliższy pierwszej wartości atrybutu sklasyfikowanego obiektu z już nie przetworzonych węzłów wartości, **przejdź do następnego węzła najbliższej wartości** (reprezentującego niższą lub większą wartość do pierwsza wartość atrybutu sklasyfikowanego obiektu).

Przyspieszenie klasyfikatora KNN dzięki zastosowaniu struktur AGDS



Rank table

1.	O8	0.45
2.	O5	0.48
3.		

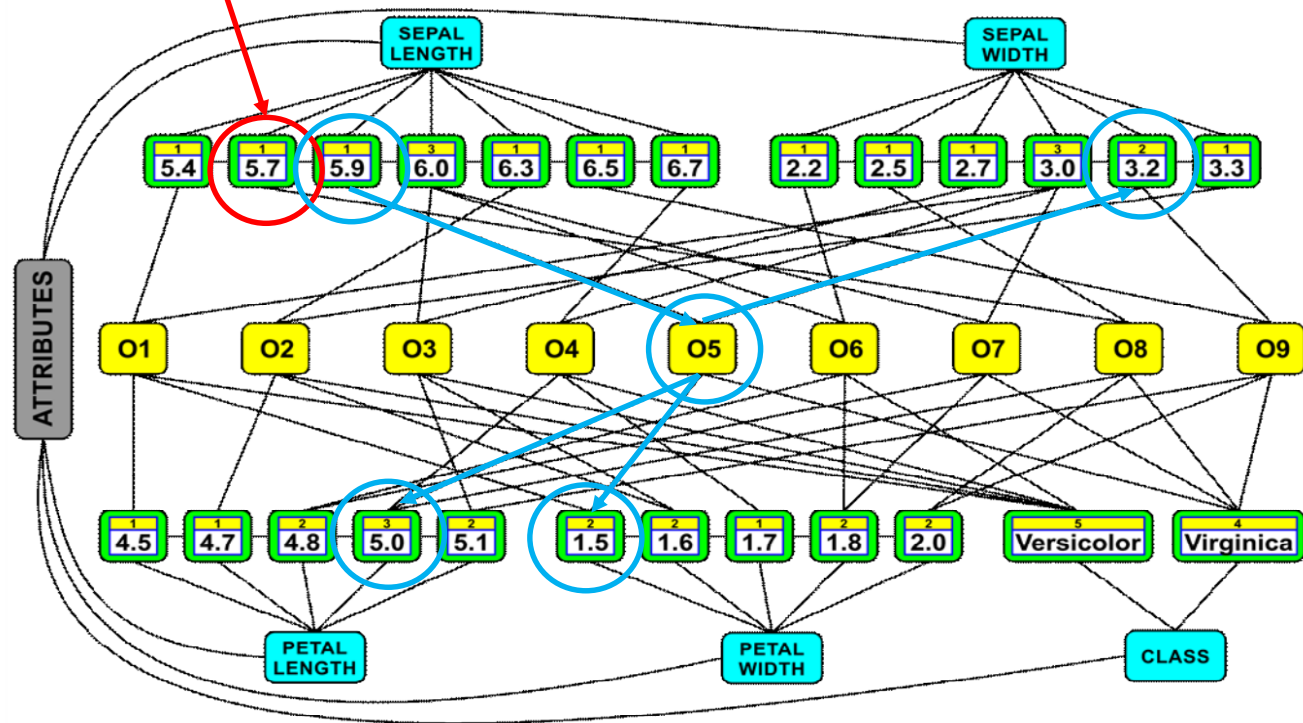
[5.7; 2.5; 4.8; 1.6]

$$d_e(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

$$d_m(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (2)$$

$$d_e(x, y) = 0.48$$

$$d_m(x, y) = 0.90$$



9. Po sprawdzeniu wszystkich węzłów obiektu połączonych z aktualnie wybranym węzłem wartości przedstawionym jako najbliższy pierwszej wartości atrybutu sklasyfikowanego obiektu z już nie przetworzonych węzłów wartości, **przejdź do następnego węzła najbliższej wartości** (reprezentującego niższą lub większą wartość do pierwsza wartość atrybutu sklasyfikowanego obiektu).

