

MPUM mini projekt 4 - Klasteryzacja

Tymoteusz Siemieniuk

Czerwiec 2024

1 Definicja problemu

Projekt ten ma za zadanie zbadanie trzech algorytmów klasteryzacji:

1. kmeans(++)
2. klastrowanie hierarchiczne
3. klasteryzacja spektralna

Algorytmy te będą testowane na kilku zbiorach danych o różnej wielkości i wymiarowości.

2 Opis algorytmów

2.1 (n)kmeans(++)

Zaimplementowałem kmeans oraz kmeans++. Oba te algorytmy mogą być dodatkowo odpalone wielokrotnie (domyślnie 20 razy) w celu znalezienia ustawienia początkowego dającego najlepsze wyniki. Porównywanie klasteryzacji zrealizowane jest w następujący sposób: im mniejsza jest średnia odległość punkt-centrum klastra do którego został on zakwalifikowany, tym lepsza jest klasteryzacja. Zatem finalna klasteryzacja to najlepsza spośród 20 przebiegów. Takie wielokrotne odpalanie kmeans(++) będę nazywał nkmeans(++) .

2.2 klastrowanie hierarchiczne

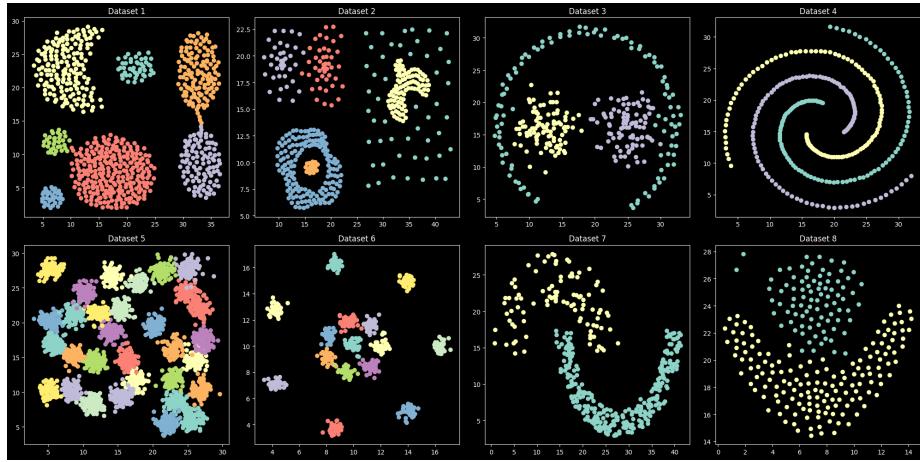
Zaimplementowałem klastrowanie hierarchiczne algorytmem Lance'a Williams'a. Funkcja ta przyjmuje parametr k , będący liczbą klastrów, mogący być równy 1 lub więcej. Oprócz samej klasteryzacji (lista etykiet) funkcja ta zwraca u mnie również strukturę drzewa (lasu dla $k > 1$) które powstało podczas przebiegu algorytmu.

2.3 klasteryzacja spektralna

Zaimplementowałem również klasteryzację spektralną zgodnie z opisem w artykule "On Spectral Clustering: Analysis and an algorithm" autorstwa: Andrew Y. Ng, Michael I. Jordan, Yair Weiss. Domyślnie jako wagę krawędzi w grafie przyjmuje, tak jak jest to sugerowane w artykule: $w_{i,j} = e^{-\|x_i - x_j\|_2^2}$.

3 Dane 2D

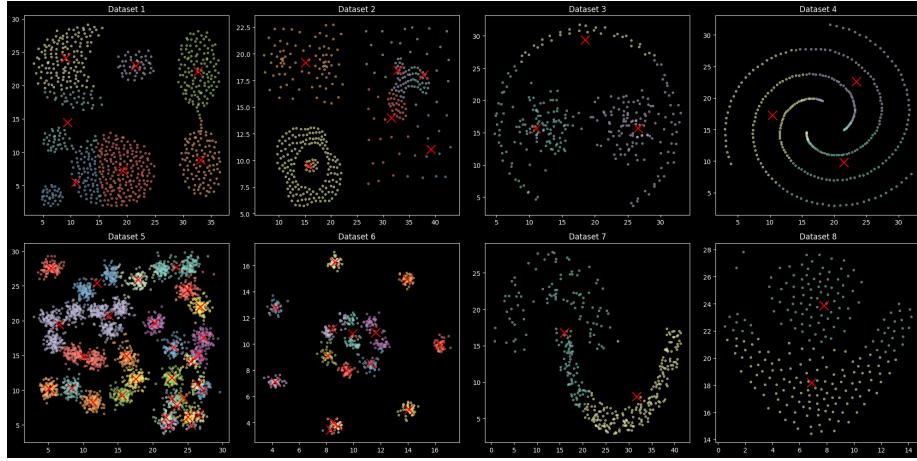
Zacznę od omówienia wyników tych algorytmów na danych dwuwymiarowych. Będziemy operować na następujących 8 zbiorach danych:



Rysunek 1: Zestawy danych 2D

3.1 kmeans

Oto wyniki klasteryzacji algorytmem kmeans - jako, że "idealne" klastrowanie jest znane, to odpalamy kmeans z liczbą klastrów równą tej wzorcowej.

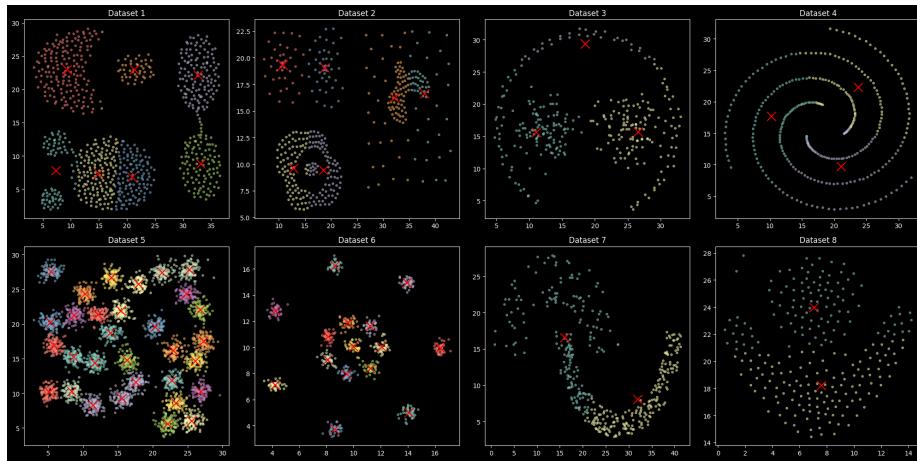


Rysunek 2: Wyniki klasteryzacji dla kmeans

Jak widać, ten algorytm w miarę radzi sobie z wypukłymi klastrami. Dąży on również do tego, że wszystkie klastry zajmują podobnej wielkości obszar przestrzeni. Kmeans poradził sobie w miarę dobrze z zbiorami 8, 5 i 6, jednak reszta wyszła dość kiepsko.

