

# SPRAWOZDANIE

Zajęcia: Zbiory Big Data i Eksploracja Danych

Prowadząca: dr inż. Ruslana Ziubina

Laboratorium nr 6 Data rozpoczęcia: 12.01.2024 Temat: Klasyfikacja i grupowanie	Rafał Klinowski Informatyka II stopień, stacjonarne, Semestr 2, gr. a
---	--

Poszczególne ćwiczenia będą wykonywane w pliku źródłowym edytowanym przy pomocy środowiska RStudio, opisanego w poprzednich laboratoriach.

## Ćw. 1.

Celem było utworzenie klasyfikatora opartego o algorytm kNN dla danych dotyczących raka.

```
> str(wbcd)
'data.frame': 569 obs. of 32 variables:
 $ id          : int  87139402 8910251 905520 868871 9012568 906539 925291 87880 862989 89827 ...
 $ diagnosis   : chr  "B" "B" "B" "B" ...
 $ radius_mean : num  12.3 10.6 11 11.3 15.2 ...
 $ texture_mean : num  12.4 18.9 16.8 13.4 13.2 ...
 $ perimeter_mean : num  78.8 69.3 70.9 73 97.7 ...
 $ area_mean    : num  464 346 373 385 712 ...
 $ smoothness_mean : num  0.1028 0.0969 0.1077 0.1164 0.0796 ...
 $ compactness_mean : num  0.0698 0.1147 0.078 0.1136 0.0693 ...
 $ concavity_mean : num  0.0399 0.0639 0.0305 0.0464 0.0339 ...
 $ points_mean   : num  0.037 0.0264 0.0248 0.048 0.0266 ...
 $ symmetry_mean : num  0.196 0.192 0.171 0.177 0.172 ...
 $ dimension_mean : num  0.0595 0.0649 0.0634 0.0607 0.0554 ...
 $ radius_se     : num  0.236 0.451 0.197 0.338 0.178 ...
 $ texture_se    : num  0.666 1.197 1.387 1.343 0.412 ...
 $ perimeter_se  : num  1.67 3.43 1.34 1.85 1.34 ...
 $ area_se       : num  17.4 27.1 13.5 26.3 17.7 ...
```

Rysunek 1. Fragment podsumowania zbioru danych.

```
> table(wbcd$diagnosis)

  B    M 
357 212
```

Rysunek 2. Rozkład zmiennej – podział na dwie klasy występujące w zbiorze.

```
> wbcd$diagnosis<- factor(wbcd$diagnosis, levels = c("B", "M"),labels = c("łagodny", "złośliwy"))
> round(prop.table(table(wbcd$diagnosis)) * 100, digits = 1)
```

```
łagodny złośliwy
 62.7    37.3
```

Rysunek 3. Zmiana oznaczeń klas i podgląd rozkładu procentowego podziału na klasy.

```
> summary(wbcd[c("radius_mean", "area_mean", "smoothness_mean")])
```

radius_mean	area_mean	smoothness_mean
Min. : 6.981	Min. : 143.5	Min. :0.05263
1st Qu.:11.700	1st Qu.: 420.3	1st Qu.:0.08637
Median :13.370	Median : 551.1	Median :0.09587
Mean :14.127	Mean : 654.9	Mean :0.09636
3rd Qu.:15.780	3rd Qu.: 782.7	3rd Qu.:0.10530
Max. :28.110	Max. :2501.0	Max. :0.16340

Rysunek 4. Wyświetlenie wartości dla pierwszych trzech zmiennych.

Ponieważ zmienne mają zupełnie różne zakresy wartości, wymagana jest normalizacja danych.

```
> # Utworzenie funkcji do normalizacji
> normalize <- function(x) {return ((x - min(x)) / (max(x) - min(x)))}
> # Przetestowanie funkcji
> normalize(c(1,2,3,4,5,6))
[1] 0.0 0.2 0.4 0.6 0.8 1.0
```

Rysunek 5. Utworzenie funkcji przeprowadzającej normalizację i przetestowanie jej dla przykładowych danych.