Przyspieszenie klasyfikatora KNN dzięki zastosowaniu struktur AGDS

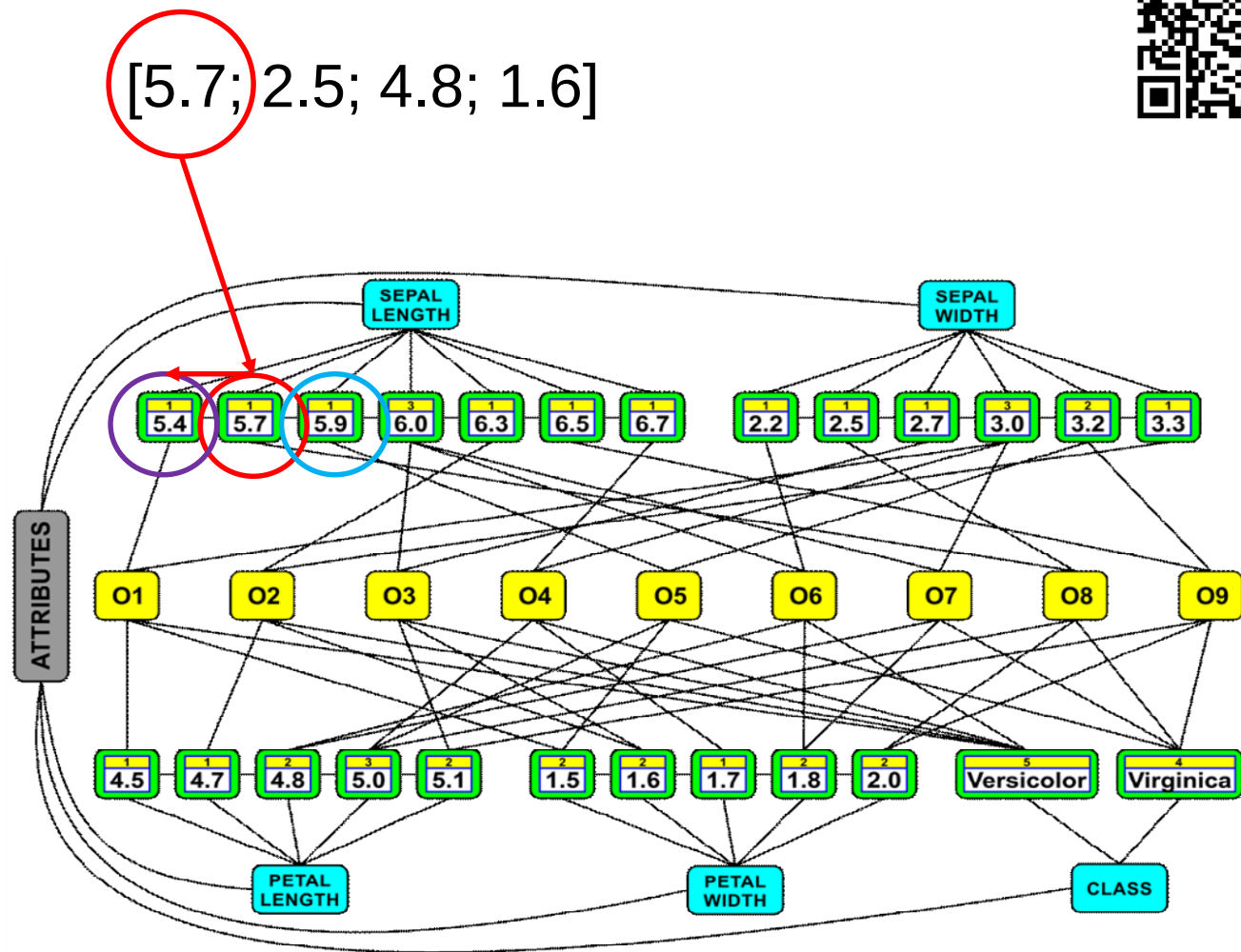


Rank table

1.	O8	0.45
2.	O5	0.48
3.		

$$d_e(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

$$d_m(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (2)$$



9. Po sprawdzeniu wszystkich węzłów obiektu połączonych z aktualnie wybranym węzłem wartości przedstawionym jako najbliższy pierwszej wartości atrybutu sklasyfikowanego obiektu z już nie przetworzonych węzłów wartości, **przejdź do następnego węzła najbliższej wartości** (reprezentującego niższą lub większą wartość do pierwsza wartość atrybutu sklasyfikowanego obiektu).