3.2 nkmeans++

nkmeans++ poradził sobie zdecydowanie lepiej ze zbiorami 5 i 6. Wciąż jednak jest wiele miejsca do poprawy.



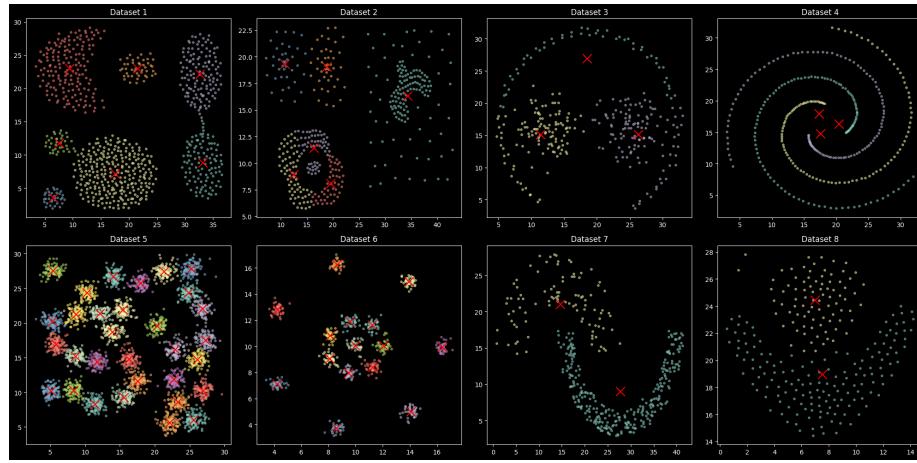
Rysunek 3: Wyniki dla nkmeans++

Wyniki dla tych algorytmów były praktycznie natychmiastowe - mniej niż

sekunda aby przetworzyć wszystkie 8 zestawów danych (brawo numpy!).

3.3 klasteryzacja spektralna

Klasteryzacja spektralna radzi sobie jeszcze lepiej, jednak wciąż ma problemy z zbiorem 2 i 3. Zwraca ona natomiast idealne wyniki dla niektórych zbiorów niewypukłych - 4 i 7, z czym kmeans miało problemy. Przetworzenie wszystkich zbiorów danych 2D zajęło tym razem 44 sekundy.



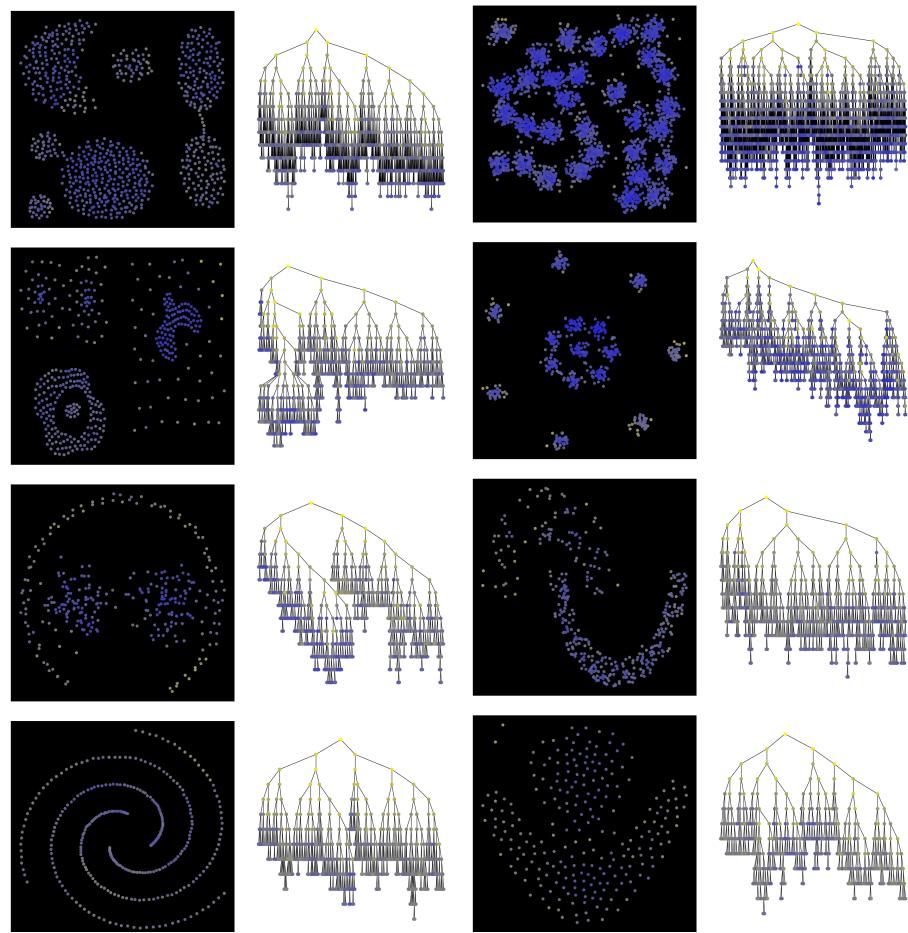
Rysunek 4: Wyniki dla klasteryzacji spektralnej

3.4 klastrowanie hierarchiczne

Klastrowanie hierarchiczne ma zdecydowanie najlepsze wyniki, jednak również liczy się ono najdłużej - na największym zbiorze danych (3k punktów) liczy się 14 minut. Tutaj są wyniki dla 'średniej' metody łączenia.

Próbowałem również pokolorować punkty w zależności od ich głębokości w drzewie, jednak to ambitne zadanie mnie przerosło - poległem na próbie komunikacji z biblioteką networkx.

