# SPRAWOZDANIE 6

## Podstawy Sztucznej Inteligencji

## Klasteryzacja K-Means

## Program WEKA 6

Natalia Gadocha 304165
Geoinformatyka III rok

Do poniższej analizy będziemy używać danych iris2D.arff. Przetestujemy na nich bowiem działanie algorytmu K-means. Wypróbujemy kolejno zbiór danych uczących oraz jego podział procentowy. Wspomniane wyniki prezentują się następująco:

- Klaster use training set

k = 2

- Within cluster sum of squared errors: 55.1796875099748

- Cluster 0: 4.7,1.4,Iris-versicolor
  Cluster 1: 4.3,1.3,Iris-versicolor

| Attribute | Full Data (150.0) | 0 (100.0) | 1 (50.0) |
|---|---|---|---|
| petallength | 3.7587 | 4.906 | 1.464 |
| petalwidth | 1.1987 | 1.676 | 0.244 |
| class | Iris-setosa | Iris-versicolor | Iris-setosa |

Clustered Instances
0 100 ( 67%)
1 50 ( 33%)

## k = 3

- Within cluster sum of squared errors: 1.8541119849021674

- Cluster 0: 4.7,1.4,Iris-versicolor
  Cluster 1: 4.3,1.3,Iris-versicolor
  Cluster 2: 5.1,2.3,Iris-virginica

| Attribute | Full Data | 0 | 1 | 2 |
|---|---|---|---|---|
| | (150.0) | (50.0) | (50.0) | (50.0) |
| petallength | 3.7587 | 4.26 | 1.464 | 5.552 |
| petalwidth | 1.1987 | 1.326 | 0.244 | 2.026 |
| class | Iris-setosa | Iris-versicolor | Iris-setosa | Iris-virginica |

0 50 ( 33%)
1 50 ( 33%)
2 50 ( 33%)

## k = 4

- Within cluster sum of squared errors: 1.4745584462478853

- Cluster 0: 4.7,1.4,Iris-versicolor
  Cluster 1: 4.3,1.3,Iris-versicolor
  Cluster 2: 5.1,2.3,Iris-virginica
  Cluster 3: 1.4,0.2,Iris-setosa

| Attribute | Full Data | 0 | 1 | 2 | 3 |
|---|---|---|---|---|---|
| | (150.0) | (25.0) | (25.0) | (50.0) | (50.0) |
| petallength | 3.7587 | 4.628 | 3.892 | 5.552 | 1.464 |
| petalwidth | 1.1987 | 1.472 | 1.18 | 2.026 | 0.244 |
| class | Iris-setosa | Iris-versicolor | Iris-versicolor | Iris-virginica | Iris-setosa |

Clustered Instances
0 25 ( 17%)
1 25 ( 17%)
2 50 ( 33%)
3 50 ( 33%)

## k = 5

- Within cluster sum of squared errors: 1.3590412110407115

- Cluster 0: 4.7,1.4,Iris-versicolor
  Cluster 1: 4.3,1.3,Iris-versicolor
  Cluster 2: 5.1,2.3,Iris-virginica
  Cluster 3: 1.4,0.2,Iris-setosa
  Cluster 4: 4.9,1.5,Iris-versicolor

| Attribute | Full Data (150.0) | 0 (20.0) | 1 (12.0) | 2 (50.0) | 3 (50.0) | 4 (18.0) |
|---|---|---|---|---|---|---|
| petallength | 3.7587 | 4.245 | 3.6333 | 5.552 | 1.464 | 4.6944 |
| petalwidth | 1.1987 | 1.305 | 1.0667 | 2.026 | 0.244 | 1.5222 |
| class | Iris-setosa | Iris-versicolor | Iris-versicolor | Iris-virginica | Iris-setosa | Iris-versicolor |

Clustered Instances

| | |
|---|---|
| 0 | 20 ( 13%) |
| 1 | 12 ( 8%) |
| 2 | 50 ( 33%) |
| 3 | 50 ( 33%) |
| 4 | 18 ( 12%) |