Następnie przeprowadzono normalizację wszystkich kolumn liczbowych ze zbioru wejściowego.



```
> # Normalizacja zbioru
> wbcd_n <- as.data.frame(lapply(wbcd[2:31], normalize))
> summary(wbcd_n[c("radius_mean", "area_mean", "smoothness_mean")])
```

radius_mean	area_mean	smoothness_mean
Min. :0.0000	Min. :0.0000	Min. :0.0000
1st Qu.:0.2233	1st Qu.:0.1174	1st Qu.:0.3046
Median :0.3024	Median :0.1729	Median :0.3904
Mean :0.3382	Mean :0.2169	Mean :0.3948
3rd Qu.:0.4164	3rd Qu.:0.2711	3rd Qu.:0.4755
Max. :1.0000	Max. :1.0000	Max. :1.0000

Rysunek 6. Normalizacja i wyświetlenie nowych zakresów danych.

Do normalizacji zbioru danych wykorzystano funkcję lapply, która wywołuje podaną funkcję normalize dla wszystkich określonych kolumn danych.

Kolejnym krokiem było zbudowanie zbioru treningowego i testowego.

wbcd_test	100 obs. of 30 variables	
wbcd_train	469 obs. of 30 variables	
Values		
wbcd_test_diag	Factor w/ 2 levels "łagodny","złośliwy": 1 1 1 1 2 1 2 1 2...	
wbcd_train_diag	Factor w/ 2 levels "łagodny","złośliwy": 1 1 1 1 1 1 1 1 2 1...	

Rysunek 7. Utworzone zbiory. Zbiór treningowy i testowy zawierają atrybuty, a zmienne „diag” zawierają tylko klasy, do których należy dany rekord ze zbioru danych.

```
> # Utworzenie klasyfikatora kNN
> wbcd_test_pred <- knn(train = wbcd_train, test = wbcd_test, cl = wbcd_train_diag, k = 21)
```

Rysunek 8. Utworzenie klasyfikatora kNN dla powyższych zbiorów oraz wartości domyślnej k=21.

Uzyskano wyniki klasyfikacji, które należy teraz porównać do wyników oczekiwanych („wbcd\_train\_diag”).

	wbcd_test_pred		
wbcd_test_diag	łagodny	złośliwy	Row Total
łagodny	61	0	61
	1.000	0.000	0.610
	0.968	0.000	
	0.610	0.000	
złośliwy	2	37	39
	0.051	0.949	0.390
	0.032	1.000	
	0.020	0.370	
Column Total	63	37	100
	0.630	0.370	

Rysunek 9. Uzyskana tabela z informacjami o poprawnych i niepoprawnych klasyfikacjach.

Powtórzymy teraz proces tworzenia zbioru danych i wykorzystania kNN po standaryzacji danych.

```
> wbcd_z <- as.data.frame(scale(wbcd[-1]))
> summary(wbcd_z)
```

radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean
Min. :-2.0279	Min. :-2.2273	Min. :-1.9828	Min. :-1.4532	Min. :-3.10935	Min. :-1.6087
1st Qu.:-0.6888	1st Qu.:-0.7253	1st Qu.:-0.6913	1st Qu.:-0.6666	1st Qu.:-0.71034	1st Qu.:-0.7464
Median :-0.2149	Median :-0.1045	Median :-0.2358	Median :-0.2949	Median :-0.03486	Median :-0.2217
Mean : 0.0000	Mean : 0.0000	Mean : 0.0000	Mean : 0.0000	Mean : 0.00000	Mean : 0.0000
3rd Qu.: 0.4690	3rd Qu.: 0.5837	3rd Qu.: 0.4992	3rd Qu.: 0.3632	3rd Qu.: 0.63564	3rd Qu.: 0.4934
Max. : 3.9678	Max. : 4.6478	Max. : 3.9726	Max. : 5.2459	Max. : 4.76672	Max. : 4.5644
concavity_mean	points_mean	symmetry_mean	dimension_mean	radius_se	texture_se
Min. :-1.1139	Min. :-1.2607	Min. :-2.74171	Min. :-1.8183	Min. :-1.0590	Min. :-1.5529
1st Qu.:-0.7431	1st Qu.:-0.7373	1st Qu.:-0.70262	1st Qu.:-0.7220	1st Qu.:-0.6230	1st Qu.:-0.6942
Median :-0.3419	Median :-0.3974	Median :-0.07156	Median :-0.1781	Median :-0.2920	Median :-0.1973
Mean : 0.0000	Mean : 0.0000	Mean : 0.00000	Mean : 0.0000	Mean : 0.0000	Mean : 0.0000
3rd Qu.: 0.5256	3rd Qu.: 0.6464	3rd Qu.: 0.53031	3rd Qu.: 0.4706	3rd Qu.: 0.2659	3rd Qu.: 0.4661
Max. : 4.2399	Max. : 3.9245	Max. : 4.48081	Max. : 4.9066	Max. : 8.8991	Max. : 6.6494

