Restauracja “Pod Złotymi Łukami”

Dane

Dane, którymi zajmujemy się tutaj zajmujemy to obserwacje reprezentujące dania.

W ramach posprzątania danych usunąłem dane nieliczbowe i znormalizowałem je (MinMax 0-1). Usunąłem też kolumny, które nie wnosiły nic ponad to co już wiemy z pozostałych (czyli wykazują liniową korelację):

- 'Total Fat (% Daily Value)
- 'Saturated Fat (% Daily Value)
- 'Cholesterol (% Daily Value)
- 'Sodium (% Daily Value)
- 'Carbohydrates (% Daily Value)
- 'Dietary Fiber (% Daily Value)

```
n[2]: runfile('/Users/petroniuss/Studies/MachineLearning_1/lab4/main.py', wdir='/Users/petroniuss/Studies/MachineLearning_1/lab4')
```

	Category	Item	Serving Size	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated
0	Breakfast	Egg McMuffin	4.8 oz (136 g)	300	120	13	20	
1	Breakfast	Egg White Delight	4.8 oz (135 g)	250	70	8	12	
2	Breakfast	Sausage McMuffin	3.9 oz (111 g)	370	200	23	35	
3	Breakfast	Sausage McMuffin with Egg	5.7 oz (161 g)	450	250	28	43	
4	Breakfast	Sausage McMuffin with Egg Whites	5.7 oz (161 g)	400	210	23	35	

	Serving Size	Calories	Calories from Fat	Total Fat	Saturated Fat	Trans Fat	Cholesterol	Sodium	Carbohydrates	Dietary Fi
0	0.122581	0.159574	0.113208	0.110169	0.25	0	0.452174	0.208333	0.219858	0.571
1	0.122581	0.132979	0.0660377	0.0677966	0.15	0	0.0434783	0.213889	0.212766	0.571
2	0.0935484	0.196809	0.188679	0.194915	0.4	0	0.0782609	0.216667	0.205674	0.571
3	0.151613	0.239362	0.235849	0.237288	0.5	0	0.495652	0.238889	0.212766	0.571
4	0.151613	0.212766	0.198113	0.194915	0.4	0	0.0869565	0.244444	0.212766	0.571

```
59.09471492223694 1.2175467669974878
```

Na tak przygotowanych danych wykonywałem dalszą część zadania.

Szacowanie jak “dobra” jest klasteryzacja

Metryka

Wybrałem indeks Daveisa-Bouldina - który bierze pod uwagę stosunki rozrzucenia danych wewnątrz klastra do separacji pomiędzy klastrami (krótko: im bardziej ściśnięte dane w klastrze i im dalej od innych klastrów tym lepiej) - czyli preferuje kule o jak najniższym promieniu. Można sobie wyobrazić klastry gdzie niekoniecznie dane leżą wewnątrz kuli ale np wewnątrz pierścienia - ta metryka nie dałaby takiemu klastrowaniu dobrej oceny.