## k = 6

- Within cluster sum of squared errors: 1.360803439407315

- Cluster 0: 4.7,1.4,Iris-versicolor
  Cluster 1: 4.3,1.3,Iris-versicolor
  Cluster 2: 5.1,2.3,Iris-virginica
  Cluster 3: 1.4,0.2,Iris-setosa
  Cluster 4: 4.9,1.5,Iris-versicolor
  Cluster 5: 4.6,1.4,Iris-versicolor

| Attribute | Full Data (150.0) | 0 (12.0) | 1 (13.0) | 2 (50.0) | 3 (50.0) | 4 (9.0) | 5 (16.0) |
|---|---|---|---|---|---|---|---|
| petallength | 3.7587 | 4.5167 | 3.6615 | 5.552 | 1.464 | 4.8 | 4.25 |
| petalwidth | 1.1987 | 1.3083 | 1.0769 | 2.026 | 0.244 | 1.5889 | 1.3938 |
| class | Iris-setosa | Iris-versicolor | Iris-versicolor | Iris-virginica | Iris-setosa | Iris-versicolor | Iris-versicolor |

0 12 ( 8%)
1 13 ( 9%)
2 50 ( 33%)
3 50 ( 33%)
4 9 ( 6%)
5 16 ( 11%)

Największy błąd i najgorsze wyniki otrzymujemy dla k równego 2. Wystarczająco dobry rezultat otrzymujemy natomiast dla k = 3. Wraz ze wzrostem kolejnej liczby klastrów (po 5) zaczyna rosnąć również błąd dopasowania. Możemy również dostrzec, że wraz ze wzrostem ilości klastrów wzrasta liczba iteracji

## ▸ Percentage split

k = 2

- Within cluster sum of squared errors: 37.7060494355294
- Cluster 0: 4,1,Iris-versicolor
  Cluster 1: 4.1,1.3,Iris-versicolor

| Attribute | Full Data (105.0) | 0 (36.0) | 1 (69.0) |
|---|---|---|---|
| petallength | 3.7333 | 1.4694 | 4.9145 |
| petalwidth | 1.1819 | 0.2556 | 1.6652 |
| class | Iris-setosa | Iris-setosa | Iris-virginica |

Clustered Instances

0 14 ( 31%)
1 31 ( 69%)

## k = 3

- Within cluster sum of squared errors: 1.3730524089909297

- Cluster 0: 4,1,Iris-versicolor
  Cluster 1: 4.1,1.3,Iris-versicolor
  Cluster 2: 1.4,0.3,Iris-setosa

| Attribute | Full Data | 0 | 1 | 2 |
|---|---|---|---|---|
| | (105.0) | (34.0) | (35.0) | (36.0) |
| petallength | 3.7333 | 4.2118 | 5.5971 | 1.4694 |
| petalwidth | 1.1819 | 1.3206 | 2 | 0.2556 |
| class | Iris-setosa | Iris-versicolor | Iris-virginica | Iris-setosa |

Clustered Instances

0 16 ( 36%)
1 15 ( 33%)
2 14 ( 31%)

## k = 4

- Within cluster sum of squared errors: 1.0790871123749346

- Cluster 0: 4,1,Iris-versicolor
  Cluster 1: 4.1,1.3,Iris-versicolor
  Cluster 2: 1.4,0.3,Iris-setosa
  Cluster 3: 4.8,1.8,Iris-virginica

| Attribute | Full Data | 0 | 1 | 2 | 3 |
|---|---|---|---|---|---|
| | (105.0) | (11.0) | (23.0) | (36.0) | (35.0) |
| petallength | 3.7333 | 3.6545 | 4.4783 | 1.4694 | 5.5971 |
| petalwidth | 1.1819 | 1.0909 | 1.4304 | 0.2556 | 2 |
| class | Iris-setosa | Iris-versicolor | Iris-versicolor | Iris-setosa | Iris-virginica |

Clustered Instances

0 3 ( 7%)
1 13 ( 29%)
2 14 ( 31%)
3 15 ( 33%)

