PUM raport 4 - Klasteryzacja k-means

Piotr Zawiślan

8 Czerwiec 2022

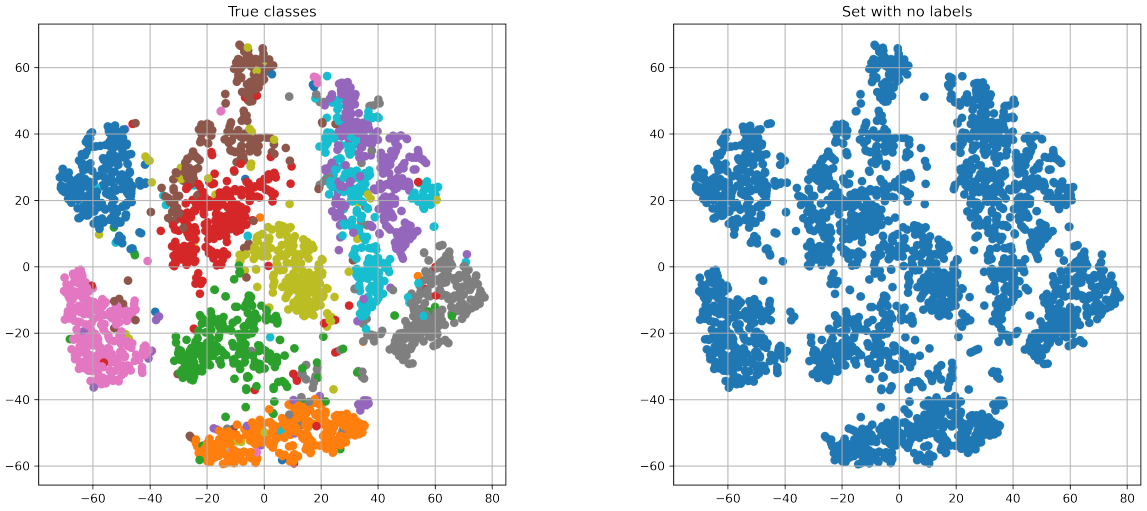
1 Wykorzystane technologie

Kod napisałem w języku Python. Użyłem modułów Numpy, Scipy, Scikit-Learn oraz Matplotlib.

2 Dane do pracy

2.1 Zbiór 2D

Zbiór dwuwymiarowy wygenerowałem podobnie jak w przykładzie w poleceniu, ale z drobną zmianą. Wykorzystałem podzbiór **3000** cyferek z MNIST. Użyłem *PCA*, aby zredukować wymiar do **120** cech. Następnie użyłem *TSNE*, aby zbiór zrzutować na 2D.
W wyniku powstały nie do końca równe i dające się odróżnić klastry, ale stwierdziłem że jest to dobry zbiór do eksperymentacji (możliwe że niesłusznie ;)).



Zbiór z oryginalnymi etykietami

Zbiór bez etykiet

Rysunek 1: Otrzymany zbiór 2D.

2.2 Zbiór FIFA

Załączony zbiór z danymi zawodników z gry FIFA 22 odpowiednio przygotowałem i oczyściłem z niepotrzebnych cech (linków itp.). Usunąłem także wszystkie wiersze, w których brakowało którejś z cech. W ten sposób uzyskałem zbiór danych z **17054** elementami i **76** cechami.

3 Narzędzia diagnostyczne

Sposoby oceny klasteryzacji, które wybrałem to:

- *Calinski-Harabasz index* (im większy tym lepiej)
- *Davies-Bouldin Index* (im mniejszy tym lepiej)