Przyspieszenie klasyfikatora KNN dzięki zastosowaniu struktur AGDS



Rank table

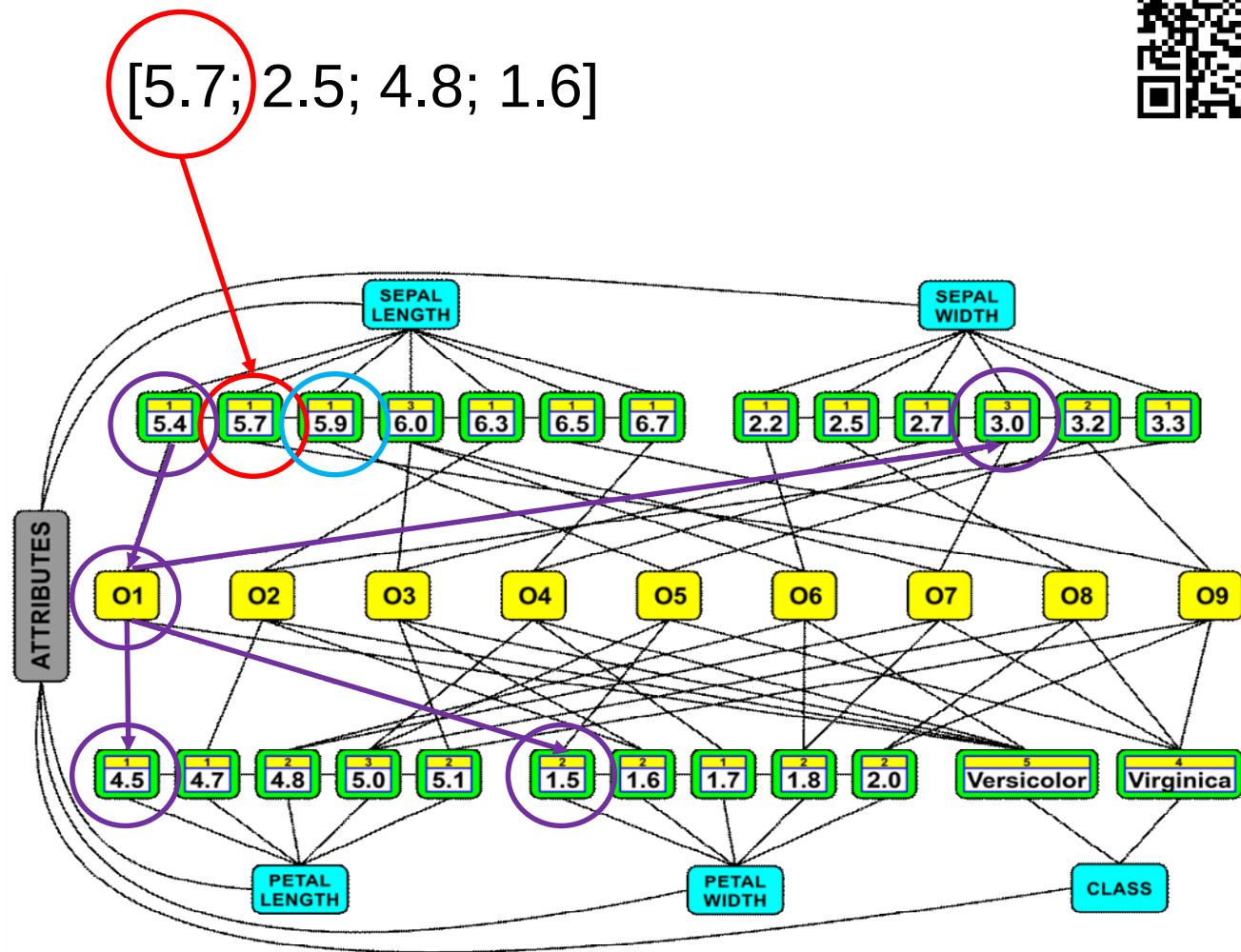
1.	O8	0.45
2.	O5	0.48
3.	O1	0.66

$$d_e(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

$$d_m(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (2)$$

$$d_e(x, y) = 0.66$$

$$d_m(x, y) = 1.2$$



9. Po sprawdzeniu wszystkich węzłów obiektu połączonych z aktualnie wybranym węzłem wartości przedstawionym jako najbliższy pierwszej wartości atrybutu sklasyfikowanego obiektu z już nie przetworzonych węzłów wartości, **przejdź do następnego węzła najbliższej wartości** (reprezentującego niższą lub większą wartość do pierwsza wartość atrybutu sklasyfikowanego obiektu).