## k = 5

- Within cluster sum of squared errors: 1.0092865057723208

- Cluster 0: 4,1,Iris-versicolor
  Cluster 1: 4.1,1.3,Iris-versicolor
  Cluster 2: 1.4,0.3,Iris-setosa
  Cluster 3: 4.8,1.8,Iris-virginica
  Cluster 4: 3.5,1,Iris-versicolor

| Attribute | Full Data (105.0) | 0 (13.0) | 1 (16.0) | 2 (36.0) | 3 (35.0) | 4 (5.0) |
|---|---|---|---|---|---|---|
| petallength | 3.7333 | 3.9923 | 4.6438 | 1.4694 | 5.5971 | 3.4 |
| petalwidth | 1.1819 | 1.2385 | 1.4812 | 0.2556 | 2 | 1.02 |
| class | Iris-setosa | Iris-versicolor | Iris-versicolor | Iris-setosa | Iris-virginica | Iris-versicolor |

Clustered Instances

| | |
|---|---|
| 0 | 7 ( 16%) |
| 1 | 8 ( 18%) |
| 2 | 14 ( 31%) |
| 3 | 15 ( 33%) |
| 4 | 1 ( 2%) |

## k = 6

- Within cluster sum of squared errors: 0.6022871581064135

- Cluster 0: 4,1,Iris-versicolor
  Cluster 1: 4.1,1.3,Iris-versicolor
  Cluster 2: 1.4,0.3,Iris-setosa
  Cluster 3: 4.8,1.8,Iris-virginica
  Cluster 4: 3.5,1,Iris-versicolor
  Cluster 5: 5.1,1.9,Iris-virginica

| Attribute | Full Data (105.0) | 0 (13.0) | 1 (16.0) | 2 (36.0) | 3 (18.0) | 4 (5.0) | 5 (17.0) |
|---|---|---|---|---|---|---|---|
| petallength | 3.7333 | 3.9923 | 4.6437 | 1.4694 | 5.2222 | 3.4 | 5.9941 |
| petalwidth | 1.1819 | 1.2385 | 1.4812 | 0.2556 | 1.8 | 1.02 | 2.2118 |

class Iris-setosa Iris-versicolor Iris-versicolor Iris-setosa Iris-virginica Iris-versicolor Iris-virginica

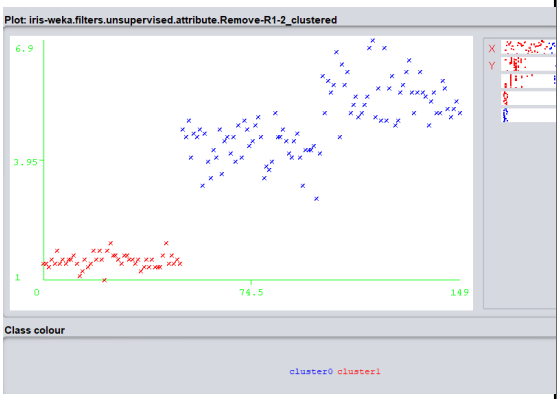
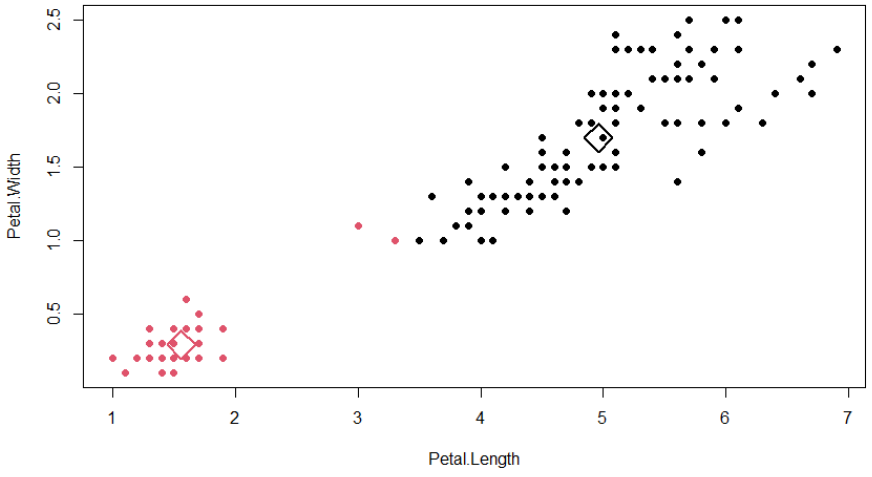
Clustered Instances
0 7 ( 16%)
1 8 ( 18%)
2 14 ( 31%)
3 5 ( 11%)
4 1 ( 2%)
5 10 ( 22%)