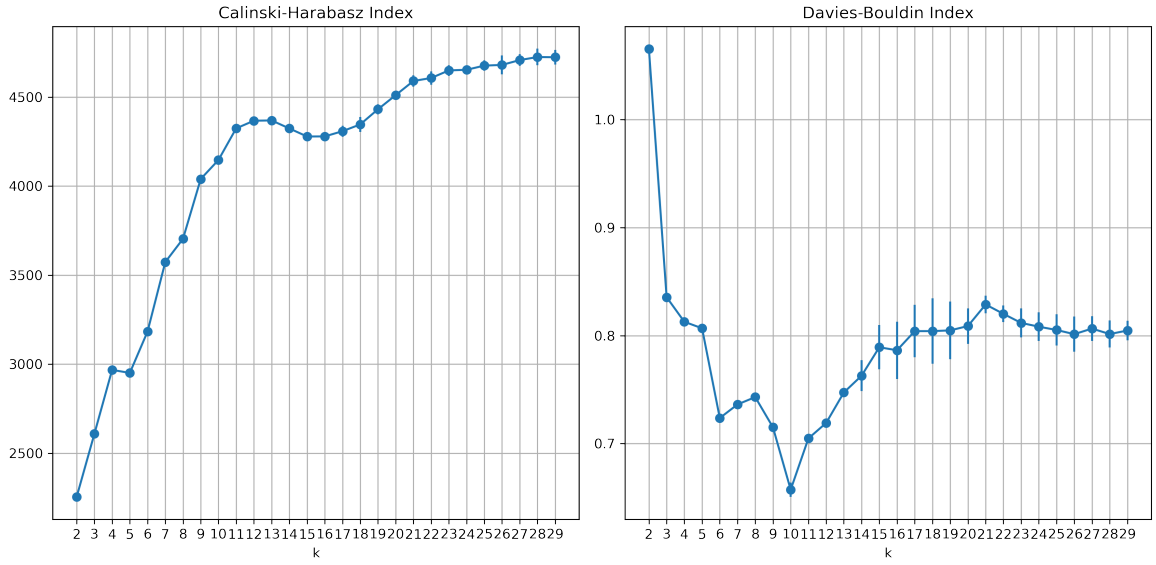
Przyjemniejszy w użyciu był ten drugi, ponieważ *Calinski-Harabasz index* miał tendencje do coraz lepszych wyników wraz z wrostem k.

4 Ustalanie właściwego poziomu rozdrobnienia

W tej sekcji zamieszczam wizualizacje działania klasteryzacji k-means dla zbioru 2D.

4.1 Podejście 'łokciowe'

Wyniki podejścia 'łokciowego' na zbiorze 2D prezentują się następująco:

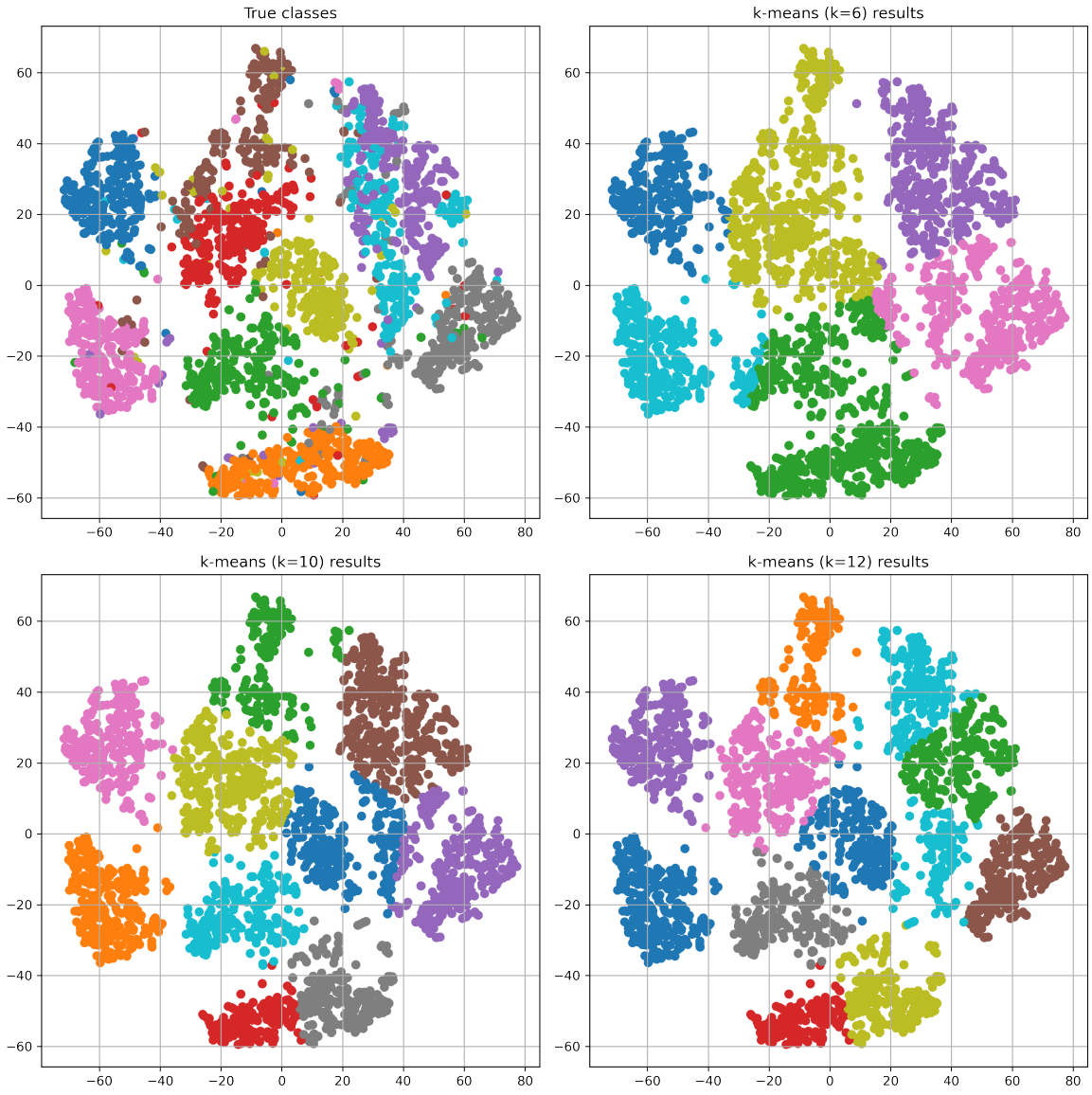


Rysunek 2: Wykresy wartości indeksów w zależności od k

- Na lewym wykresie widać 'górkę' w okolicach $k \in [10, 13]$. Potem wykres maleje, a następnie znów rośnie.
- Na prawym natomiast minimum w punkcie $k = 10$ jest od razu widoczne. W punkcie $k = 6$ też daje się zauważyć pewne 'minimum lokalne'.

Na prawym wykresie rzeczywista liczba klastrów została poprawnie odgadnięta, na lewym - prawie.

Na poniższym obrazku zawarte są wyniki działania klasteryzacji dla $k \in \{6, 10, 12\}$ oraz rzeczywiste klasy (cyferki).



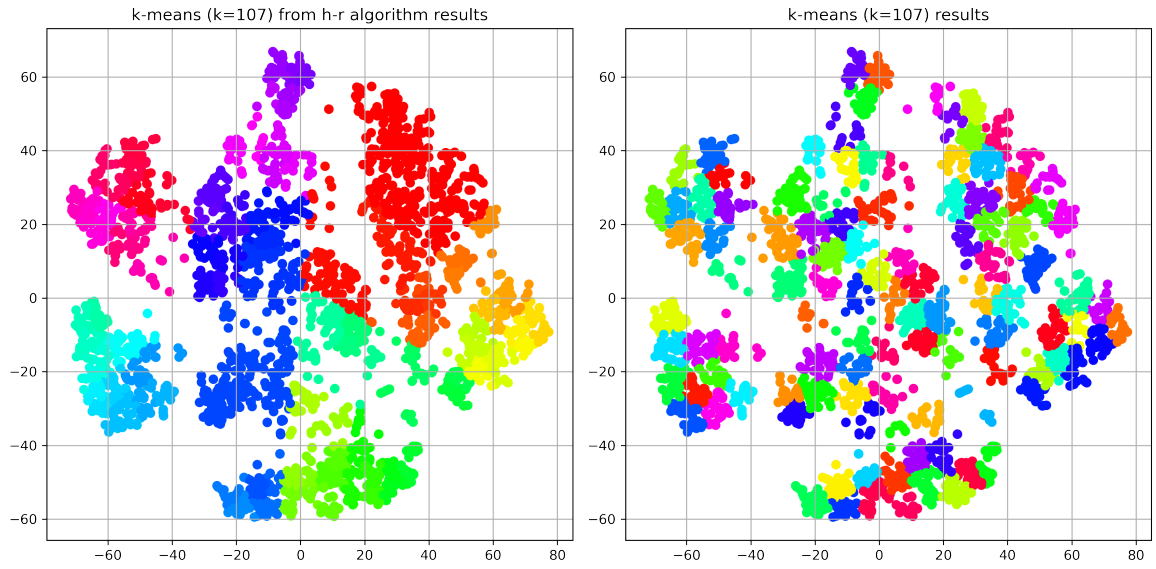
Rysunek 3: Wyniki klasteryzacji k-means dla $k \in \{6, 10, 12\}$ oraz rzeczywiste klasy.

Dla $k = 10$ więszkość klastrów wygląda niezwykle zgrabnie. Wyjątkiem jest podzbiór punktów na samym dole podzielony na dwa klastry (czerwony i szary) oraz ciemnoniebieski klaster, który 'zabiera' części klastrów brązowego oraz fioletowego. Ciekawą alternatywą dla $k = 10$ jest podział $k = 12$, w którym nie można się do wielu rzeczy przyczepić. Podział na $k = 6$ klastrów nie do końca pasuje do naszego zbioru.

4.2 Podejście 'hierarchiczno-rekurencyjne'

W algorytmie posłużyłem się *testem Shapiro-Wilka*. Niestety przy progu akceptacji p-wartości mierzącym 10^{-9} oraz maksymalnej głębokości rekurencji równej 8 algorytm sprawował się dość miernie i dla zbioru 2D zwracał 'optymalne' $k = 107$.