Rysunek 10. Zbiór danych po standaryzacji. W oczy rzucają się wartości ujemne kolumn.

wbcd_test_pred_z			
wbcd_test_diag	łagodny	złośliwy	Row Total
----- ----- ----- -----			
łagodny	59	2	61
	0.967	0.033	0.610
	0.967	0.051	
	0.590	0.020	
----- ----- ----- -----			
złośliwy	2	37	39
	0.051	0.949	0.390
	0.033	0.949	
	0.020	0.370	
----- ----- ----- -----			
Column Total	61	39	100
	0.610	0.390	
----- ----- ----- -----			

Rysunek 11. Tabela uzyskana dla drugiego przypadku.

Dla tych danych i parametrów uzyskano lepszą dokładność w przypadku normalizacji danych, niż w przypadku standaryzacji.

Przetestujmy teraz oba przypadki dla innych wartości parametru k.

wbcd_test_diag	wbcd_test_pred		Row Total
	łagodny	złośliwy	
łagodny	61	0	61
	1.000	0.000	0.610
	0.953	0.000	
	0.610	0.000	
złośliwy	3	36	39
	0.077	0.923	0.390
	0.047	1.000	
	0.030	0.360	
Column Total	64	36	100
	0.640	0.360	

Rysunek 12. Klasyfikator kNN, k=15, dane normalizowane.

wbcd_test_diag	wbcd_test_pred_z		Row Total
	łagodny	złośliwy	
łagodny	61	0	61
	1.000	0.000	0.610
	0.953	0.000	
	0.610	0.000	
złośliwy	3	36	39
	0.077	0.923	0.390
	0.047	1.000	
	0.030	0.360	
Column Total	64	36	100
	0.640	0.360	

Rysunek 13. Klasyfikator kNN, k=15, dane standaryzowane.

Dla k=15 uzyskano identyczne wyniki, które nieznacznie różnią się od uzyskanych dla k=21.

wbcd_test_diag	wbcd_test_pred		Row Total
	łagodny	złośliwy	
łagodny	61	0	61
	1.000	0.000	0.610
	0.968	0.000	
	0.610	0.000	
złośliwy	2	37	39
	0.051	0.949	0.390
	0.032	1.000	
	0.020	0.370	
Column Total	63	37	100
	0.630	0.370	

Rysunek 14. Klasyfikator kNN, k=5, dane normalizowane.

wbcd_test_diag	wbcd_test_pred		Row Total
	łagodny	złośliwy	
łagodny	60	1	61
	0.984	0.016	0.610
	0.968	0.026	
	0.600	0.010	
złośliwy	2	37	39
	0.051	0.949	0.390
	0.032	0.974	
	0.020	0.370	
Column Total	62	38	100
	0.620	0.380	

Rysunek 15. Klasyfikator kNN, k=3, dane normalizowane.