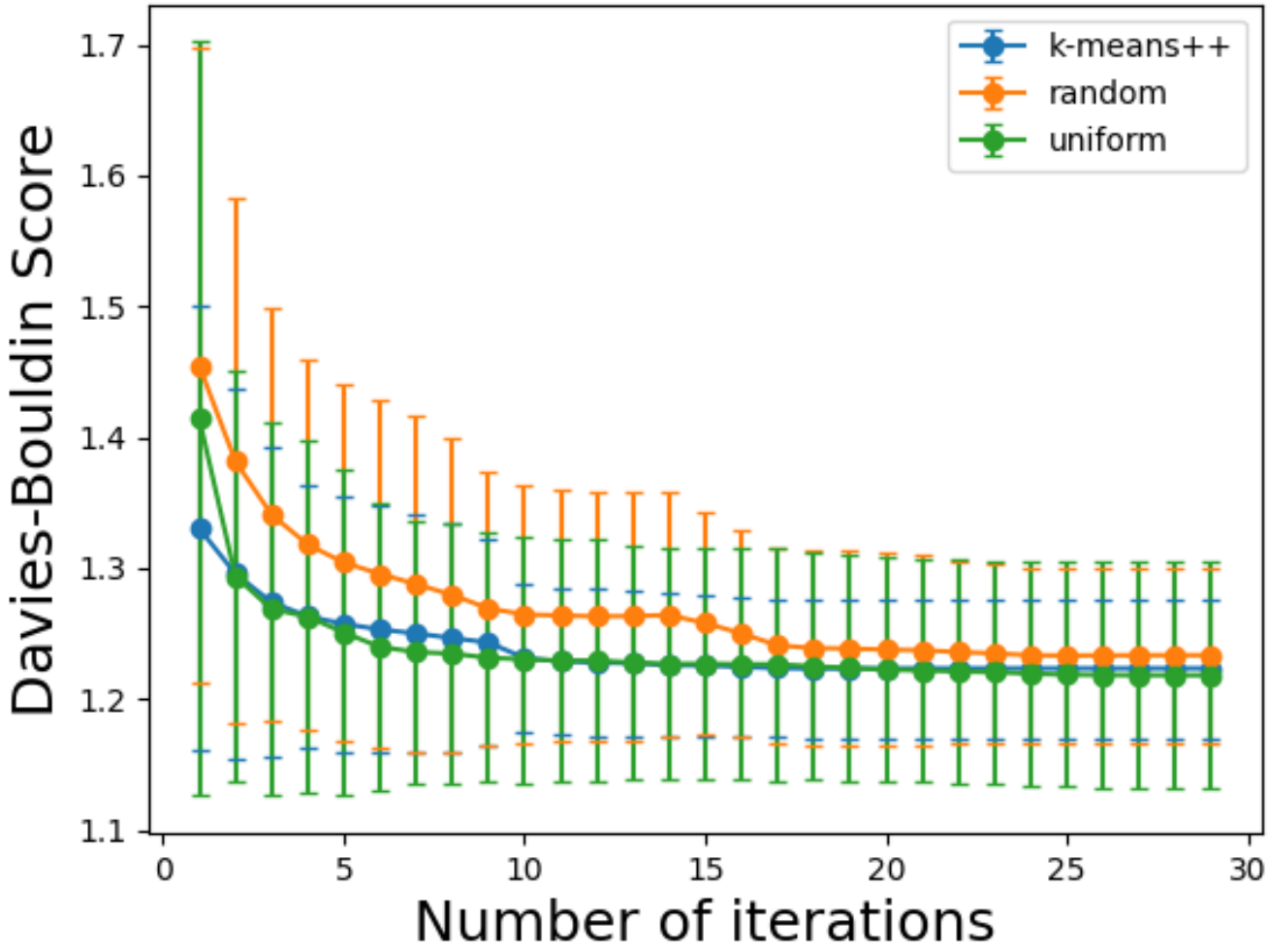
Im niższa wartość, tym lepsza jest klasteryzacja.

Obserwacja działania algorytmu dla kolejnych iteracji

Można zobaczyć jak inicjalizacja ma się do zbieżności, a także do osiąganego rozwiązania. Widać, że inicjalizacja k-means++ zazwyczaj zaczynała od lepszego rozwiązania (to podejście cechuje też najniższa wariancja) ale prowadziło do gorszego rozwiązania.

Najlepiej radziło sobie losowanie każdego parametru z rozkładem jednostajnym - daje to dużą wariancję ale jest to pożądane bo dajemy sobie większą szansę na znalezienie lepszego rozwiązania, wynika to ze stochastycznej natury algorytmu.

Wyniki mierzyłem dla k = 5 i powtarzałem 50-krotnie.



