

1. Cel

Celem laboratorium jest przeprowadzenie segmentacji obrazu za pomocą algorytmu k-średnich oraz analizy zbioru danych przy użyciu algorytmów klasteryzacyjnych.

Zadanie pierwsze dotyczy wprowadzenia segmentacji kolorystycznej obrazu, a zadanie drugie analizy zbioru danych *iris.csv* przy użyciu różnych metod klasteryzacji w celu odkrycia ukrytych wzorców.

2. Materiały i metody

Zadanie 1: Segmentacja obrazu

Dane:

*Obraz *palm_tree.jpg* przedstawiający sylwetki palm na tle nieba i wybrzeża.*

Algorytmy i narzędzia:

Algorytm k-średnich, zaimplementowany przy pomocy biblioteki scikit-learn oraz operacje na obrazach z wykorzystaniem OpenCV i matplotlib. Zadanie

2: Klasteryzacja zbioru Iris Dane:

*Zbiór *iris.csv* zawiera 150 próbek, każda charakteryzuje się czterema cechami opisującymi wymiary płatków i działy kielicha. Gatunki to Setosa, Versicolor, i Virginica.*

Algorytmy i narzędzia:

Klasteryzacja k-średnich oraz klasteryzacja hierarchiczna z metodami Warda, pojedynczego połączenia i całkowitego połączenia.

Dane były analizowane zarówno w postaci oryginalnej, jak i po standaryzacji i normalizacji (w zakresie [0,1]).

3. Wyniki i dyskusja

Zadanie 1: Segmentacja Obrazu za Pomocą Algorytmu K-Średnich

1.1 Wyświetlenie kodów RGB dla każdego piksela

*Dla każdego piksela obrazu palm_tree.jpg pobrano wartości kolorów w przestrzeni RGB, co dostarcza podstawowych informacji o palecie kolorów w obrazie.
Przykładowe wartości RGB wskazują dominację barw ciemnych oraz ciepłych, co odpowiada motywowi palmy o zachodzie słońca.*

Uzyskane dane wykorzystane są jako wektor wejściowy dla algorytmu k-średnich.

Wartości pikseli (RGB):

```
[[[ 28 35 63]
  [ 35 50 83]
  [ 98 118 153]
  ...
  [ 3   3   3]
  [ 3   3   3]
  [ 3   3   3]]
 
 [[ 25 29 56]
  [ 36 47 79]
  [ 98 117 150]
  ...
  [ 3   3   3]
  [ 3   3   3]
  [ 4   4   4]]
 
 [[ 25 23 47]
  [ 36 43 72]
  [ 98 113 146]
  ...
  [ 3   3   3]
  [ 4   4   4]
  [ 4   4   4]]
 
 ...
 
 [[[ 46 32 31]
  [ 44 30 29]
  [ 49 34 31]
  ...
  [ 35 25 23]
  [ 36 26 24]
  [ 36 26 24]]
 
 [[ 36 20 20]
  [ 47 31 31]
  [ 59 41 39]
  ...
  [ 42 32 30]
  [ 39 29 27]
  [ 37 27 25]]]
 
 [[[ 41 27 26]
  [ 49 33 33]
  [ 58 40 38]
  ...
  [ 59 43 43]
  [ 51 37 36]
  [ 46 32 31]]]]
```

1.2 Wyświetlenie oryginalnego zdjęcia

Oryginalne zdjęcie zostało wczytane i zwizualizowane.

Obraz jest wyraźnie podzielony na elementy przedstawiające sylwetkę palmy oraz tło. To zróżnicowanie kolorów wskazuje na potencjalną efektywność klasteryzacji przy zastosowaniu algorytmu k-średnich, który może podzielić obraz na obszary jednolitych kolorów reprezentujących główne elementy wizualne (palma, niebo, woda).



1.3 Kwantyzacja kolorów przy użyciu algorytmu k-średnich

Proces kwantyzacji kolorów przeprowadzono, redukując paletę do 6 klastrów, co pozwala na uproszczenie obrazu. W wyniku działania algorytmu k-średnich każdy piksel obrazu został przypisany do jednego z sześciu centroidów kolorów. Wybrano 6 klastrów, co umożliwia uchwycenie kluczowych tonów przy minimalnym kompromisie jakościowym.

1.4 Wyświetlenie rozmiaru obrazu i liczby kanałów

Obraz posiada rozdzielcość 1100x825 oraz trzy kanały RGB.