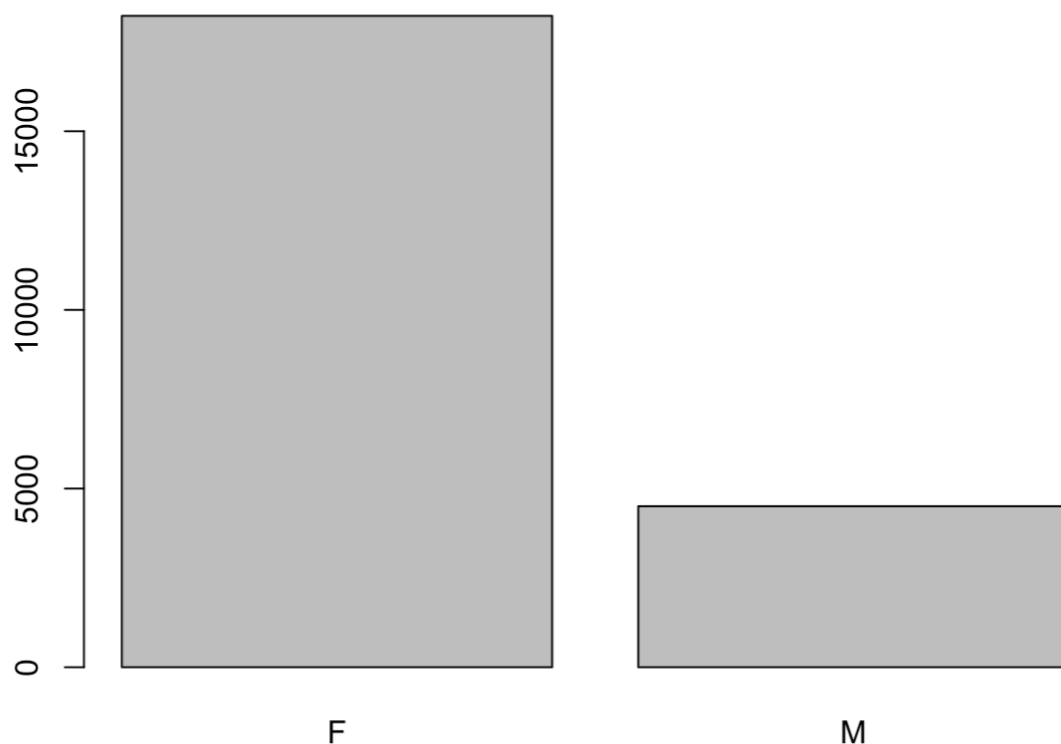
## Ćw. 2.

W tej części utworzono, w oparciu o klasyfikator „k średnich”, model grupowania danych dotyczących aktywności w portalach społecznościowych studentów szkół średnich.

```
> str(ds)
'data.frame': 25027 obs. of 40 variables:
 $ gradyear : int 2006 2006 2006 2006 2006 2006 2006 2006 2006 2006 ...
 $ gender    : chr "M" "F" "M" "F" ...
 $ age       : num 19 18.8 18.3 18.9 19 ...
 $ friends   : int 7 0 69 0 10 142 72 17 52 39 ...
 $ basketball : int 0 0 0 0 0 0 0 0 0 0 ...
 $ football  : int 0 1 1 0 0 0 0 0 0 0 ...
 $ soccer    : int 0 0 0 0 0 0 0 0 0 0 ...
 $ softball  : int 0 0 0 0 0 0 0 1 0 0 ...
 $ volleyball : int 0 0 0 0 0 0 0 0 0 0 ...
 $ swimming  : int 0 0 0 0 0 0 0 0 0 0 ...
 $ cheerleading: int 0 0 0 0 0 0 0 0 0 0 ...
 $ baseball  : int 0 0 0 0 0 0 0 0 0 0 ...
```

Rysunek 16. Fragment wczytanego zbioru danych.

Eksploracja danych – utworzono histogram płci występujących w zbiorze danych.



Rysunek 17. Histogram płci w zbiorze danych.

Należało rozwiązać problem brakujących danych w kolumnie gender i age. W przypadku gender, brakujące dane ustawiamy na wartość 3. Dla age, odrzucone zostały wartości niepoprawne i oddalone (np. osób, które są zbyt młode lub zbyt stare, by uczęszczać do szkoły średniej).

```
> # Uzupełnienie brakujących wartości
> ds$gender<-ifelse(is.na(ds$gender), 3, ds$gender)
```

Rysunek 18. Zamiana brakujących wartości zmiennej „gender” na 3.

```
> summary(ds$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  3.086  16.750  17.572  18.225  18.390 106.927   4149
```

Rysunek 19. Dane dotyczące zmiennej age.

```
> ds$age <- ifelse(ds$age >= 13 & ds$age < 20, ds$age, NA)
> summary(ds$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
 13.03  16.74  17.55  17.53  18.36  20.00   4512
```

Rysunek 20. Zamiana wartości niepoprawnych i ponowne wyświetlenie danych dla tej zmiennej.