Przyspieszenie klasyfikatora KNN dzięki zastosowaniu struktur AGDS

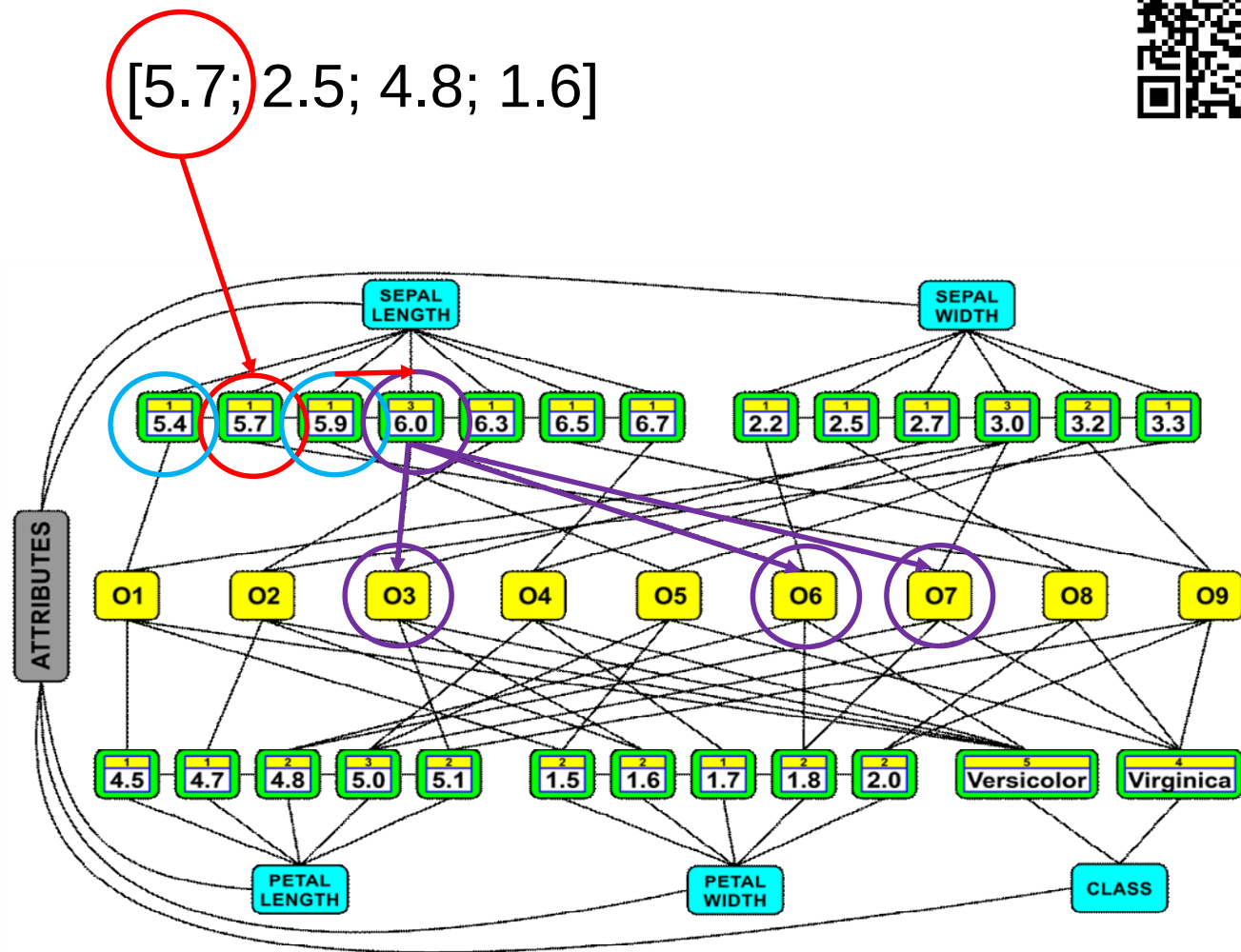


Rank table

1.	O8	0.45
2.	O5	0.48
3.	O1	0.66

$$d_e(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

$$d_m(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (2)$$



9. Po sprawdzeniu wszystkich węzłów obiektu połączonych z aktualnie wybranym węzłem wartości przedstawionym jako najbliższy pierwszej wartości atrybutu sklasyfikowanego obiektu z już nie przetworzonych węzłów wartości, **przejdź do następnego węzła najbliższej wartości** (reprezentującego niższą lub większą wartość do pierwsza wartość atrybutu sklasyfikowanego obiektu).

Przyspieszenie klasyfikatora KNN dzięki zastosowaniu struktur AGDS



Rank table

1.	O8	0.45
2.	O3	0.47
3.	O5	0.48
	O1	0.66

$$d_e(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

$$d_m(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (2)$$

O3 $d_e(x, y) = 0.47$

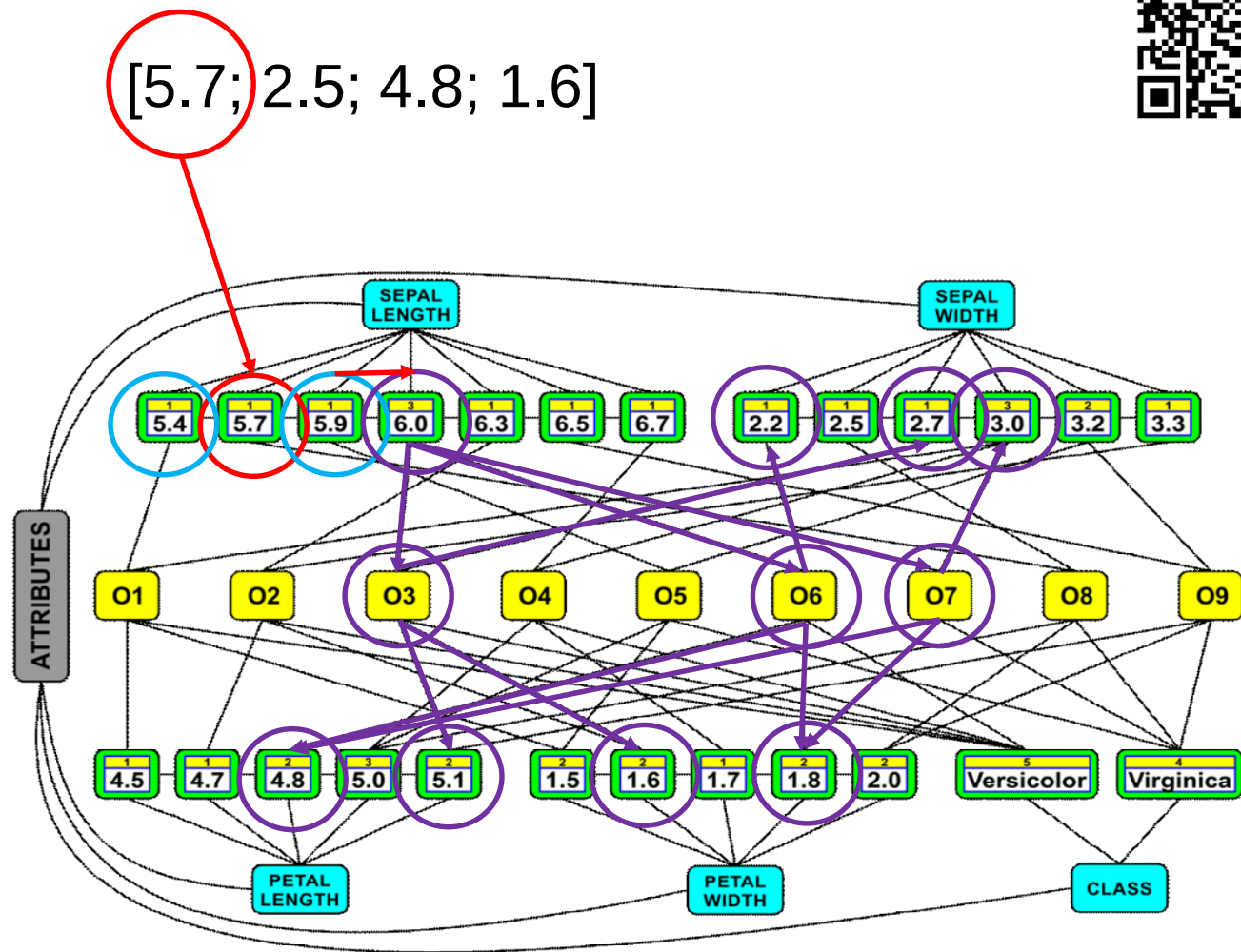
$d_m(x, y) = 0.8$

O6 $d_e(x, y) = 0.75$

$d_m(x, y) = 1.1$

O7 $d_e(x, y) = 0.62$

$d_m(x, y) = 1.0$



9. Po sprawdzeniu wszystkich węzłów obiektu połączonych z aktualnie wybranym węzłem wartości przedstawionym jako najbliższy pierwszej wartości atrybutu sklasyfikowanego obiektu z już nie przetworzonych węzłów wartości, **przejdź do następnego węzła najbliższej wartości** (reprezentującego niższą lub większą wartość do pierwsza wartość atrybutu sklasyfikowanego obiektu).