Po wykorzystaniu opcji Percentage split możemy dostrzec, iż wszelkie błędy wynikające z przypasowań są mniejsze w porównaniu do Use training set. Powyższe przyporządkowania mocno się również różną od poprzednich.

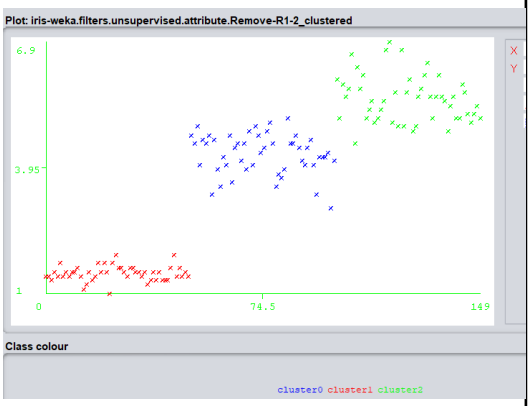
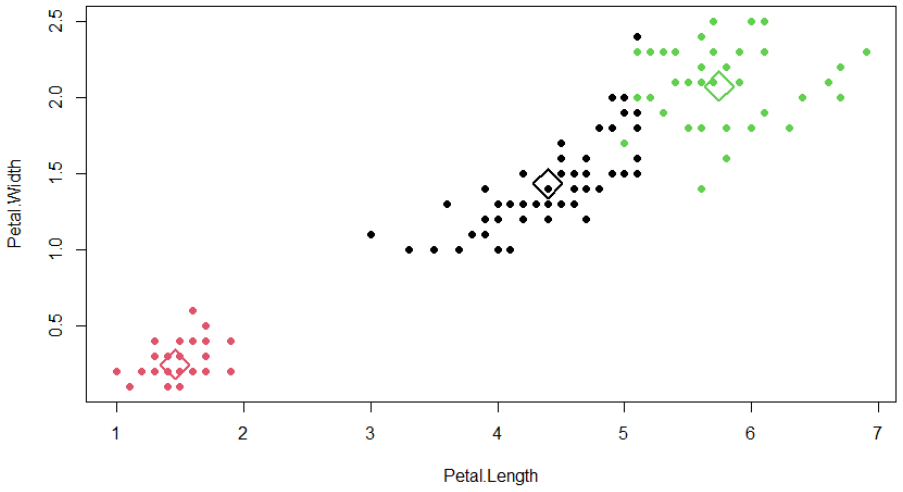
## Przedstawienie w formie graficznej

Wyżej omówione wyniki przedstawimy jeszcze w formie graficznej. Zostanie wykorzystane do tego środowisko R. Wszystkie dane zostaną umieszczone na wykresach wraz z odpowiednimi dopasowanymi do nich centroidami.