Teraz brakujące dane dotyczące wieku zostały zastąpione średnią posiadanych w zbiorze, w zależności od roku ukończenia szkoły.

	gradyear	age
1	2006	18.65586
2	2007	17.70617
3	2008	16.76770
4	2009	15.83416

Rysunek 21. Obliczony średni wiek ucznia w zależności od roku, w którym skończył szkołę.

```
> ave_age <- ave(ds$age, ds$gradyear, FUN =function(x) mean(x, na.rm = TRUE))
> tail(ave_age, n=200)
 [1] 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416
[12] 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416
[23] 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416
[34] 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416
[45] 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416
[56] 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416
[67] 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416
[78] 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416
[89] 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416
[100] 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416 15.83416
```

Rysunek 22. Utworzenie wektora zawierającego odpowiedni wiek dla każdego ucznia po kolei, na podstawie wartości średniej zależnej od roku ukończenia szkoły.

```
> ds$age <- ifelse(is.na(ds$age), ave_age, ds$age)
> summary(ds$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 13.03  16.77  17.60  17.52  18.39  20.00
```

Rysunek 23. Dane dotyczące wieku po zamianie brakujących wartości. Wartości średnie i kwartyli tylko nieznacznie się zmieniły.



Teraz konieczna była normalizacja danych. Przeprowadzono ją dla wszystkich kolumn „zainteresowań”, które będą istotne w procesie grupowania. Ponownie wykorzystano w tym celu funkcję `lapply`.

```
> # Normalizacja danych
> interests <- ds[5:40]
> interests_z <- as.data.frame(lapply(interests, scale))
```

Rysunek 24. Przeprowadzenie normalizacji danych dotyczących zainteresowań.

Podczas normalizacji uzyskano informację o brakujących danych w wybranych kolumnach. Konieczna jest więc ich eliminacja. W tym celu wykorzystano funkcję `na.omit()`.

```
# Usunięcie brakujących danych dotyczących zainteresowań
ds <- na.omit(ds)
```

Rysunek 25. Usunięcie brakujących danych przy pomocy funkcji `na.omit()`. Usunięty został 1 rekord.

Teraz możliwe było grupowanie danych w oparciu o algorytm „k-Means”.

```
> teen_clusters <- kmeans(interests_z, 5)
> teen_clusters$size
[1] 18830 861 468 4150 717
> teen_clusters$centers
```

	basketball	football	soccer	softball	volleyball	swimming	cheerleading	baseball	tennis
1	-0.1150739	-0.12540118	-0.05922527	-0.078599282	-0.08796999	-0.08798505	-0.1000401	-0.12827538	-0.03671702
2	0.3546621	0.38398471	0.12358414	0.166066892	0.11013543	0.27125122	0.1543331	0.14216343	0.09411963
3	0.8586602	1.41140817	0.23362787	0.238673327	0.08051949	0.02050538	0.2552452	5.50827925	0.03417941
4	0.3691963	0.32607959	0.23018145	0.294489362	0.37941523	0.33315743	0.4154047	-0.05011712	0.15416187
5	-0.1011709	0.02360391	-0.07780342	0.004478259	-0.07058289	0.04325152	-0.1290239	-0.10720500	-0.06335186

	sports	cute	sex	sexy	hot	kissed	dance	band	marching
1	-0.09336476	-0.17155121	-0.091038918	-0.07746271	-0.126366473	-0.12935099	-0.14226338	-0.12864305	-0.1387778
2	0.87829626	0.49172824	2.037658314	0.54402005	0.298154593	3.04039859	0.47899800	0.36359475	-0.0308560
3	0.81936128	0.20690882	0.049476094	0.10149677	-0.003355748	-0.05246870	-0.03479051	-0.08692830	-0.1087195
4	0.16676115	0.65922128	-0.007092772	0.23117620	0.521816613	-0.02530659	0.54753032	-0.05234457	-0.1269430
5	-0.10275382	-0.03580252	-0.047253556	-0.02323589	-0.057460070	-0.07325849	0.01455146	3.30154254	4.4873751