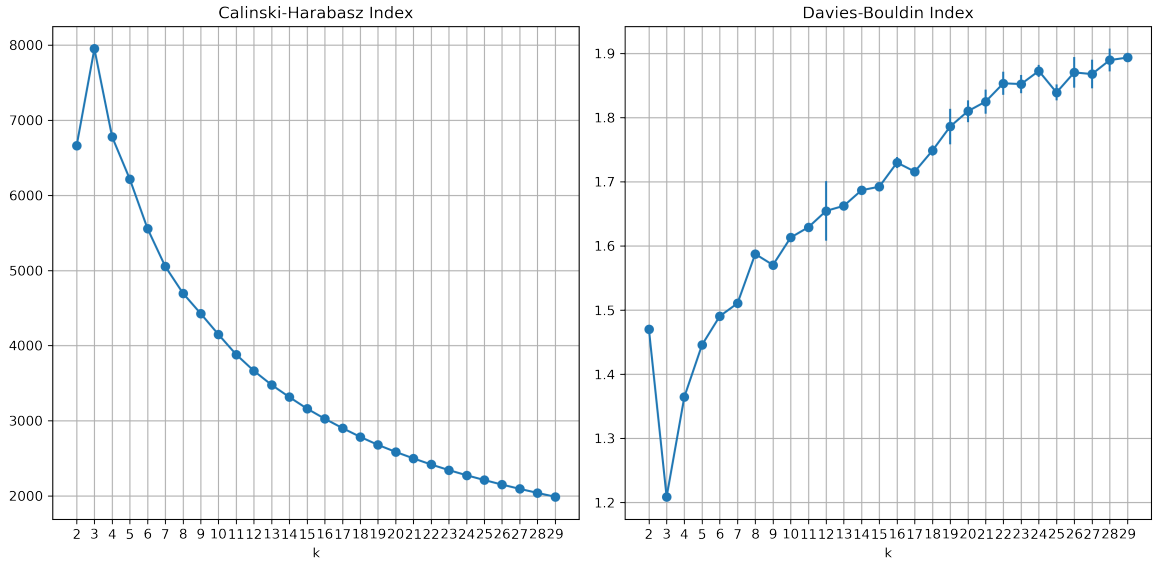
Poniżej wizualizacja wynikowych etykiet (na lewym obrazku trochę słabo widać poszczególne etykiety, ale za to widać jak algorytm dzielił zbiór na podklastry).



Rysunek 4: Wyniki klasteryzacji k-means dla $k = 107$ (bezpośrednio zwrócone przez algorytm, oraz 'na czysto'.

5 Testowanie metod na zbiorze FIFA

5.1 Podjeście łokciowe



Rysunek 5: Wykresy wartości indeksów w zależności od k dla zbioru FIFA.

- Interpretując powyższy wykres jedynymi sensownymi k wydają się 3 oraz 8.
- Wykonując klasteryzację dla $k = 3$, 7 z 10 najlepszych piłkarzy (wg cechy 'overall') znajduje się razem w klastrze, a dla $k = 8$ - 5 z 10.

- Podział wydaje się mieć sens, ale ciężko mi porównać klastry między sobą (wrzucenie Messiego, Lewandowskiego oraz Ronaldo do jednego klastra wydaje mi się sensowne, ale na tym kończy się moja znajomość zawodników piłki nożnej :)).

5.2 Podejście 'hierarchiczno-rekurencyjne'

W drugim sposobie próbowałem różnych wartości parametrów `max_depth` oraz `p_value_threshold`, ale nie znalazłem takiej ich kombinacji, aby wynik nie był równy $k = 2^{max\_depth}$.

- $k = 64$
- Tym razem też istnieje jeden klaster zawierający 7 z 10 najlepszych piłkarzy (wg cechy 'overall'). Ogólnie piłkarze w jednym klastrze mieli bardzo podobną cechę 'overall' (np. 86 ± 3).
- Wątpie, że wybór $k = 64$ ma sens, ponieważ algorytm ani razu nie znalazł rozkładu choć trochę przypominającego rozkład normalny. Jednakże sam podział, znając algorytm kmeans, musi mieć chociaż krztynę sensu :).

6 Wnioski

Podsumowując, metoda 'łokciowa' z wizualizacją mi bardziej przypadła do gustu, ponieważ nie jest ona oparta na wykrywaniu rozkładów normalnych.
Druga metoda jest bardzo ciekawa, ale niestety nie dała ona dobrych wyników. Możliwe, że dla bardziej 'normalnych' klastrów by ona zadziałała, ale z drugiej strony wydaje mi się, że można po prostu użyć metody, w której nie trzeba podawać parametru k.
Co do wyboru zbioru 2D - możliwe, że jest nieco zbyt zaszumiony, a klastry są zbyt mało 'koliste', aby zastosować k-means.