Przyspieszenie klasyfikatora KNN dzięki zastosowaniu struktur AGDS



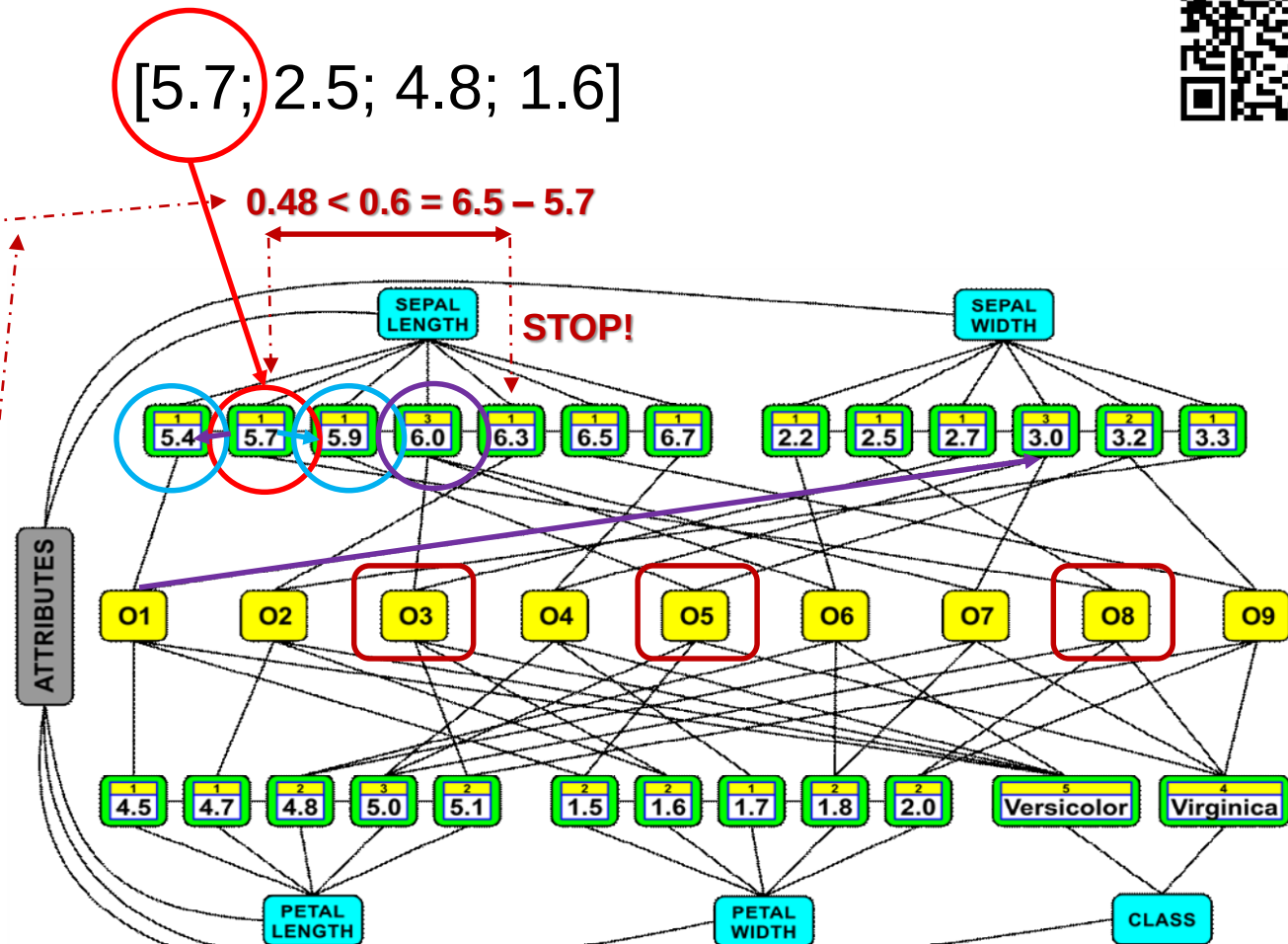
Rank table

1.	O8	0.45
2.	O3	0.47
3.	O5	0.48

K=3
NAJBLIŻSZYCH
SĄSIADÓW

$$d_e(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

$$d_m(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (2)$$



WARUNEK STOPU

10. Następnie przejdź do kroku 5, jeśli różnica między tymi dwiema wartościami jest mniejsza niż odległość ostatniego obiektu przechowywanego w tabeli rankingowej; w przeciwnym razie zakończ algorytm, ponieważ tabela rankingowa zawiera już k najbliższych sąsiadów wraz z ich odległościami od klasyfikowanego obiektu.