	music	rock	god	church	jesus	bible	hair	dress	blonde
1	-0.1228684	-0.1001560	-0.09335991	-0.13317574	-0.07154295	-0.06000127	-0.18965762	-0.13642795	-0.02519394
2	1.1912596	1.2366984	0.36365615	0.16497589	0.08669110	0.06894618	2.66282775	0.56751079	0.35392384
3	0.1037457	0.5267178	-0.02654942	0.14897979	-0.03413654	0.02517903	0.04724646	-0.12786429	0.11318140
4	0.2325271	0.1088912	0.34054636	0.54640779	0.29742144	0.24728913	0.31158584	0.50834243	0.03017226
5	0.3827021	0.1711817	0.06139027	0.03952579	0.07557832	0.04522663	-0.05109391	0.04258137	-0.01186936

	mall	shopping	clothes	hollister	abercrombie	die	death	drunk
1	-0.172424361	-0.21745912	-0.18057141	-0.14862549	-0.14617512	-0.085683312	-0.07131817	-0.08234664
2	0.655610870	0.26716686	1.27714844	0.29785851	0.41444882	1.679838423	0.96950231	1.81414831
3	-0.005681041	-0.15323049	-0.03127980	-0.06998824	-0.06481591	-0.019018654	-0.04696344	-0.07368429
4	0.658770310	0.95611337	0.55444106	0.64617733	0.60880148	0.042809234	0.11978246	0.02059503
5	-0.068296090	-0.04384115	0.01986512	-0.14884200	-0.14024440	-0.002345471	0.04611065	-0.08700091

	drugs
1	-0.10423464
2	2.56570927
3	-0.06048576
4	-0.03894597
5	-0.07866715

Rysunek 26. Przeprowadzenie grupowania danych z podziałem na 5 klas. Wyświetlenie liczności tych klas oraz ich środków dla poszczególnych atrybutów.

Przekazywany parametr 5 określa ilość grup, jakiej oczekujemy. W przypadku takich danych nie wiemy, ile grup w nich występuje, w związku z tym sprawdzimy wyniki algorytmu dla kilku parametrów.

Powtórzmy powyższe operacje dla ilości grup równej 4.

```
> teen_clusters <- kmeans(interests_z, 4)
> teen_clusters$size
[1] 898 4522 18883 723
> teen_clusters$centers
```

	basketball	football	soccer	softball	volleyball	swimming	cheerleading	baseball	tennis
1	0.3549884	0.38689360	0.13676443	0.17226247	0.10406483	0.26709352	0.1711320	0.25592697	0.09174995
2	0.5028222	0.51426188	0.27235081	0.35777573	0.39028757	0.30629459	0.4217696	0.34442332	0.16679211
3	-0.1333520	-0.14241106	-0.06869668	-0.09286473	-0.09566434	-0.08761498	-0.1044990	-0.09062006	-0.04255447
4	-0.1029815	0.02244164	-0.07909465	-0.02626391	-0.07178546	0.04082936	-0.1212501	-0.10529209	-0.04573895

	sports	cute	sex	sexy	hot	kissed	dance	band	marching
1	0.8759297	0.49966163	1.993898159	0.53045284	0.32893067	2.95049555	0.50653953	0.36329995	-0.02047365
2	0.2908656	0.61879470	-0.003225317	0.22830157	0.46316823	-0.03135590	0.49084374	-0.05806506	-0.12991691
3	-0.1073170	-0.17132250	-0.092243026	-0.07945487	-0.12452833	-0.13006358	-0.14258745	-0.13005333	-0.13861570
4	-0.1043059	-0.01632498	-0.047179264	-0.01159062	-0.05305385	-0.07159472	0.02491136	3.30860839	4.45829870

	music	rock	god	church	jesus	bible	hair	dress	blonde
1	1.1883113	1.3031090	0.37013846	0.16765319	0.08508293	0.06699865	2.65658755	0.5726466	0.35590378
2	0.2213224	0.1496590	0.32054796	0.51800735	0.27397775	0.24310677	0.28326889	0.4354754	0.03785934
3	-0.1240678	-0.1040575	-0.09658812	-0.13343049	-0.07246446	-0.06308824	-0.19241082	-0.1334016	-0.02552656
4	0.3801502	0.1631643	0.05805145	0.03677203	0.07332574	0.04398560	-0.04600837	0.0491921	-0.01214861

	mall	shopping	clothes	hollister	abercrombie	die	death	drunk	drugs
1	0.68416933	0.28163145	1.32906846	0.3403540	0.4109293	1.639694752	0.94902248	1.76033439	2.52593922
2	0.58138222	0.83151852	0.46983760	0.5610234	0.5400367	0.041779659	0.10857169	0.01432555	-0.05035959
3	-0.16921146	-0.21092636	-0.17694486	-0.1452948	-0.1434859	-0.087800073	-0.07291988	-0.08390134	-0.10502266
4	-0.06663137	-0.04164486	0.03200657	-0.1369076	-0.1405477	-0.004764502	0.04669798	-0.08471834	-0.07942537

Rysunek 27. Przeprowadzenie grupowania z podziałem na 4 klasy.