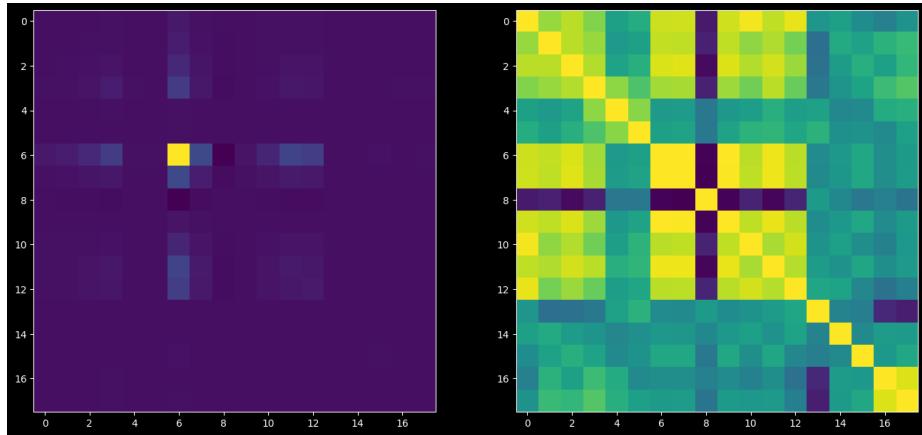
Powinno być mniej więcej tak, że izolowane wierzchołki będą bardziej żółte... może i rzeczywiście trochę tak jest.



Rysunek 5: Drzewa otrzymane po przebiegu klasteryzacji hierarchicznej

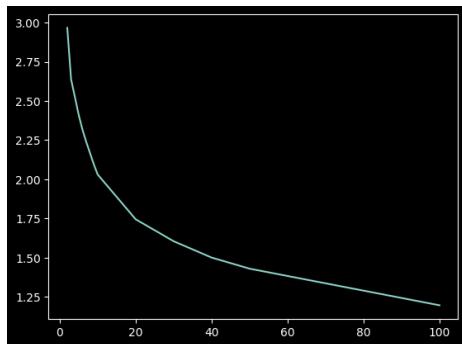
4 Dane 18D

Analizę danych zacząłem od narysowania macierzy kowariancji:

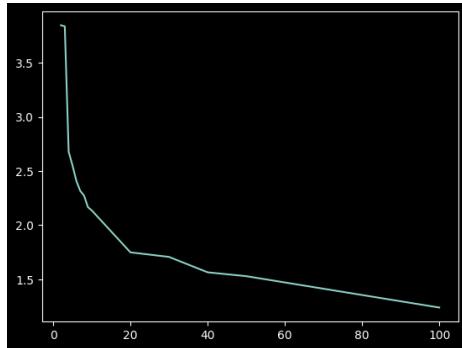


Rysunek 6: Macierze kowariancji odpowiednio dla nieznormalizowanych i znormalizowanych danych

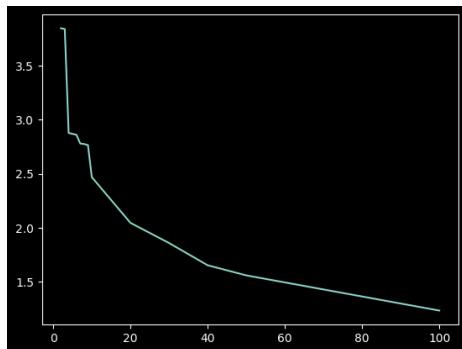
Następnie przystąpiłem do obliczenia funkcji błędu klasteryzacji dla różnych k oraz dla różnych algorytmów. Funkcja błędu klasteryzacji to tak jak poprzednio, średnia odległość wierzchołka od najbliższego centrum klastra. Wyniki wygładzają następująco:



Rysunek 7: Funkcja błędu w zależności od k dla kmeans++



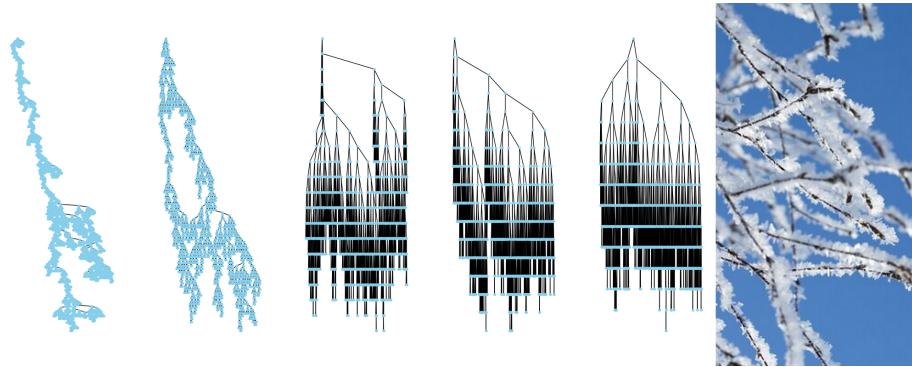
Rysunek 8: Funkcja błędu w zależności od k dla klasteryzacji spektralnej



Rysunek 9: Funkcja błędu w zależności od k dla klasteryzacji hierarchicznej

Klasteryzację spektralną i hierarchiczną dla wielu różnych k można napisać efektywniej niż to zrobiłem - w spektralnej na początku wykonywane są te same obliczenia, niezależnie od k, a w hierarchicznej większe k oznacza po prostu skończenie działania algorytmu wcześniej. Nie po to jednak ludzie wymyślili komputery, żeby się męczyć, ale po to, żeby to one męczyły się za nas.

Oto wykresy drzew powstały przy klasteryzacji hierarchicznej dla metryki euklidesowskiej oraz różnych metod łączenia klastrów.

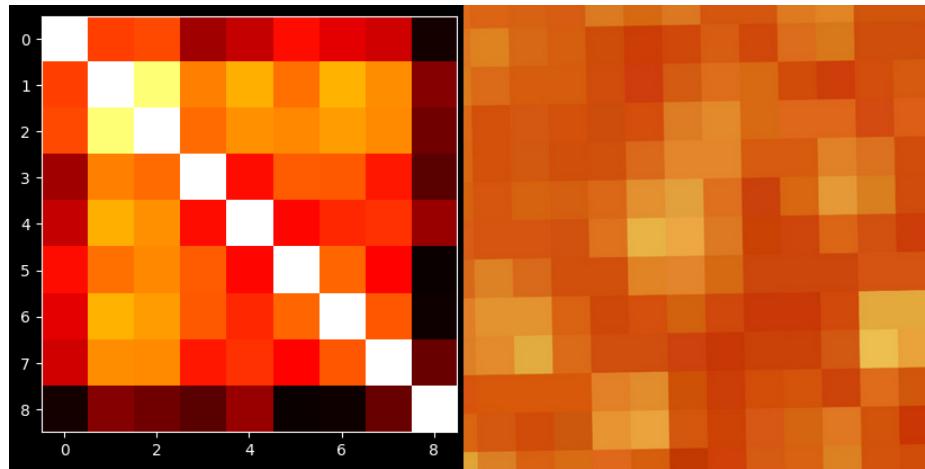


Rysunek 10: Drzewa hierarchiczne dla następujących metod łączenia: pojedyńcze, centroidalne, średnie, pełne, Warda. Najbardziej z prawej prawdziwe drzewo, dla porównania.