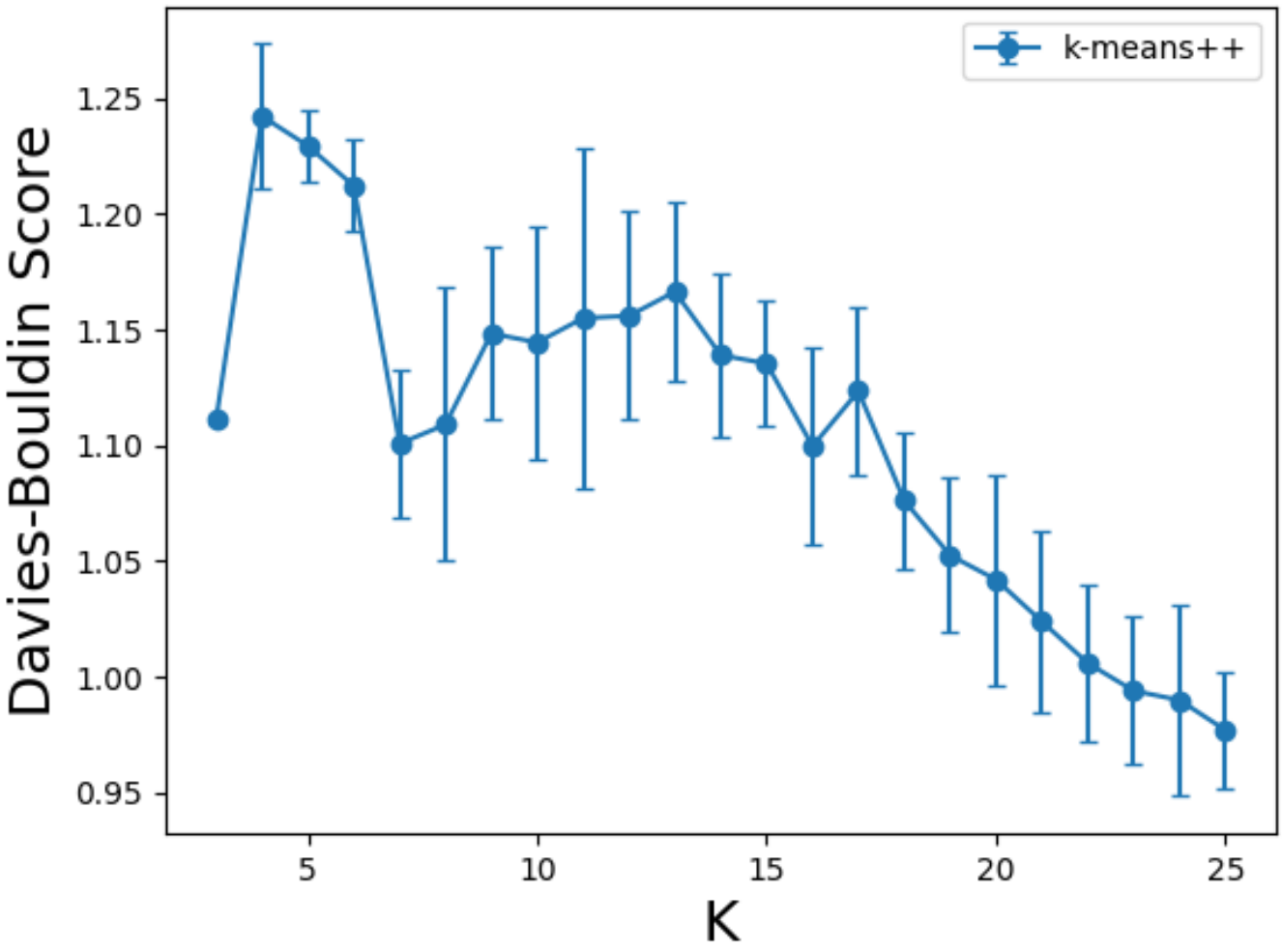
Implementacja

Początkowo myślałem, że ciężko to będzie zrobić bez kopiowania kawałka implementacji SKLearna ale po dłuższej analizie kodu źródłowego okazało się, że można po prostu za każdym razem podawać poprzednio uzyskane środki klastrów jeśli tylko ustawimy maksymalną ilość iteracji na 1.

```
while True:
    kmeans.fit(X)

    labels = kmeans.labels_
    centroids = kmeans.cluster_centers_

    if np.array_equal(state, labels):
        break

    kmeans.init = centroids
    state = labels

    inertia = kmeans.inertia_
    score = sklearn.metrics.davies_bouldin_score(X, labels)
    i += 1

    ys.append(score)
    print(i, inertia, score)
```

Wybranie najlepszego K

Wykres wydaje się być zastanawiający jeśli nie zdajemy sobie sprawy, że indeks najlepiej oceni klasteryzację gdzie każdy punkt jest osobnym klastrem (co uświadomiłem sobie na zajęciach). Tak więc opierając się na tym wykresie można by uznać, że najlepsza klasteryzacja jest dla k w przedziale 5-9. Najsensowniej byłoby zobaczyć czy inne metryki wykazują podobne tendencje.