Sprawdźmy, co się stanie dla większej ilości grup (k=12).

```
> teen_clusters <- kmeans(interests_z, 12)
> teen_clusters$size
[1] 1670 2176 657 502 1431 265 14407 1335 544 1068 32 939
> teen_clusters$centers
```

	basketball	football	soccer	softball	volleyball	swimming	cheerleading	baseball
1	1.446102018	1.45775749	0.42099711	1.15290912	0.975493781	0.05421854	0.11221583	1.43979243
2	0.006478608	0.06967047	0.03550183	-0.03263120	0.005058965	0.30615399	0.43825571	-0.08889792
3	0.070450900	0.14913035	0.12700120	0.08792130	0.135937270	0.22622020	0.33577506	0.01715269
4	-0.080606898	0.05474019	-0.09506875	-0.04754351	-0.070940281	0.06403718	-0.10972390	-0.11521535
5	-0.093660636	-0.13088286	-0.05918053	-0.09355375	-0.095633823	-0.03196555	-0.06252559	-0.10034466
6	0.232886176	0.24277988	0.17040329	-0.02724710	0.282382148	0.27053785	0.27032752	0.05720860
7	-0.171275914	-0.18705752	-0.06596358	-0.12718139	-0.115557773	-0.08829909	-0.10006095	-0.13904508
8	-0.067036139	-0.04656476	0.05055677	0.01004546	0.001931384	0.06344445	0.05646676	-0.06060979
9	0.510877898	0.47616796	0.16001137	0.19629587	0.154367032	0.34266397	0.18018759	0.30008440
10	-0.100819969	-0.06310248	-0.02981151	-0.05345148	-0.067248957	-0.03815784	-0.04528911	-0.07706872
11	0.035467073	0.09127178	0.01562215	-0.21154059	-0.163954735	0.11469917	-0.07602392	0.16310643
12	0.024592848	-0.02771078	0.05392404	0.03067932	0.025127006	0.07948935	0.03220884	-0.04049967