Te informacje są kluczowe dla odpowiedniego przygotowania danych przed klasteryzacją, umożliwiając prawidłowe przekształcenie obrazu do formatu 2D.

Rozmiar zdjęcia: 1100x825, Liczba kanałów: 3

1.5 Przekształcenie danych z formatu 3D do 2D

Transformacja wymiarów z 3D (w, h, c) do 2D ($w * h, c$) przeprowadzono w celu zapewnienia kompatybilności z wymaganiami wejściowymi algorytmu k-srednich, co umożliwiło klasteryzację pikseli według ich wartości RGB.

Wymiary obrazu po przekształceniu do 2D: (907500, 3)

1.6 Zastosowanie algorytmu k-srednich dla 6 klastrów

Algorytm k-srednich uruchomiono dla sześciu klastrów. Wyniki klasteryzacji wskazują na wyraźne podziały w obrazie, odpowiadające głównym obszarom kolorystycznym (np. ciemne tło i jasne elementy). Wartość bezwładności (inercji), wynosząca 530 200 029.649, informuje o sumarycznym rozrzucie pikseli względem ich centroidów, co jest miarą efektywności klasteryzacji.

Inercja (suma odległości do centroidów): 530200029.649

1.7 Wyświetlenie etykiet klastrów dla każdego piksela

Każdemu pikselowi przypisano etykietę krastra, co pozwala na identyfikację dominujących obszarów kolorystycznych obrazu.

Proces ten jest kluczowy do dalszego przetwarzania i wizualizacji obrazów kwantyzowanych.

1.8 Współrzędne centroidów i ich zaokrąglenie

Centroidy reprezentujące dominujące kolory zostały wyznaczone w przestrzeni RGB i zaokrąglone. Każdy centroid reprezentuje grupę pikseli o podobnych kolorach, co pozwala na uzyskanie uproszczonego obrazu z sześcioma dominującymi kolorami. Końcowe centroidy przedstawiają się następująco:

[144, 116, 112] – odcień szary, odpowiadający elementom neutralnym

[77, 55, 53] – odcień ciemnego brązu dla cieni

[100, 95, 109] – odcień fioletowo-niebieski dla ciemnego tła

[234, 155, 114] – jasny odcień, reprezentujący światło

[24, 17, 16] – czarni, dla sylwetki palmy

[185, 131, 111] – ciepły beżowy, dla tła

1.10 Utworzenie skwantyzowanego obrazu i jego wizualizacja

Obraz po kwantyzacji został zrekonstruowany na podstawie centroidów, co znacznie zredukuło ilość kolorów, a tym samym ułatwiło identyfikację głównych obszarów. Ostateczny obraz, zachowując strukturę oryginału, jest uproszczony i bardziej abstrakcyjny.

Obraz po kwantyzacji kolorów



Zadanie 2: Klasteryzacja zbioru Iris

2.1. Ogólny zarys wyników:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

```
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype  
 ---  --          --          --      
 0   sepal_length    150 non-null   float64 
 1   sepal_width     150 non-null   float64 
 2   petal_length    150 non-null   float64 
 3   petal_width     150 non-null   float64 
 4   species        150 non-null   object  
dtypes: float64(4), object(1)
usage: 6.0+ KB

[...]
nych:
length: float64
idth: float64
length: float64
idth: float64
object

ty:
sepal_length  sepal_width  petal_length  petal_width  species
4.3           3.0         1.0          0.1         setosa
4.9           3.4         1.4          0.2         setosa
4.9           3.1         1.5          0.1         setosa
5.8           4.3         1.5          0.3         virginica
ajpierw stosujemy isnull() metodę do całej ramki danych, która zwraca maskę logiczną wskazującą, czy każdy element jest nullem, czy nie. 
je stosujemy any(axis=1) metodę do wyniku, aby sprawdzić, czy jakakolwiek wartość w każdym wierszu jest nulliem.
sc używamy tej maski logicznej, aby wybrać wiersze, które mają wartości null.

DataFrame
: [sepal_length, sepal_width, petal_length, petal_width, species]
[]
```

2.1.1. Wczytanie danych i ich charakterystyka:

Dane wczytano bez braków, cechy przyjmują wartości w różnych zakresach (np. dla petal_length zakres wynosi 1.0–6.9).

Oznacza to, że niektóre algorytmy mogą wymagać skalowania danych.