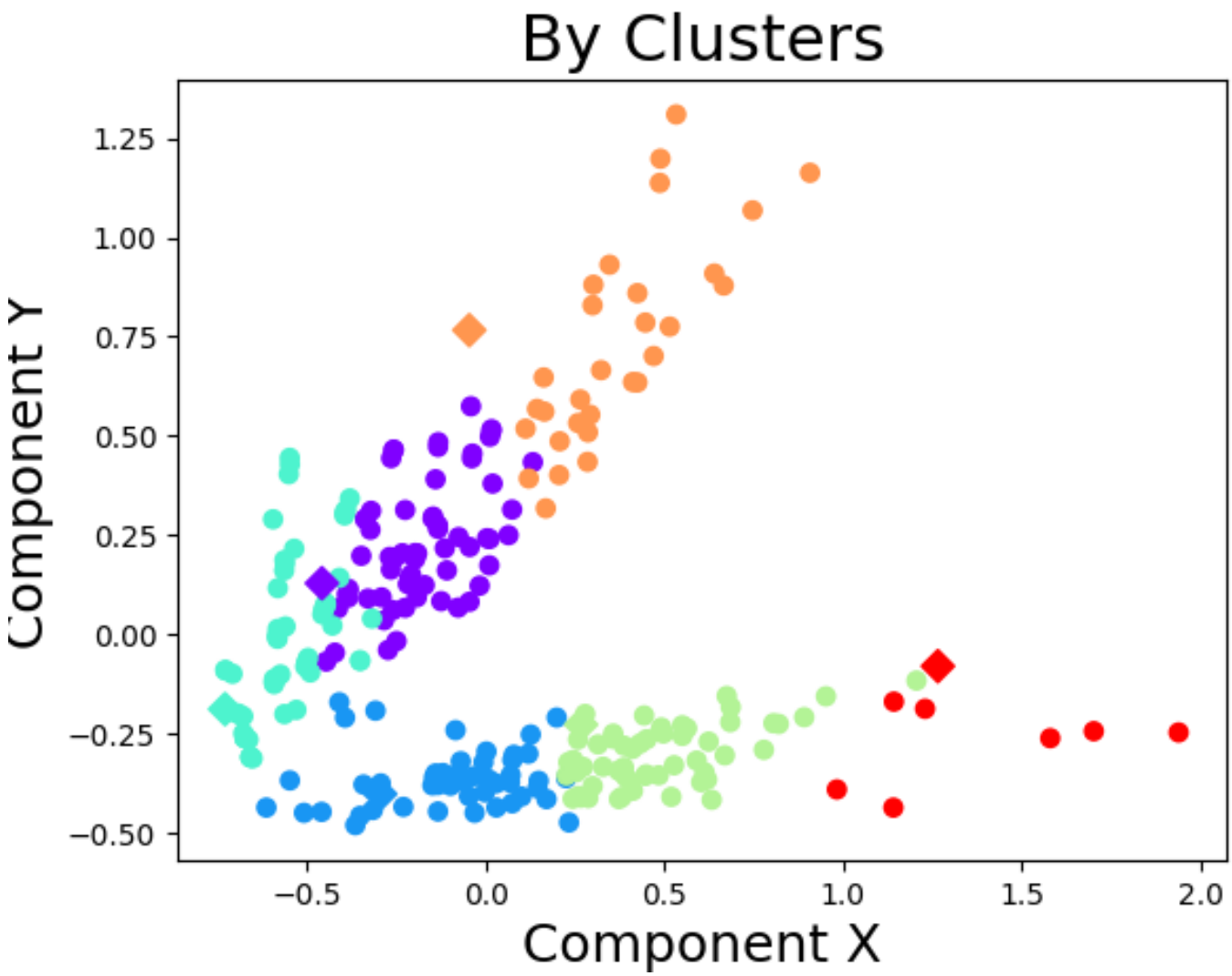
Wizualizacja rezultatów z wykorzystaniem PCA