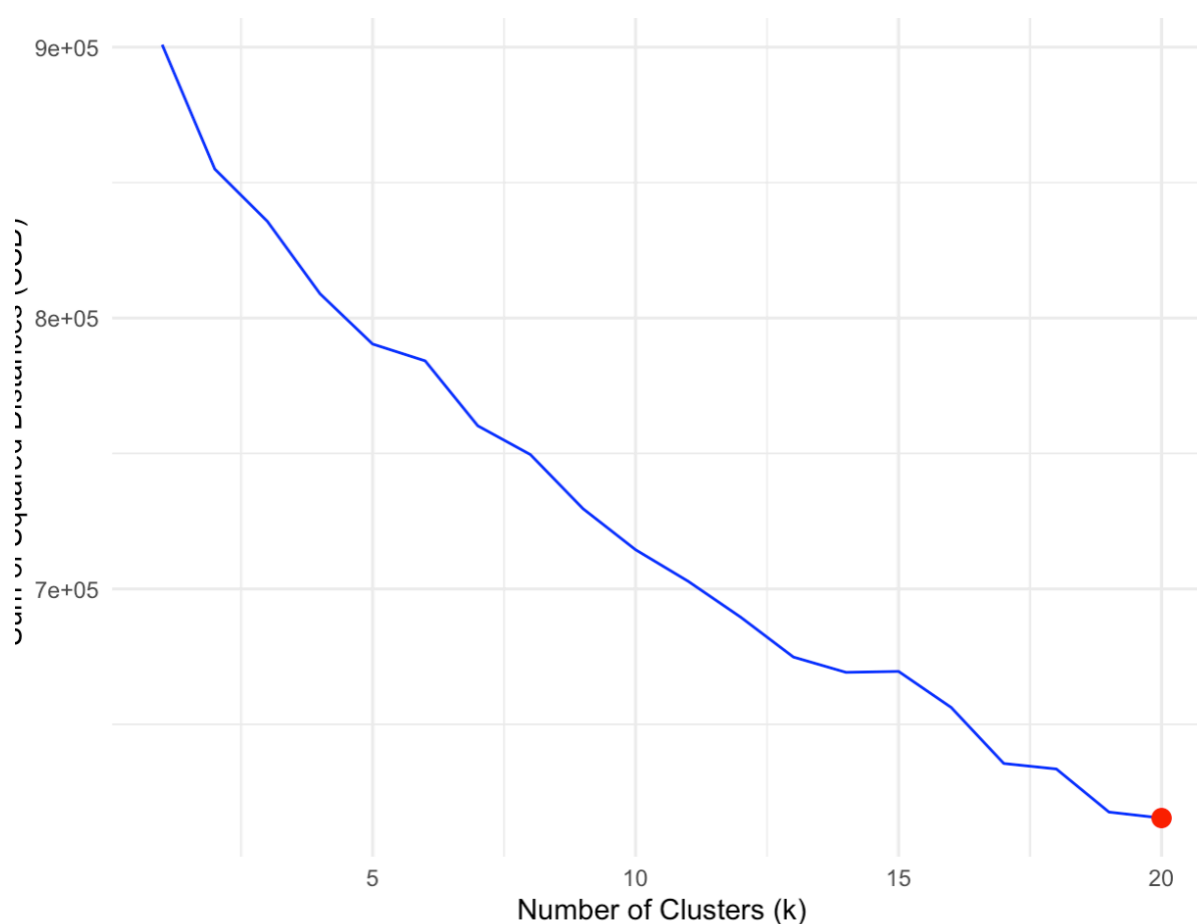
Rysunek 28. Przeprowadzenie grupowania z podziałem na 12 klas.

## Wnioski.

W przypadku klasyfikatora kNN, dla danych dotyczących raka nie uzyskano większych różnic w jakości uzyskanych wyników po zmianie wartości k. Najlepsze wyniki uzyskano dla k=5, jednak nawet dla większych wartości (15, 21) utworzono klasyfikator, który z dobrą skutecznością rozpoznawał klasę, do której należy każdy przypadek na podstawie innych atrybutów. Warto jednak zauważyć, że zbiór danych wejściowych, a w szczególności danych testowych, był dość mały, w związku z czym klasyfikator może nie radzić sobie tak dobrze dla przypadków, których nie było w oryginalnym zbiorze. Niemniej jednak przykład ten pokazuje, że nawet prosty klasyfikator oparty o algorytm „k najbliższych sąsiadów” może być skutecznie wykorzystywany w celach diagnozy.

W przypadku grupowania przy pomocy algorytmu k-Means widzimy, że – zarówno dla  $k=5$ ,  $k=4$  jak i  $k=12$  – jedna z uzyskanych grup jest znacznie większa od pozostałych. Na tej podstawie można wyciągnąć wniosek, że większość uczniów szkoły średniej ma zbliżone zainteresowania. Nawet w przypadku podziału na 12 grup uzyskujemy taką, która jest kilkakrotnie większa od pozostałych, a więc mimo że uzyskujemy więcej szczegółowych grup, i tym samym możliwe jest znalezienie dodatkowych zależności w danych, algorytm k-Means nadal przyporządkowuje większość uczniów do tej samej grupy.

W celu znalezienia optymalnej ilości grup można przeprowadzić analizę na kilka sposobów. Jednym z nich jest metoda polegająca na obliczeniu, dla różnych wartości  $k$ , sumy kwadratów odległości punktów w grupie od wyznaczonego środka tej grupy. Następnie możliwe jest określenie  $k$  jako wartości, dla której suma ta przestaje rosnąć w dużym tempie („zakrzywia się”).



Rysunek 29. Przeprowadzenie opisanej wyżej metody wyznaczenia optymalnej wartości  $k$ . Widzimy, że dla tego zbioru danych, wartość reprezentująca sumę kwadratów odległości od środków przestaje szybko spadać dopiero dla dużych wartości  $k$  (w okolicy 18).