5 Dane rp

Przede wszystkim ciekawi mnie, co może oznaczać enigmatyczna nazwa 'rp_data'. W każdym razie oto wyniki błędu klasteryzacji w zależności od liczby klastrów dla różnych algorytmów:

Oto wykresy drzew, tak jak poprzednio:



Rysunek 11: Macierz kowariancji dla znormowanych danych. Po prawej tekstura lawy z minecraft, dla porównania.



Rysunek 12: Drzewa hierarchiczne dla następujących metod łączenia: pojedyńcze, centroidalne, średnie, pełne, Warda.

Długie ścieżki na dołach drzew mogą świadczyć o tym, że istnieją grupy punktów o bardzo dużym podobieństwie.

6 Bonus - wykrywanie dominujących kolorów

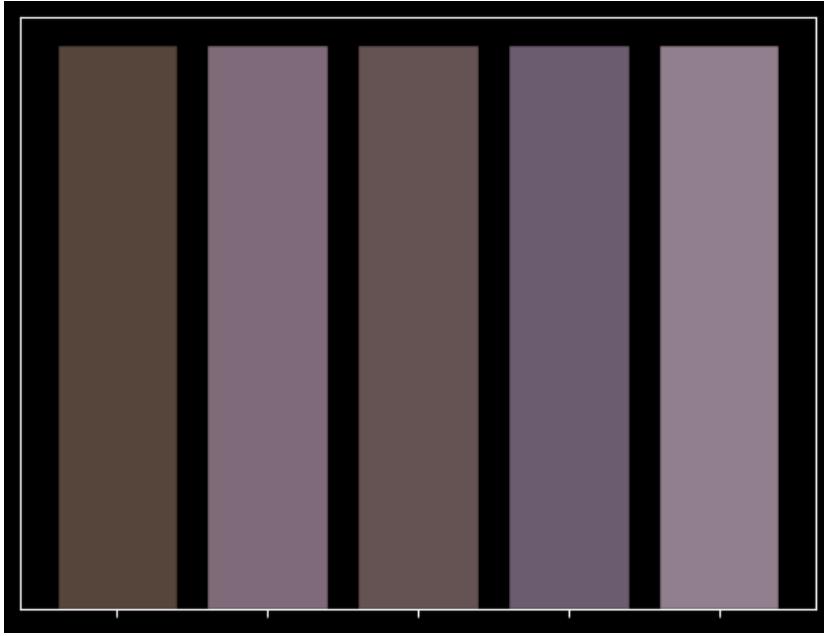
Ciekawym zastosowaniem klasteryzacji wydaje mi się wykrywanie dominujących kolorów na obrazach/zdjęciach. Główna idea polega na tym, że zbiór punktów które będziemy klasteryzować to po prostu zbiór kolorów pikseli - rozważałem dwa modele - RGB oraz CMYK. Warto zauważyć, że przy takim podejściu kompletnie ignorujemy pozycje pikseli - a kolory często zawdzięczają swój urok swojemu położeniu na tle innych. Ze względów praktycznych obrazy są skalowane w dół do rozmiaru około 30x30 pikseli - nie ma to większego wpływu na wyniki.

Analizę zacznę od obrazu Monet'a - "Domy parlamentu, zachód słońca". Jest to moim zdaniem dobry obraz do analizy, ponieważ znajduje się na nim czerwone malutkie słońce, którego kolor moim zdaniem powinien się znaleźć w kilku najbardziej dominujących kolorach, pomimo swojego małego rozmiaru. Jak się jednak okaże, nie będzie to takie oczywiste dla algorytmów klasteryzacji.

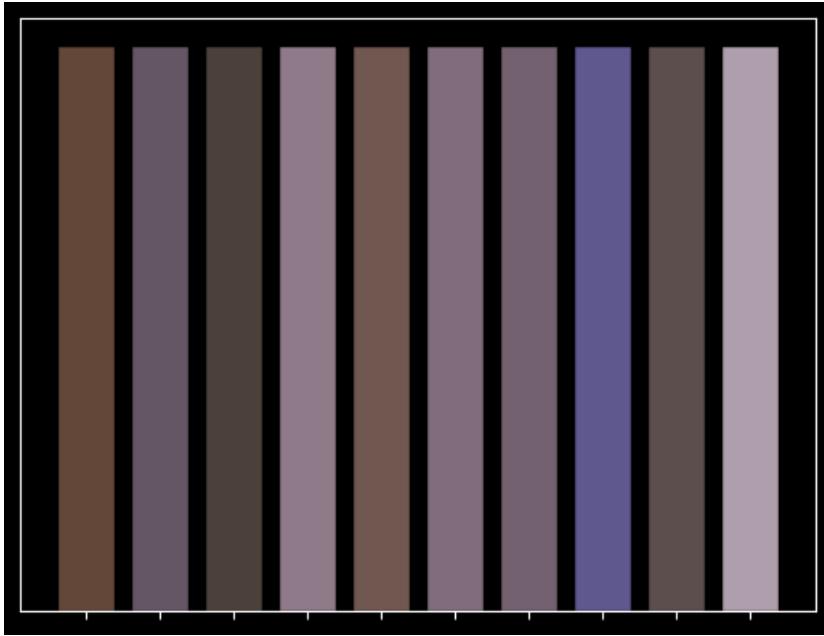


Rysunek 13: Claude Monet - Domy parlamentu, zachód słońca (1903)