k = 2

| WEKA | R |
| --- | --- |
|  |  |

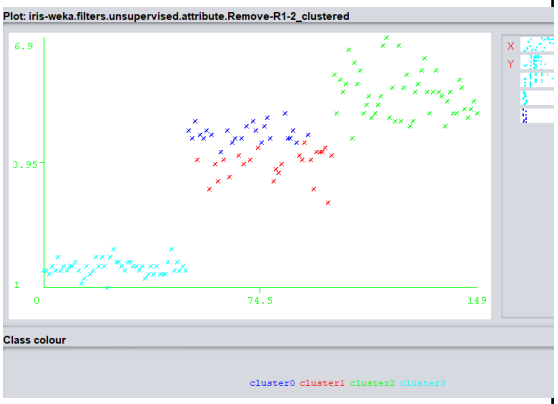
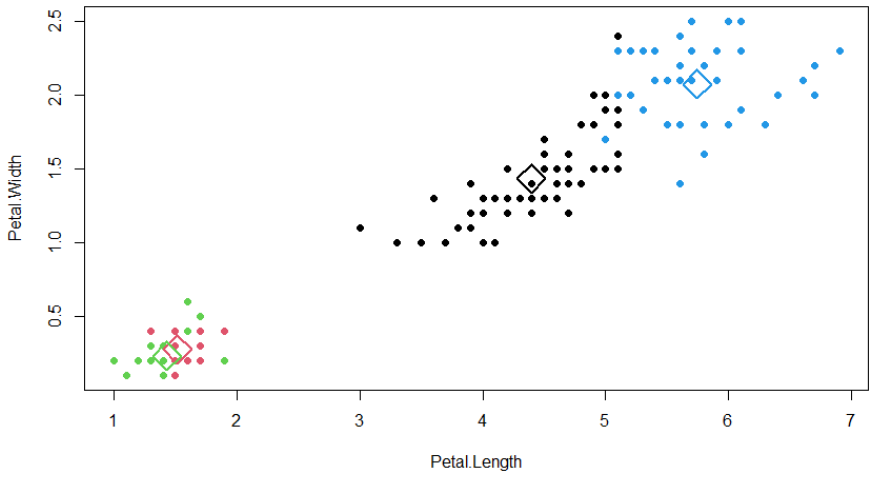
## k = 3

| WEKA | R |
| --- | --- |
|  |  |

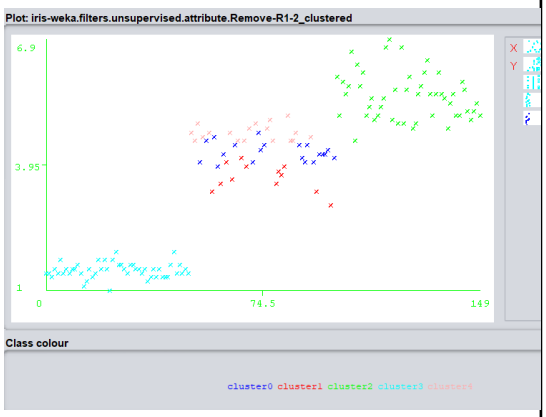
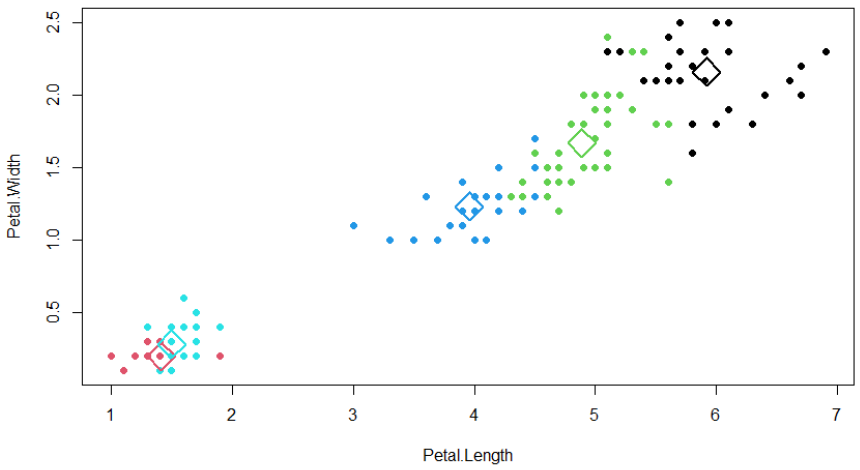