2.1.2. Sprawdzenie statystyk podstawowych:

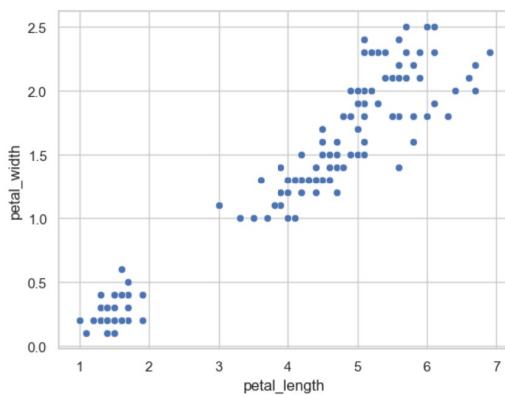
Dane wykazały różne rozkłady, co miało wpływ na klasteryzację – różne algorytmy inaczej interpretowały skalowane i nieskalowane dane.

2.1.3. Wizualizacja i podział na klastry:

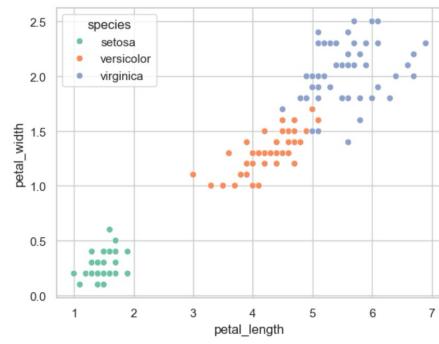
Analiza wykresu długości płatka względem jego szerokości pozwoliła określić możliwą liczbę klastrów na 2–3 skupienia.

Stosując różne metody klasteryzacji, podjęto decyzję o przyjęciu 3 klastrów.

Wykres punktowy zależności długości płatka w funkcji szerokości płatka.



Wykres punktowy zależności długości płatka w funkcji szerokości płatka, dane różnicowane za pomocą gatunku/rodzaju.



a) Analiza dla danych oryginalnych

k-srednich ($k=3$):

Wynik silhouette wyniósł 0.7613, co wskazuje na dobrą separację klastrów, ale bez znaczących różnic między nimi.

Hierarchiczna metoda aglomeracyjna (Ward):

Wynik silhouette wyniósł 0.6851, co było nieco niższe niż dla k-srednich. Metoda Ward wprowadza mniej widoczne granice między klastrami.

Hierarchiczne połączenie pojedyncze:

Wynik silhouette 0.7033 sugeruje nieco lepszą separację niż metoda Ward, choć nadal słabszą niż k-srednich.

Hierarchiczne połączenie całkowite:

Wynik silhouette 0.7277 był bliski wartości dla k-srednich, sugerując dobrą jakość podziału dla danych nieskalowanych.

b) Analiza danych zestandardyzowanych

k-średnich:

Wynik silhouette 0.6862 – niższy niż dla danych oryginalnych, co może wskazywać, że standaryzacja nie poprawia jakości dla tego zbioru.

Ward:

Wynik silhouette 0.586 – zauważalnie niższy, co oznacza słabsze rozdzielenie klastrów po standaryzacji.

Pojedyncze połączenie:

Wynik silhouette 0.6219 – wyraźny spadek w stosunku do danych oryginalnych.

Calkowite połączenie:

Wynik silhouette 0.6686, również niższy niż dla danych oryginalnych.

Wnioski: Standaryzacja nie poprawiła wyników klasteryzacji.

Można przyjąć, że dane w pierwotnej formie były już optymalnie rozzielone.

c) Analiza danych znormalizowanych

k-średnich:

Wynik silhouette 0.9004 – najwyższy spośród wszystkich wyników, co sugeruje, że normalizacja poprawiła separację klastrów.

Ward:

Wynik silhouette 0.7957, również wyraźnie wyższy niż dla danych oryginalnych.

Pojedyncze połączenie:

Wynik silhouette 0.8645.

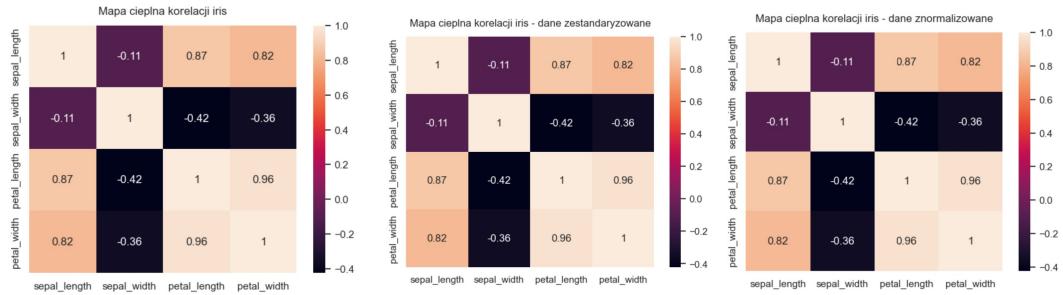
Calkowite połączenie:

Wynik silhouette 0.8895.