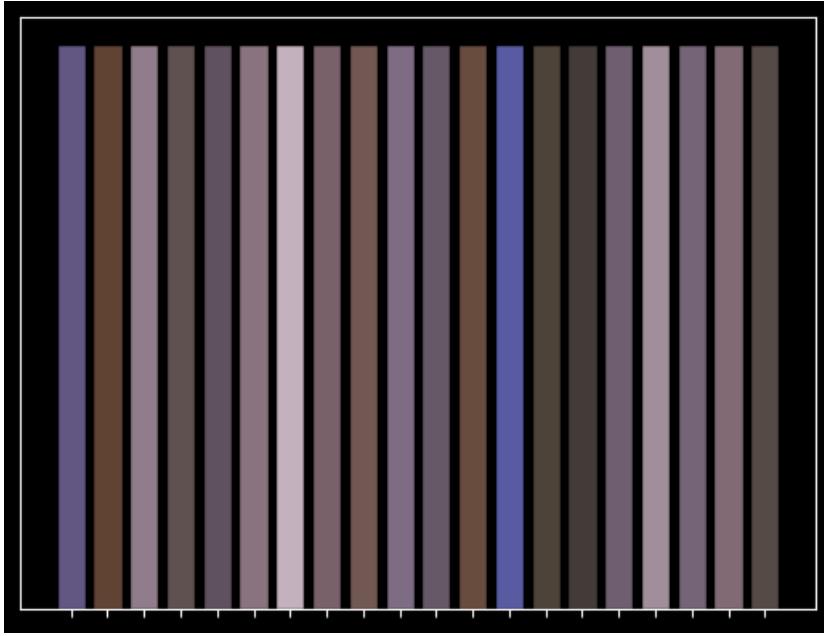
Algorytmy kmeans oraz klasteryzacja spektralna radzą sobie średnio z wykrywaniem dominujących kolorów. Nawet jeśli ustawimy $k=20$, to nie pojawia się kolor słońca (skandal(!)). Testowałem dla różnych metryk, jednak wyniki są kiepskie.



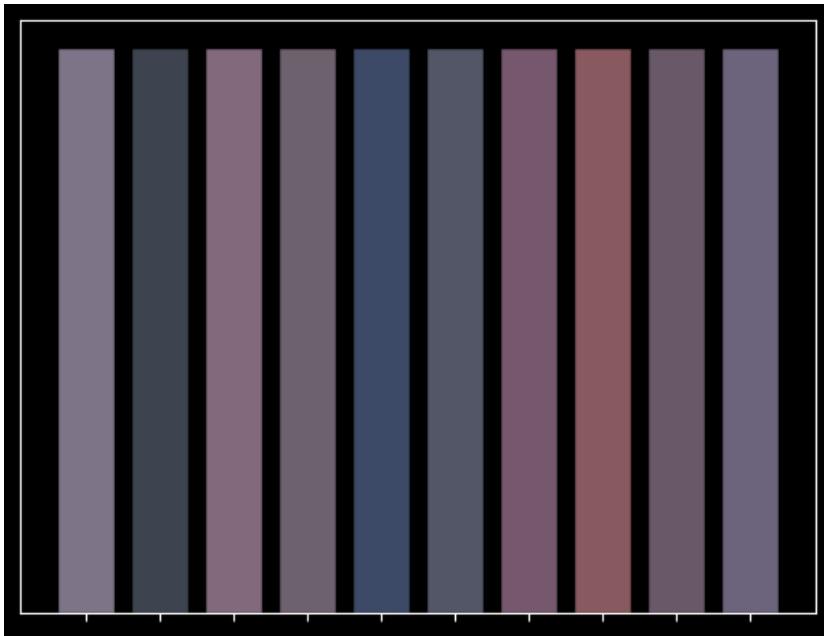
Rysunek 14: 5 dominujących kolorów wg kmeans



Rysunek 15: 10 dominujących kolorów wg kmeans



Rysunek 16: 20 dominujących kolorów wg kmeans

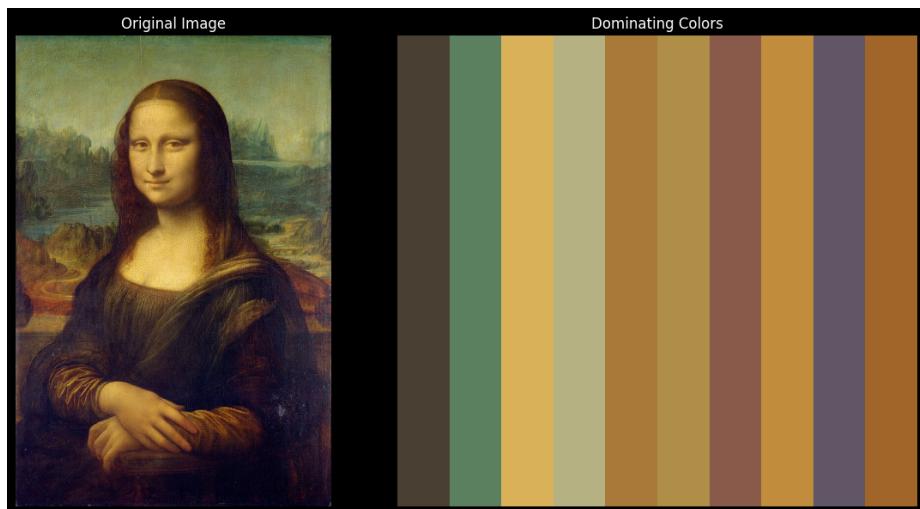


Rysunek 17: 10 dominujących kolorów wg klasteryzacji spektralnej

Na szczęście klasteryzacja hierarchiczna radzi sobie dobrze:



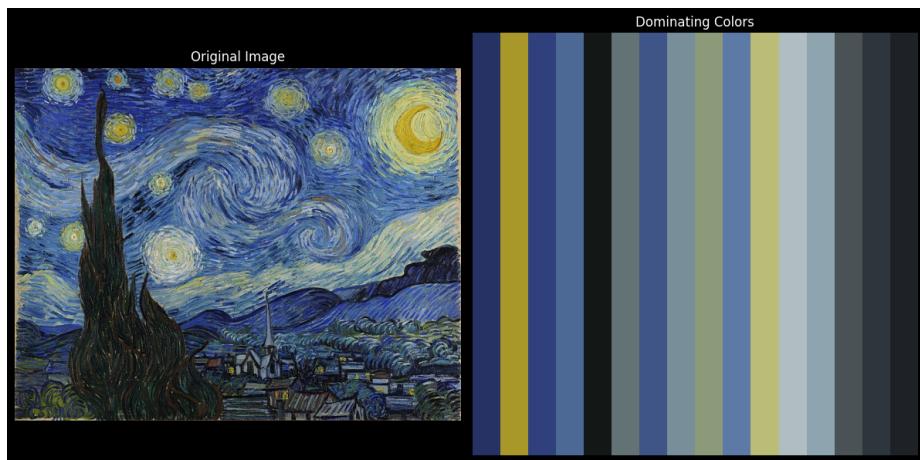
Rysunek 18: 10 dominujących kolorów wg klasteryzacji hierarchicznej, pojedyńcze łączenie