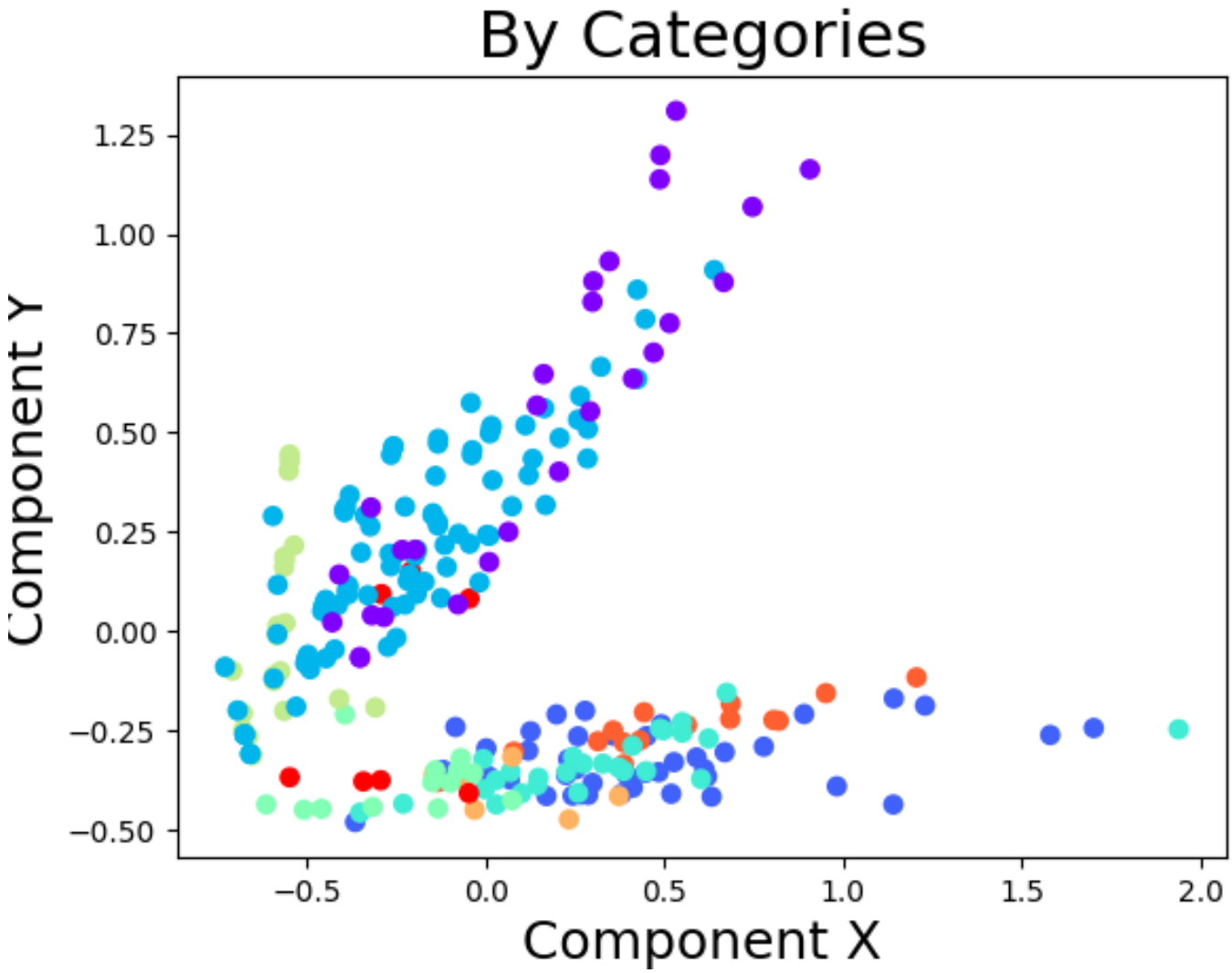
Sprawdziłem kilka k w wybranym wyżej przedziale (wszystkie wyglądały w miarę sesnownie)

Dane faktycznie są zgrupowane w klastry nawet po wykonaniu PCA. Widać, że grupowanie po kategoriach jest bardzo gruboziarniste i nie oddaje, tego jak dane (co do wartości) faktycznie się rozkładają.

Co do interpretacji to ciężko stwierdzić co zawiera każdy klaster. Elementry w klastrach są zgrupowane dość równomiernie:

- 65
- 47
- 56
- 55
- 30
- 7 - ten klaster wydaje się być znacznie mniejszy od pozostałych i zbierać odrzutki.


By Clusters
Component Y
Component X
1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.5
0.0
0.5
1.0
1.5
2.0



Wpływ przygotowania danych

Myślę, że przygotowanie danych miało bardzo duży wpływ na jakość klasteryzacji - na początku spróbowałem zobaczyć co wyjdzie bez usuwania kolumn i wykres z indeksem Daviesa-Bouldina wydawał się ciągle maleć (ale bez wyraźnego spadku w pewnym miejscu) - więc właściwie to nic nie mówił (oprócz tego, że coś jest źle :)).