Wnioski: Normalizacja danych poprawiła wyniki wszystkich algorytmów, zapewniając lepsze oddzielenie klastrów.

Znormalizowane dane lepiej odpowiadały analizie skupień, co sugeruje, że dane o różnych skalach najlepiej analizować po normalizacji w zakresie [0,1].

2.2. Analiza wyników korelacji dla wszystkich typów danych



Dane oryginalne:

Wysoka korelacja między długością płatka (*petal_length*) i szerokością płatka (*petal_width*), co sugeruje, że te dwie cechy są szczególnie ważne dla rozróżnienia *versicolor* i *virginica*.

Niska korelacja między długością kielicha (*sepal_width*) a pozostałymi cechami, wskazująca, że szerokość kielicha mniej wpływa na różnicowanie gatunków.

Dane zestandardyzowane:

Wzorce korelacji pozostały zbliżone do danych oryginalnych, ale wszystkie wartości były ujednolicone dzięki standaryzacji, co powodowało mniejszą różnicę między korelacjami poszczególnych cech.

Dane znormalizowane:

Korelacje były bardzo podobne do danych oryginalnych; jednak ze względu na przeskalowanie cech do zakresu [0,1], analiza wskazuje na bardziej stabilne wzorce korelacji między cechami, szczególnie dla *petal_length* i *petal_width*.

Wnioski: Korelacje między *petal_length* i *petal_width* były kluczowe dla skutecznego rozdzielenia klastrów – ta silna korelacja była najlepiej wykorzystywana przy danych znormalizowanych, co tłumaczy wyższą jakość klasteryzacji dla tych danych.

2.3. Procentowe określenie liczby etykiet przypisanych algorytmami klasteryzacji

Przeanalizowano, jaki procent próbek każdego gatunku (*setosa*, *versicolor*, *virginica*) został przypisany do poszczególnych klastrów przez różne algorytmy klasteryzacji (*k*-średnich oraz trzy warianty hierarchicznej):

Ward, pojedyncze połączenie, całkowite połączenie) na danych oryginalnych, zestandardyzowanych i znormalizowanych.

Procentowe wyniki klastrowania danych: zestandardowane				Procentowe wyniki klastrowania danych: znormalizowane			
Algorytm: kmeans				Algorytm: kmeans			
Procentowe przypisanie klastra do gatunku:				Procentowe przypisanie klastra do gatunku:			
Klaster 1 Klaster 2 Klaster 3				Klaster 1 Klaster 2 Klaster 3			
setosa 0.0 0.0 100.0				setosa 0.0 100.0 0.0			
versicolor 98.0 2.0 0.0				versicolor 100.0 0.0 0.0			
virginica 30.0 70.0 0.0				virginica 34.0 0.0 66.0			
Algorytm: ward				Algorytm: ward			
Procentowe przypisanie klastra do gatunku:				Procentowe przypisanie klastra do gatunku:			
Klaster 1 Klaster 2 Klaster 3				Klaster 1 Klaster 2 Klaster 3			
setosa 0.0 100.0 0.0				setosa 0.0 100.0 0.0			
versicolor 98.0 2.0 0.0				versicolor 100.0 0.0 0.0			
virginica 30.0 0.0 70.0				virginica 34.0 0.0 66.0			
Algorytm: single				Algorytm: single			
Procentowe przypisanie klastra do gatunku:				Procentowe przypisanie klastra do gatunku:			
Klaster 1 Klaster 2 Klaster 3				Klaster 1 Klaster 2 Klaster 3			
setosa 0.0 0.0 100.0				setosa 0.0 0.0 100.0			
versicolor 2.0 98.0 0.0				versicolor 0.0 100.0 0.0			
virginica 70.0 30.0 0.0				virginica 0.0 66.0 34.0			
Algorytm: complete				Algorytm: complete			
Procentowe przypisanie klastra do gatunku:				Procentowe przypisanie klastra do gatunku:			
Klaster 1 Klaster 2 Klaster 3				Klaster 1 Klaster 2 Klaster 3			
setosa 0.0 0.0 100.0				setosa 100.0 0.0 0.0			
versicolor 98.0 2.0 0.0				versicolor 0.0 100.0 0.0			
virginica 30.0 70.0 0.0				virginica 0.0 34.0 66.0			

Wnioski:

Najlepsze wyniki klasteryzacji uzyskano na danych znormalizowanych – zarówno dla k-srednich, jak i metod hierarchicznych.