Rysunek 19: 10 dominujących kolorów wg klasteryzacji hierarchicznej, pojedyńcze łączenie



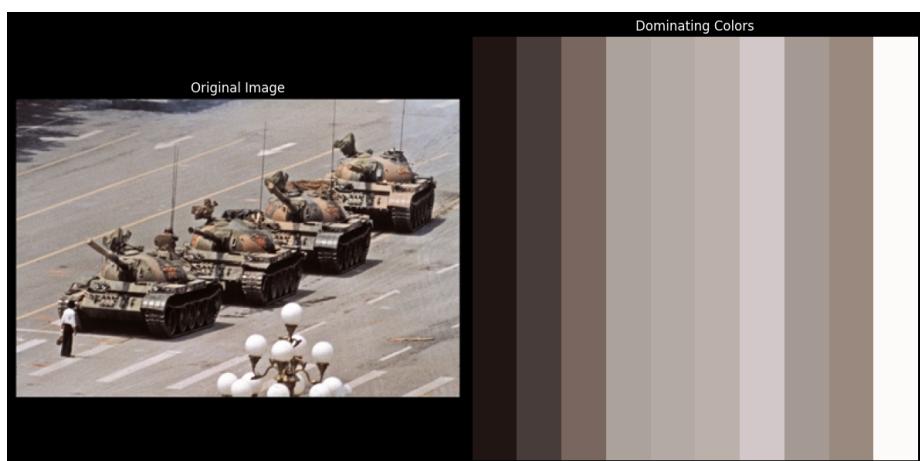
Rysunek 20: 5 dominujących kolorów wg klasteryzacji hierarchicznej, pojedyńczełączenie



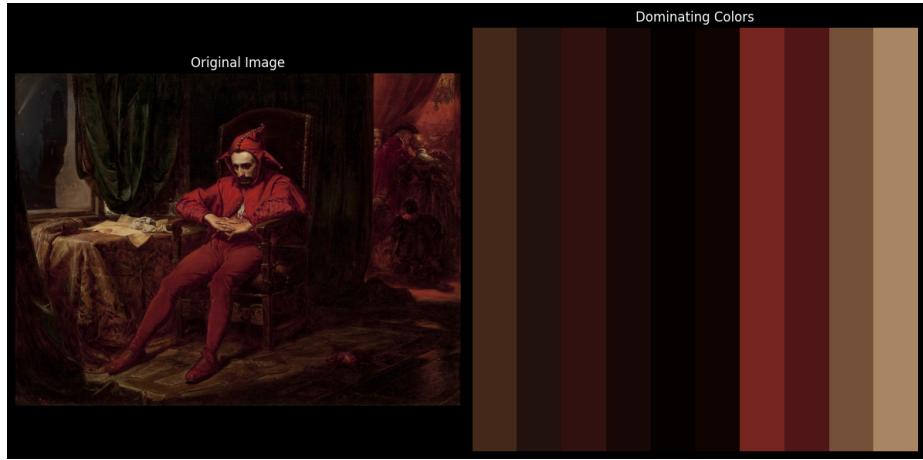
Rysunek 21: 16 dominujących kolorów wg klasteryzacji hierarchicznej, łączenie warda



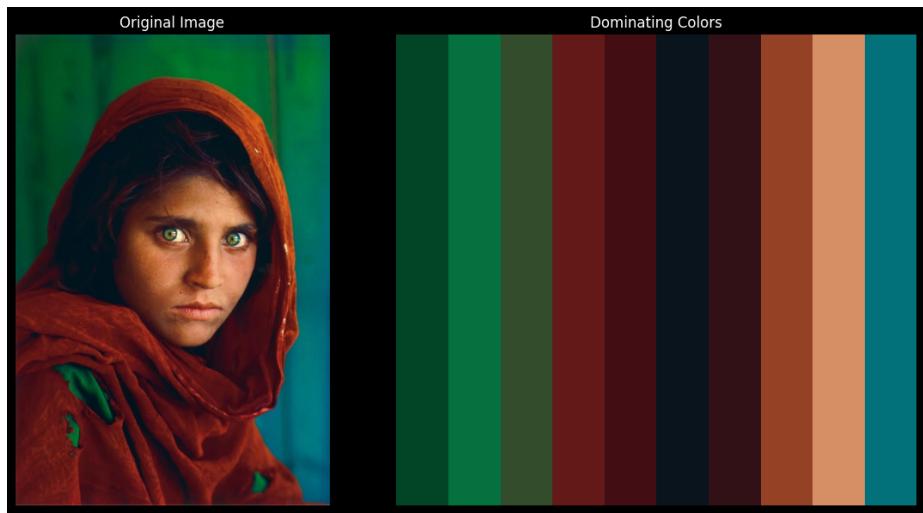
Rysunek 22: 10 dominujących kolorów wg klasteryzacji hierarchicznej, łączenie warda