Przyspieszenie klasyfikatora KNN dzięki zastosowaniu struktur AGDS

Rank table

1.	O8	0.45	Virginica
2.	O3	0.47	Versicolor
3.	O5	0.48	Virginica

K=3
NAJBLIŻSZYCH
SĄSIADÓW

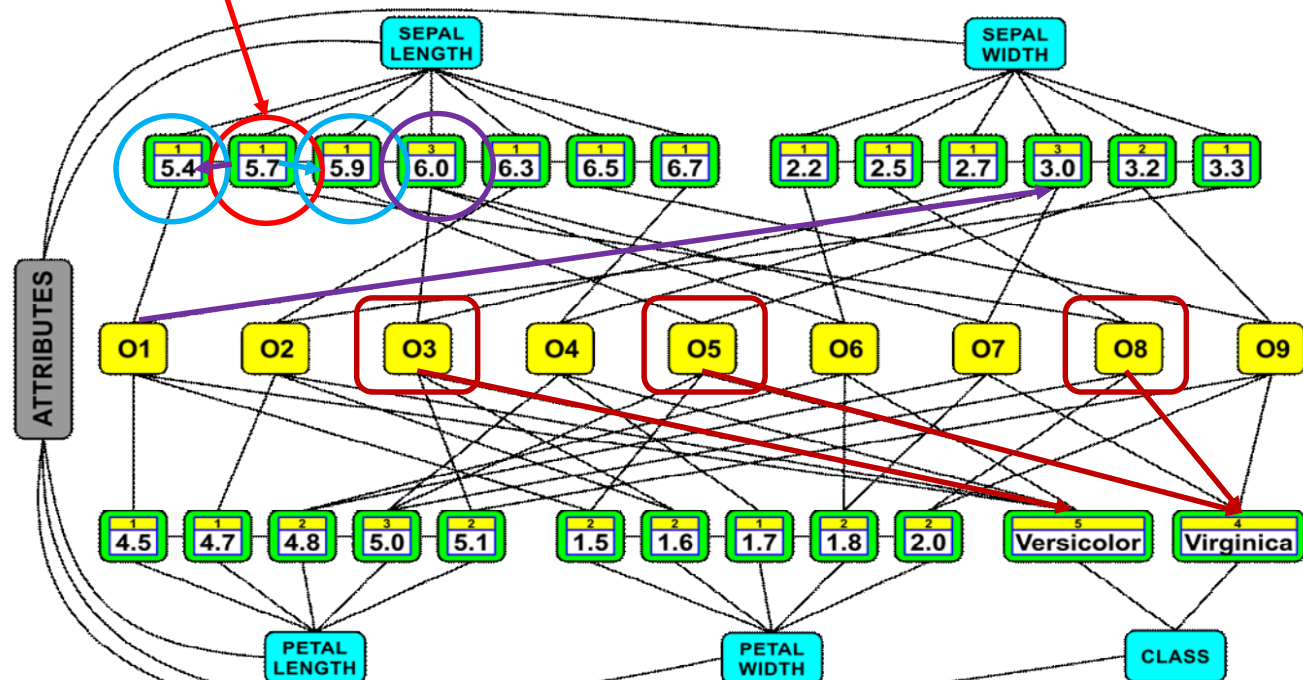
[5.7; 2.5; 4.8; 1.6]

GŁOSY

2 x Virginica
1 x Versicolor

ZWYCIĘSKA
KLASA

SAMPLE OBJECTS	ATTRIBUTES				CLASS LABEL
	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	
O1	5.4	3.0	4.5	1.5	Versicolor
O2	6.3	3.3	4.7	1.6	Versicolor
O3	6.0	2.7	5.1	1.6	Versicolor
O4	6.7	3.0	5.0	1.7	Versicolor
O5	6.0	2.2	5.0	1.5	Virginica
O6	5.9	3.2	4.8	1.8	Versicolor
O7	6.0	3.0	4.8	1.8	Virginica
O8	5.7	2.5	5.0	2.0	Virginica
O9	6.5	3.2	5.1	2.0	Virginica



11. Policz głosy i sprawdź, która klasa ma najwięcej głosów, w celu ustalenia zwycięskiej klasy i uzyskania klasyfikacji dla KNN.

Porównanie Wyników i Wydajności



Efektywność czasowa:

Table 2. Comparison of classification time using kNN and kNN+AGDS.

Dataset	Number of instances	Number of attributes	kNN classification time [ms]	kNN+AGDS classification time [ms]	kNN+AGDS construction time [ms]
Iris	150	4	0.10	0.08	1
Banknote	1372	4	0.29	0.09	5
HTRU2	17898	8	3.14	0.09	134
Shuttle	43500	9	7.67	1.06	278
Credit Card	30000	23	8.69	1.07	499
Skin	245057	3	26.87	1.10	683
Drive	58509	48	46.15	1.24	2224
HEPMASS	1048576	28	362.32	1.41	31214

Czas klasyfikacji dla KNN stosującego struktury AGDS jest prawie stały bez względu na rozmiar używanych zestawów danych treningowych.