## k = 4

| WEKA | R |
| --- | --- |
|  |  |

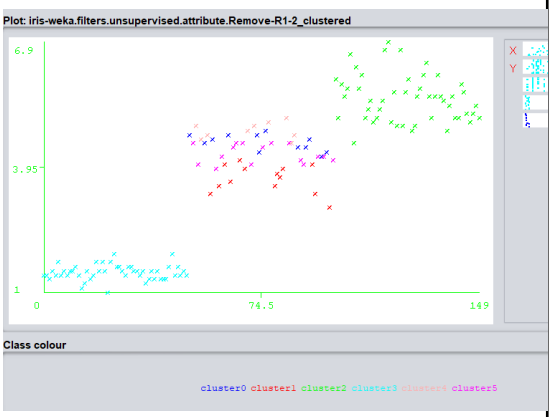
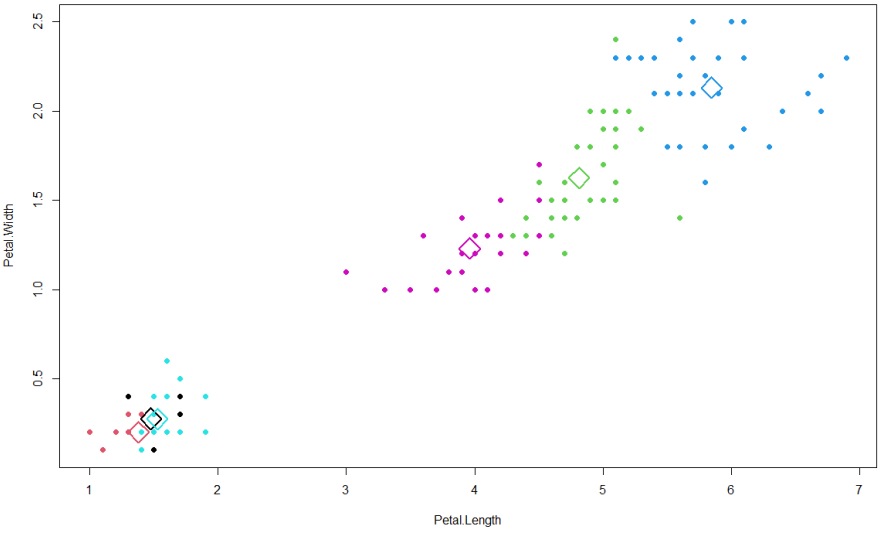
## k = 5

| WEKA | R |
| --- | --- |
|  |  |

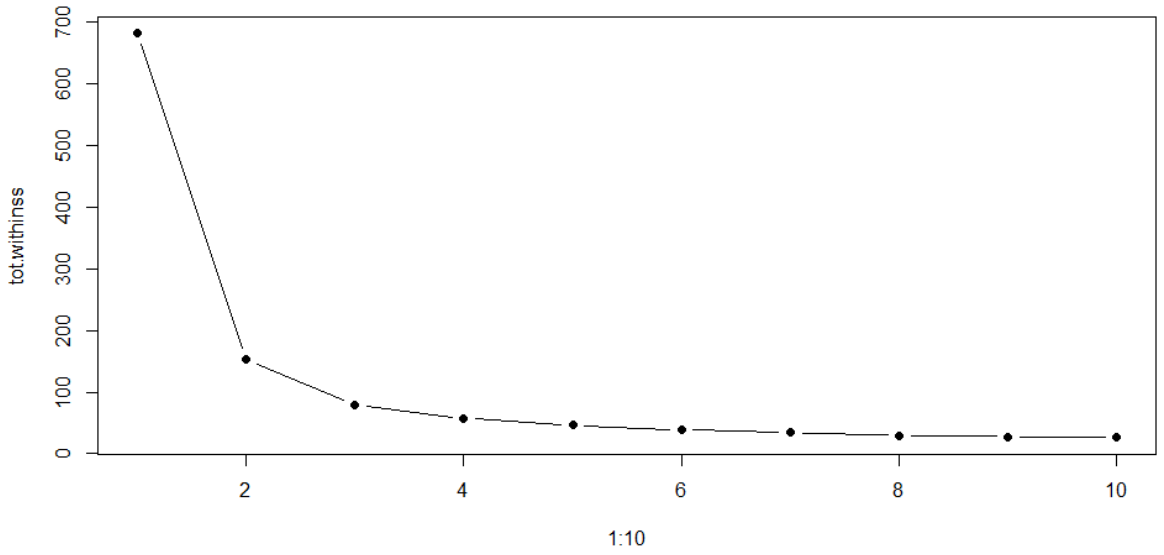




## k = 6.

| WEKA | R |
| --- | --- |
|  |  |

Środowisko R oferuje dodatkowo wyrysowanie wykresu, który przedstawia jak w zależności od liczby klastrów będzie zmieniać się dopasowanie. W naszym przypadku zauważamy, iż już po trzech klastrach wartości przestają gwałtownie się zmieniać. Jest to potwierdzeniem na to, że nasz zbiór możemy podzielić na trzy klastry, żeby wynik analiz był satysfakcjonujący.