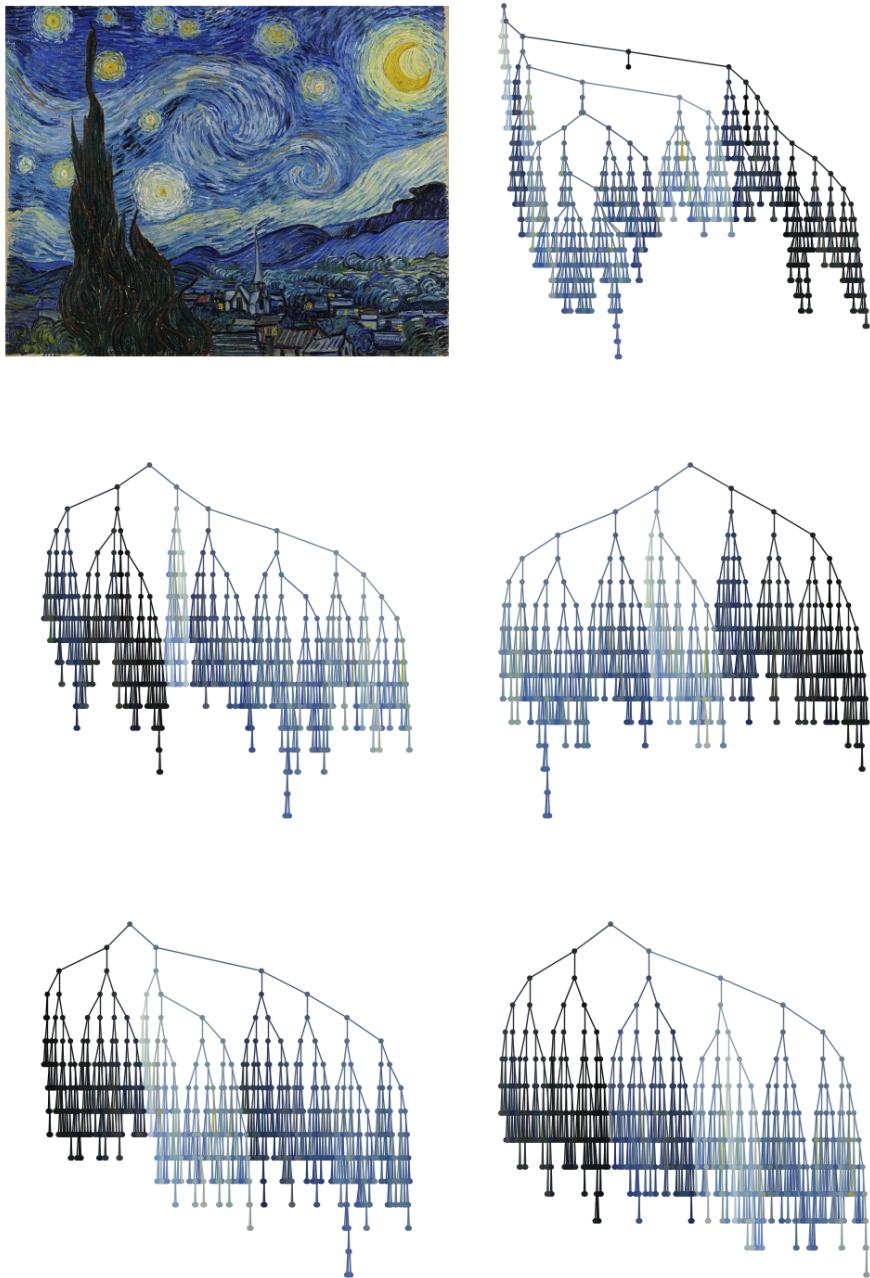
Rysunek 23: 10 dominujących kolorów wg klasteryzacji hierarchicznej, łączenie warda



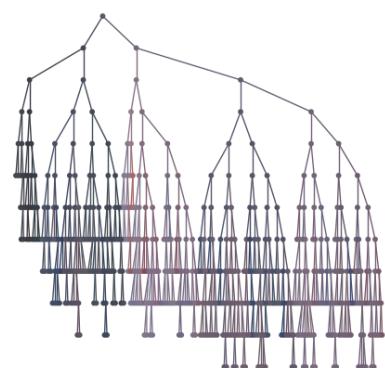
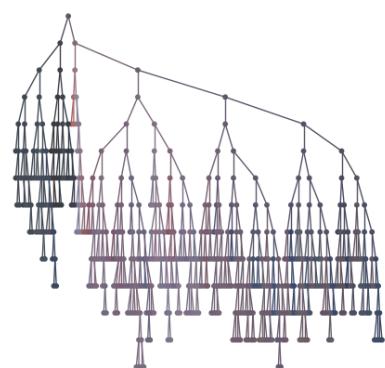
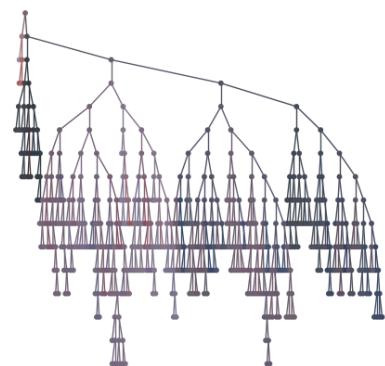
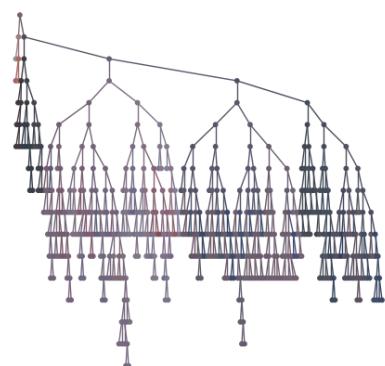
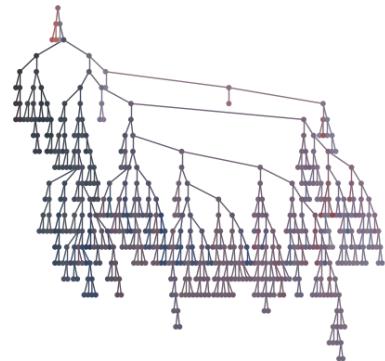
Rysunek 24: 10 dominujących kolorów wg klasteryzacji hierarchicznej, łączenie warda