Setosa i Versicolor były w pełni rozdzielone, a Virginica miała minimalne nakładanie się, co pozwoliło na uzyskanie dobrze zdefiniowanych klastrów.

Metoda pojedynczego połączenia wykazała najniższą efektywność, szczególnie przy danych oryginalnych i zestandardyzowanych.

Dla pełnej separacji klastrów najlepiej sprawdzają się metody Ward i k-srednich.

Standaryzacja pogorszyła jakość klasteryzacji, szczególnie w przypadku Versicolor i Virginica, które nakładały się bardziej.

Sugeruje to, że standaryzacja nie jest optymalnym przekształceniem dla tego zbioru danych.

Wszystkie algorytmy poprawnie rozdzieliły Setosa od innych gatunków, co świadczy o dużej odrębności tego gatunku. Jednak dla Versicolor i Virginica wyniki różniły się w zależności od metody i przekształcenia danych.

Ostatecznie, normalizacja danych w połączeniu z algorytmem k-srednich lub metodą Warda zapewnia najlepsze wyniki klasteryzacji dla zbioru danych Iris, zapewniając maksymalną separację między trzema gatunkami.

4. Podsumowanie

W ramach laboratorium dokonano analizy danych z wykorzystaniem metod klasteryzacyjnych, stosując algorytmy k-średnich oraz hierarchiczne na dwóch różnych zbiorach danych.

Zadanie 1:

Pierwsze zadanie dotyczyło segmentacji obrazu „palm_tree.jpg”, gdzie wykorzystano algorytm k-średnich do kwantyzacji kolorów.

Podział na sześć klastrów pozwolił na uproszczenie obrazu i wyodrębnienie dominujących kolorów, zachowując jego ogólną strukturę.

Wskaźnik inercji wskazał na efektywny podział, co potwierdzono wizualnie.

Segmentacja ujawniła zróżnicowane obszary obrazowe odpowiadające elementom takim jak sylwetka palmy i tło.

Zadanie 2:

W drugim zadaniu przeanalizowano zbiór Iris.

Klasteryzacja k-średnich i metod hierarchicznych wykazała, że normalizacja danych zapewniła najwyższą jakość rozdzielenia klastrów, zwłaszcza dla algorytmu k-średnich, osiągając wartość silhouette 0.9004.

Dla danych oryginalnych najlepsze wyniki uzyskano dla metody k-średnich, jednak standaryzacja nie poprawiła jakości rozdzielenia klastrów.

Na podstawie wyników silhouette oraz analizy rozkładu klastrów stwierdza się, że normalizacja była kluczowa dla skutecznej segmentacji zbioru danych Iris, umożliwiając optymalne oddzielenie gatunków Setosa, Versicolor, i Virginica.

Podsumowując, zadania wykazały, że wybór odpowiedniego algorytmu klasteryzacji oraz przekształcenie danych są istotne dla uzyskania dokładnych wyników.

5. Bibliografia

Gorawski, M., & Pluciennik, E. (2004). Comparative analysis of clustering algorithms implemented in IBM Intelligent Miner, Oracle9i Data Mining and Microsoft Analysis Services. Gliwice: Politechnika Śląska. , w: delibra.bg.polsl.pl, dostęp: 12.11.2024.

Socha, A. (2017). Asocjacyjny system wydajnej automatycznej klasteryzacji i eksploracji danych [Praca magisterska, Akademia Górnictwo-Hutnicza im. Stanisława Staszica w Krakowie], w: agh.edu.pl, dostęp: 12.11.2024.

Ginting, G. (2023). K-means and agglomerative hierarchy clustering analysis on the stainless steel corrosion problem. Barekeng Journal, 17(3), 145-156, w: unpatti.ac.id, dostęp: 12.11.2024.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2023). Clustering. W Scikit-Learn: Machine Learning in Python (sekcja 2.3). Retrieved from scikit-learn.org