Porównanie Wyników i Wydajności



Wydajność czasowa jako funkcja liczby instancji i atrybutów:

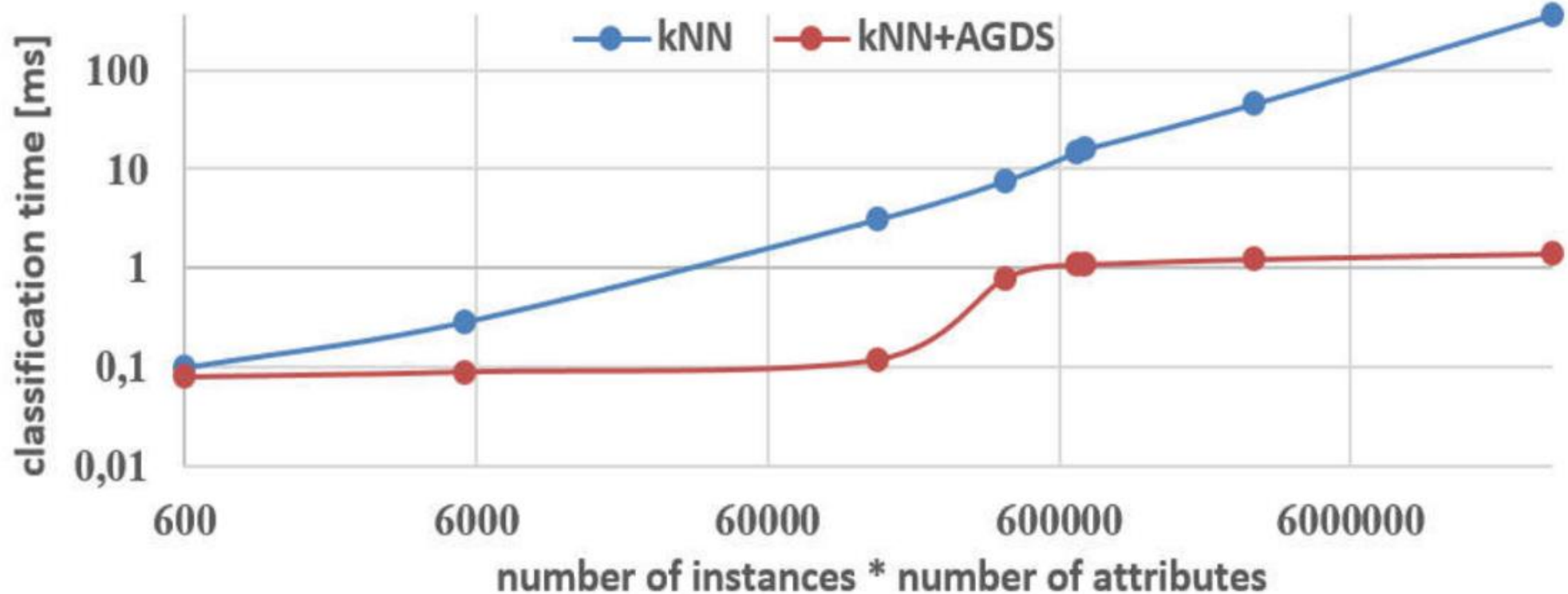


Fig. 2. Classification time as a function of the number of the instances multiplied by the number of attributes, i.e. the number of data stored in the training data tables.

Rozmiar danych uczących i liczba atrybutów nie wpływają istotnie na wydajność KNN podczas stosowania struktur AGDS, jak ma to miejsce w klasycznych klasyfikatorach KNN.

Porównanie Wyników i Wydajności



Efektywność pamięciowa algorytmu:

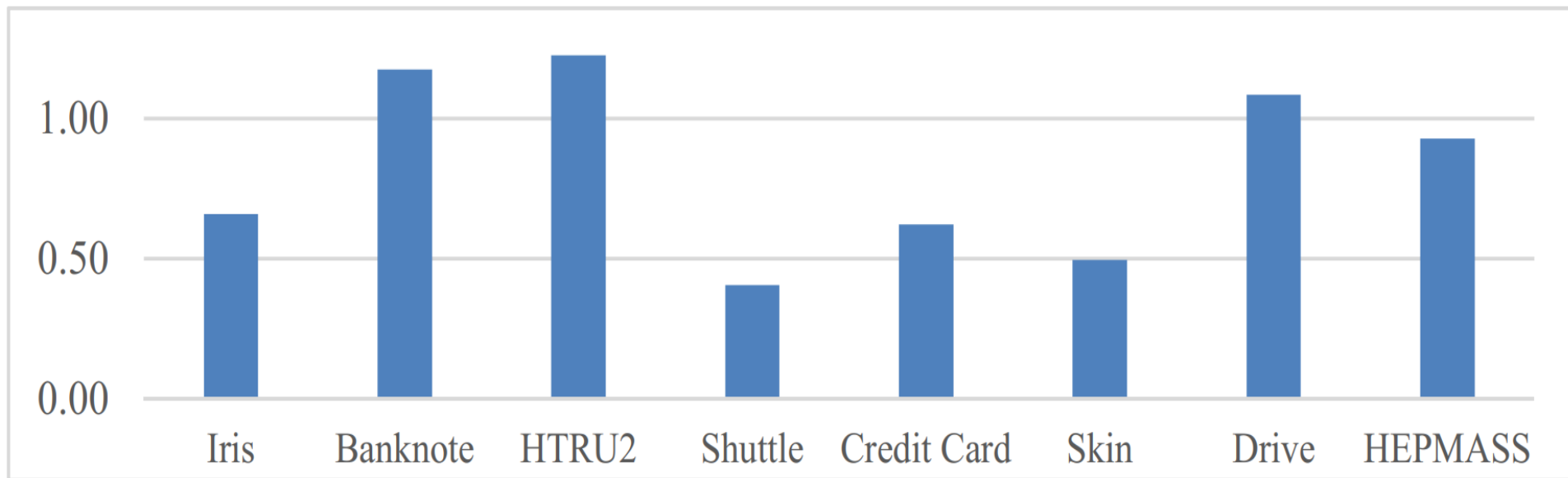


Fig. 3. Memory usage ratio of using AGDS structures to arrays for various training data.