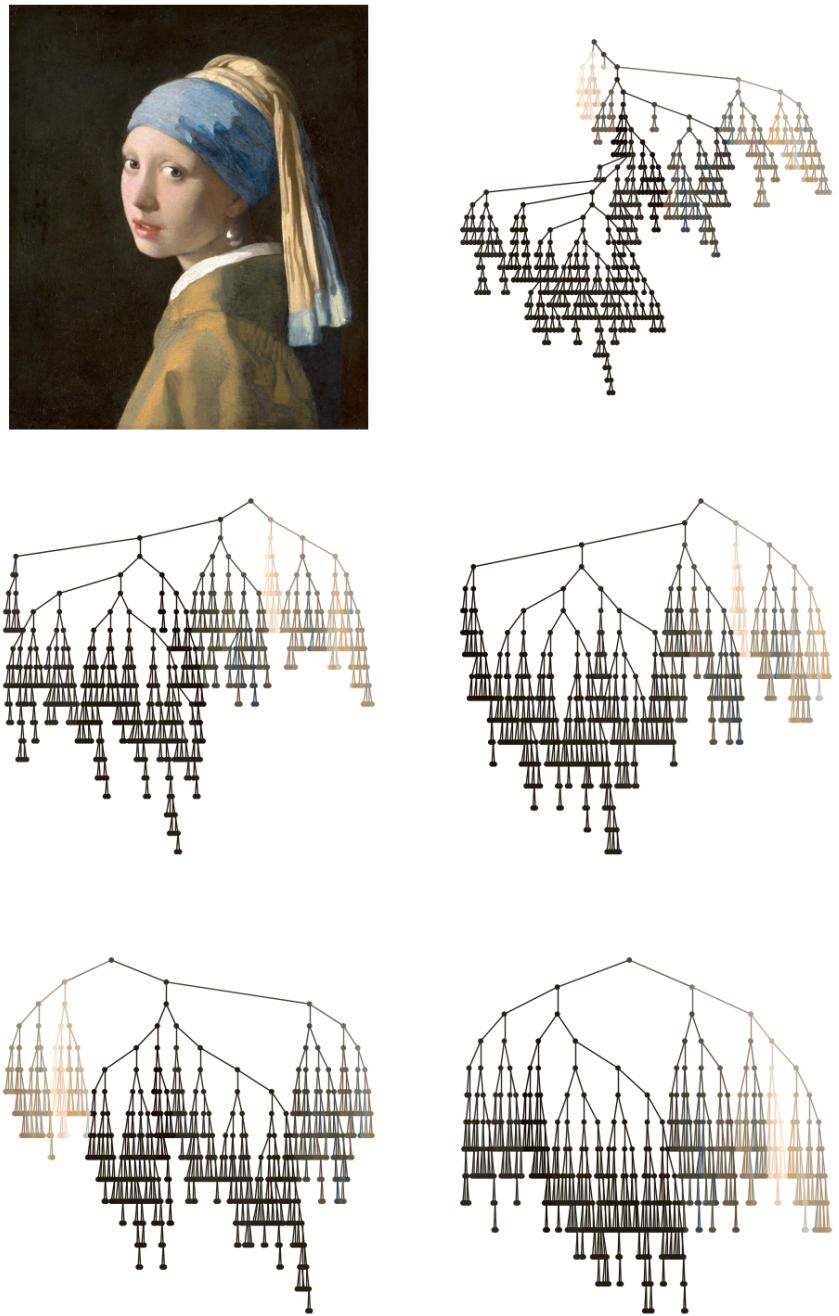
Rysunek 25: 10 dominujących kolorów wg klasteryzacji hierarchicznej, łączenie warda



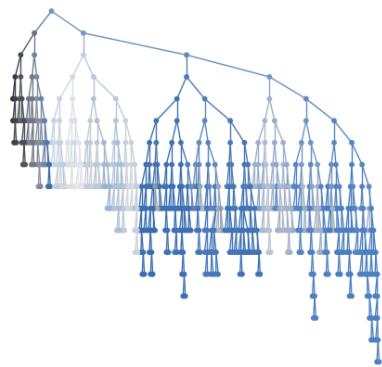
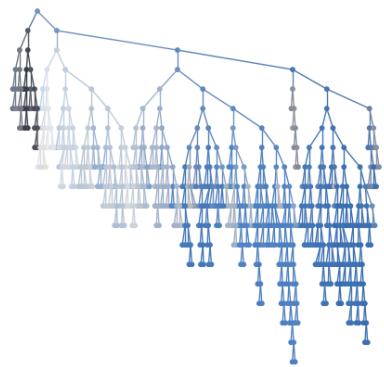
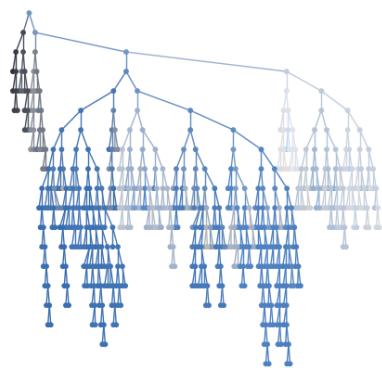
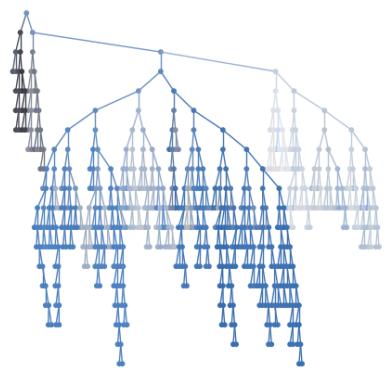
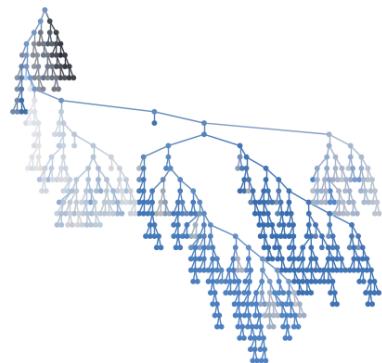
Rysunek 26: Drzewa hierarchiczne dla różnych metod łączenia: pojedyńcze, centroidalne, średnie, pełne, Warda



Rysunek 27: Drzewa hierarchiczne dla różnych metod łączenia: pojedyncze, centroidalne, średnie, pełne, Warda



Rysunek 28: Drzewa hierarchiczne dla różnych metod łączenia: pojedyńcze, centroidalne, średnie, pełne, Warda



Rysunek 29: Drzewa hierarchiczne dla różnych metod łączenia: pojedyńcze, centroidalne, średnie, pełne, Warda

7 Podsumowanie

Cieźko jest oceniać wyniki klasteryzacji - używana przeze mnie miara nie zawsze jest dobra. Wydaje mi się jednak, że krzywa błąd klasteryzacji-liczba klastrów może być użytecznym narzędziem do banania 'zgrupowania' danych. Ponad to, przyglądając się drzewom postaćm przy klasteryzacji hierarchicznej można dowiedzieć się czegoś o danych. Możnaby też wprowadzić metryki oparte na powstałyach drzewach - np.: średnia głębokość liścia, średnia szerokość drzewa, średnia wielkość poddrzewa, etc.. Myślę, że niektóre z nich mogłyby okazać się sensownymi pojęciami w analizie danych.

8 Możliwe dalsze kroki

Oto pomysły na dalszą analizę:

- Wykrycie kilku (np. 8) dominujących kolorów per obraz dla wielu obrazów danego artysty/artystów i zrobienie klasteryzacji na tuplach $8 \times 3 = 24$ wymiarowych (na zbiorach kolorów używanych w danym obrazie). Być może powiedziałoby to coś o stylu danego artysty lub o generalnych tendencjach estetycznych danej epoki.
- Zrobienie programu, który wykrywa dominujące kolory na tapcie i na tej podstawie dobiera kolory terminala i systemowego interfejsu.
- Opracowanie różnych metryk spójności opartych na drzewach hierarchicznych.