## Próba optymalizacji algorytmu

### Co wiemy o algorytmie?

Algorytm K-means ma niską złożoność, przez co jest wydajny w działaniu. Najlepiej sobie radzi przy dużych zbiorach i niskich ilościach grup. Jest on skalowalny oraz tworzy zwarte skupiska - działa więc lepiej dla dobrze ukształtowanych podziałów zbioru.
Jego minusem natomiast jest brak identyfikacji wartości odstających i szumu - działa dobrze głównie dla sferycznych skupisk i jednorodnej gęstości. Nie jest brana pod uwagę również rzeczywista struktura zbiory, tylko z góry przypisana liczba klastrów. Natomiast biorąc różne wartości początkowe otrzymamy różne wyniki.

### Jak więc go ulepszyć?

Skuteczną metodą jest zmiana sposobu znajdowania odległości poprzez zmianę distanceFunction. Znaczącym ulepszeniem była również zmiana ilości klastrów. W naszym przypadku była to liczba 3. Skuteczną zmianą było także wykorzystanie warzonej miary odległości uwzględniającej znaczenie atrybutów. Opcja startowania algorytmu z losowo wybieranych centroidów daje najlepsze wyniki - unika bowiem trafiania w lokalne minimum. Możemy też zmniejszać liczbę potrzebnych iteracji - do nawet trzech.

We wspominanym algorytmie dążymy do tego, aby móc wybrać wynik z najmniejszą odległością kwadratową. Chcemy, żeby klastry były podzielone na podobne rozmiary. Wypróbować należy różne wybory liczby klastrów k oraz ziarna RNG.
Nieistotne atrybuty mogą powodować zmniejszanie optymalności algorytmu. Można spróbować usunąć potencjalnie nieistotne atrybuty przed analizą klastrów.

## Obliczenie współczynnika CH

Indeks Calińskiego - Harabasza jest wykorzystywany do obliczania estymacji liczby klastrów. Jego wzór prezentuje się następująco:

$$CH(\xi_K) = \frac{\mathrm{tr}\left(\boldsymbol{B}(\xi_K)\right)/(K-1)}{\mathrm{tr}\left(\boldsymbol{W}(\xi_K)\right)/(N-K)}$$

Możemy go również wykorzystać w programie WEKA. Użyjemy do tego bowiem metody CascadeSimpleKMeans, która to pozwala wybrać najlepsze k według kryterium indeksu CH. Nasze wyniki prezentują się następująco:

k = 2

- cascade> mean CH: [ 330,47 ] cascade> k (yields highest mean CH): 2
- cascade> seed (highest CH for k=2) : -1155869325
- Clustered Instances

  0 100 ( 67%)
  1 50 ( 33%)

## k = 3

- cascade> mean CH: [ 330,47 4530,98 4061,18 ] cascade> k (yields highest mean CH): 3
- cascade> seed (highest CH for k=3) : -1465154083
- Clustered Instances

  0 50 ( 33%)
  1 50 ( 33%)
  2 50 ( 33%)

## k = 4

- cascade> mean CH: [ 3838,57 ]cascade> k (yields highest mean CH): 4
- cascade> seed (highest CH for k=4) : -138487339
- Clustered Instances

  0 50 ( 33%)
  1 25 ( 17%)
  2 25 ( 17%)
  3 50 ( 33%)

Wyniki jakie otrzymaliśmy są podobne do pierwszej zastosowanej metody. Ustawiając natomiast parametry min = 2, max = 4 najlepszym przypasowaniem ponownie okazuje się być k = 3.