Klasyfikatory KNN + AGDS zwykle zużywają mniej pamięci niż klasyfikatory KNN ze względu na liczbę zduplikowanych wartości w danych treningowych.

Podsumowanie i Ważne Uwagi



- ✓ Struktury AGDS zapewniają szybki dostęp do wartości sąsiednich i podobnych obiektów ze względu na agregacje i uporządkowanie wszystkich wartości jednocześnie dla wszystkich atrybutów.
- ✓ AGDS przechowuje dane wraz z najbardziej powszechnymi relacjami pionowymi i poziomymi, więc nie ma potrzeby zapętlania i poszukiwania tych relacji marnując zasoby.
- ✓ Typowe operacje na strukturach AGDS przyjmują czas logarytmiczny $O(\log)$ wyznaczony względem liczby unikalnych wartości atrybutów, lecz oczekiwana złożoność dla rzeczywistych danych zawierających wiele duplikatów jest zwykle stała $O(1)$.
- ✓ Wydajność prezentowanego algorytmu klasyfikacji stosującego struktury AGDS rośnie wraz z ilością danych szkoleniowych w porównaniu z klasycznymi algorytmami KNN.

Algorytm, pseudokod ... można znaleźć w artykule naukowym



Associative Graph Data Structures Used for Acceleration of K Nearest Neighbor Classifiers

Adrian Horzyk^(✉)  and Krzysztof Gołdon 

AGH University of Science and Technology, Krakow, Poland
horzyk@agh.edu.pl, krzysztofgoldon@gmail.com

Abstract. This paper introduces a new associative approach for significant acceleration of k Nearest Neighbor classifiers (kNN). The kNN classifier is a lazy method, i.e. it does not create a computational model, so it is inefficient during classification using big training data sets because it requires going through all training patterns when classifying each sample. In this paper, we propose to use Associative Graph Data Structures (AGDS) as an efficient model for storing training patterns and their relations, allowing for fast access to nearest neighbors during classification made by kNNs. Hence, the AGDS significantly accelerates the classification made by kNNs, especially for large and huge training datasets. In this paper, we introduce an Associative Acceleration Algorithm and demonstrate how it works on this associative structure substantially reducing the number of checked patterns and quickly selecting k nearest neighbors for kNNs. The presented approach was compared to classic kNN approaches successfully.

```
FindNextClosest(val)
if (valueNodeLessClosest == null) and (valueNodeGreaterClosest == null)
then
    valueNodeLessClosest = BinSearchEqualOrLess(val)
    if (valueNodeLessClosest == null)
    then valueNodeGreaterClosest = First
        return valueNodeGreaterClosest
    else if (valueNodeLessClosest.Val == val)
    then valueNodeGreaterClosest = valueNodeLessClosest
        return valueNodeGreaterClosest
    else if (valueNodeLessClosest.IsNotMax)
    then valueNodeGreaterClosest = valueNodeLessClosest.Next
        if (val - valueNodeLessClosest.Val < valueNodeGreaterClosest.Val -
            val)
        then return valueNodeLessClosest
        else return valueNodeGreaterClosest
    else valueNodeGreaterClosest = null
        return valueNodeLessClosest
else if (val - valueNodeLessClosest.Val < valueNodeGreaterClosest.Val - val)
then if (valueNodeLessClosest.IsNotMin)
    then valueNodeLessClosest = valueNodeLessClosest.Prev
    else if (valueNodeGreaterClosest.IsNotMax)
    then valueNodeGreaterClosest = valueNodeGreaterClosest.Next
        else return null
else if (valueNodeGreaterClosest.IsNotMax)
    then valueNodeGreaterClosest = valueNodeGreaterClosest.Next
    else if (valueNodeLessClosest.IsNotMin)
    then valueNodeLessClosest = valueNodeLessClosest.Prev
        else return null
if (val - valueNodeLessClosest.Val < valueNodeGreaterClosest.Val - val)
then return valueNodeLessClosest
else return valueNodeGreaterClosest
```

A. Horzyk and K. Gołdon, Associative Graph Data Structures Used for Acceleration of K Nearest Neighbor Classifiers, In: 27th International Conference on Artificial Neural Networks (ICANN 2018), Springer-Verlag, LNCS 11139, pp. 648-658, 2018.

Bibliografia and Literatura

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016, ISBN 978-1-59327-741-3 or PWN 2018.
2. Holk Cruse, Neural Networks as Cybernetic Systems, 2nd and revised edition.
3. R. Rojas, Neural Networks, Springer-Verlag, Berlin, 1996.
4. A. Horzyk, Associative Graph Data Structures with an Efficient Access via AVB+trees, In: 2018 11th International Conference on Human System Interaction (HSI), 2018, IEEE Xplore, pp. 169 - 175.
5. A. Horzyk and K. Gołdon, Associative Graph Data Structures Used for Acceleration of K Nearest Neighbor Classifiers, In: 27th International Conference on Artificial Neural Networks (ICANN 2018), Springer-Verlag, LNCS 11139, pp. 648-658, 2018.



**Akademia
Górnictwo-Hutnicza**



Adrian Horzyk
horzyk@agh.edu.pl

