VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

Papildomi vizualizavimo skyriai

Praktinis darbas Nr. 3

Darbą atliko:

Roland Gulbinovič ir Matas Amšiejus

Duomenų mokslas III kursas, 2 grupė

Vilnius 2022

Turinys

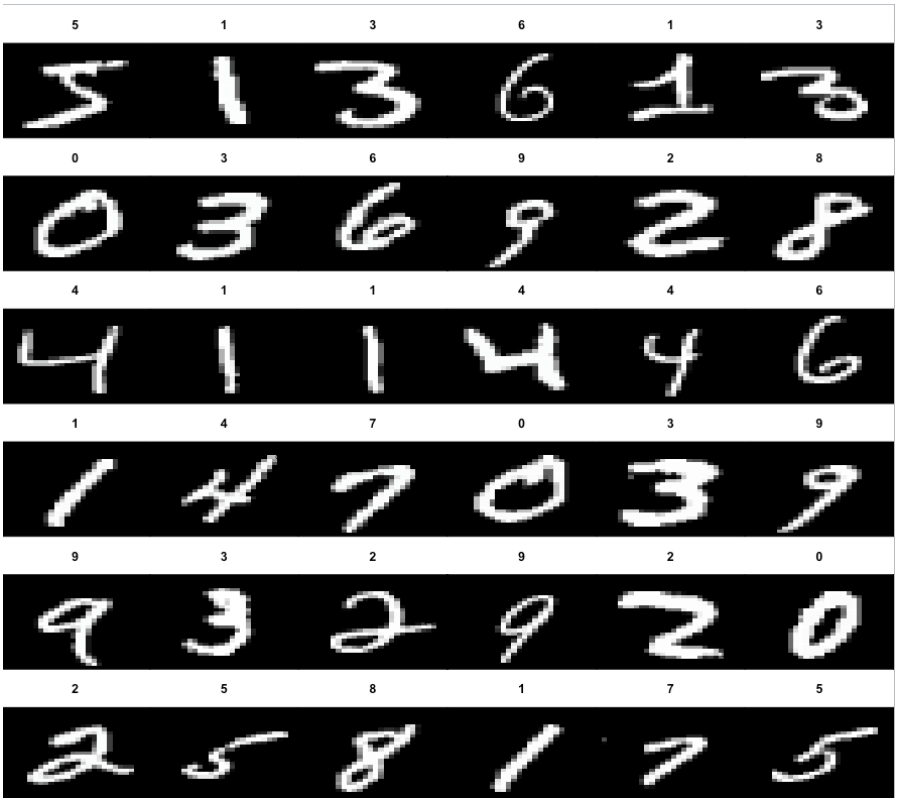
Įvadas

Duomenys

Darbui naudojama *MNIST* duomenų aibė. Tai duomenų rinkinys, sudarytas iš 60 000 mažų kvadratinių 28 × 28 pikselių pilkų atspalvių vaizdų su ranka rašytais skaitmenimis nuo 0 iki 9. Duomenų atributai:

- *784* stulpeliai atitinkantis kiekvienam pikseliui, kiekvieno stulpelio reikšmė parodo kaip stipriai nuspalvintas pikselis. [0; 255];
- *Label* – skaitmuo nuo 0 iki 9.

Kadangi duomenų yra labai daug, mes išsirinkome tik po 200 įrašų kiekvieno skaitmens (žr. kodą). Taigi darbui naudosime tik 2000 skaitmenų imtį.



1 pav. MNIST duomenų aibė.

Tyrimo tikslas

Šio darbo tikslas yra išskirti tiriamoje duomenų aibėje klasterius bei apibrėžti susidariusių klasterių specifiką.

Tyrimo uždaviniai

- Pasirinkti duomenų aibę klasterizavimui;

- Naudojant empyrinį, *Elbow* ir vidutinio silueto metodą įvertinti optimalų klasterių skaičių;
- Suklasterizuoti duomenis naudojant k-means ir hierarchinį klasterizavimo algoritmą;
- Suklasterizuoti ir vizualizuoti duomenis, gautus panaudojus dimensijos mažinimo algoritmus;
- Pateikti susidariusių klasterių aprašomąsias statistikas ir palyginti kas pasikeitė klasterizavus originalius duomenis, ir sumažinus dimensiją;
- Apibendrinti gautus rezultatus.

Klasterizavimas naudojant k-means

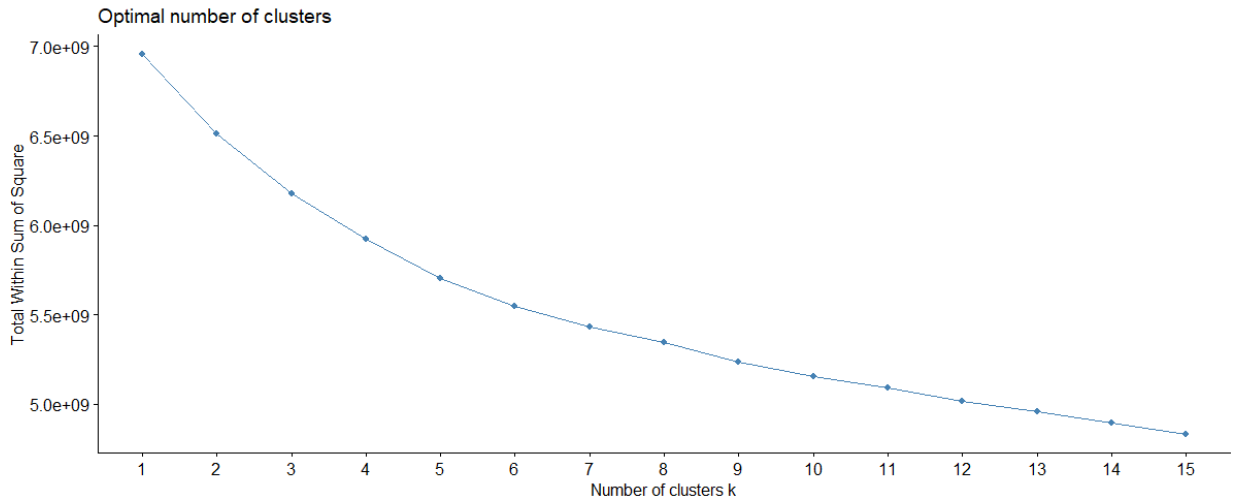
K-means algoritmas, pagal nurodytą k klasterių skaičių, priskiria kiekvieną duomenų tašką artimiausiam klasteriui, išlaikant kiek įmanoma mažesnius atstumus tarp taškų klasteryje ir jo centroido. (Plačiau žr. čia).

Optimalaus klasterių skaičiaus nustatymas

Iš pradžių mums reikia įvertinti optimalų klasterių skaičių. Tam naudosime empyrinį, *Elbow* ir vidutinio silueto metodą.

Pritaikius *empyrinio metodo* formulę: $N\_klast = \sqrt{m/2}$, kur m – objektų skaičius, gauname *19.81161*. Taigi, klasterių skaičius bus nedidesnis nei 19.

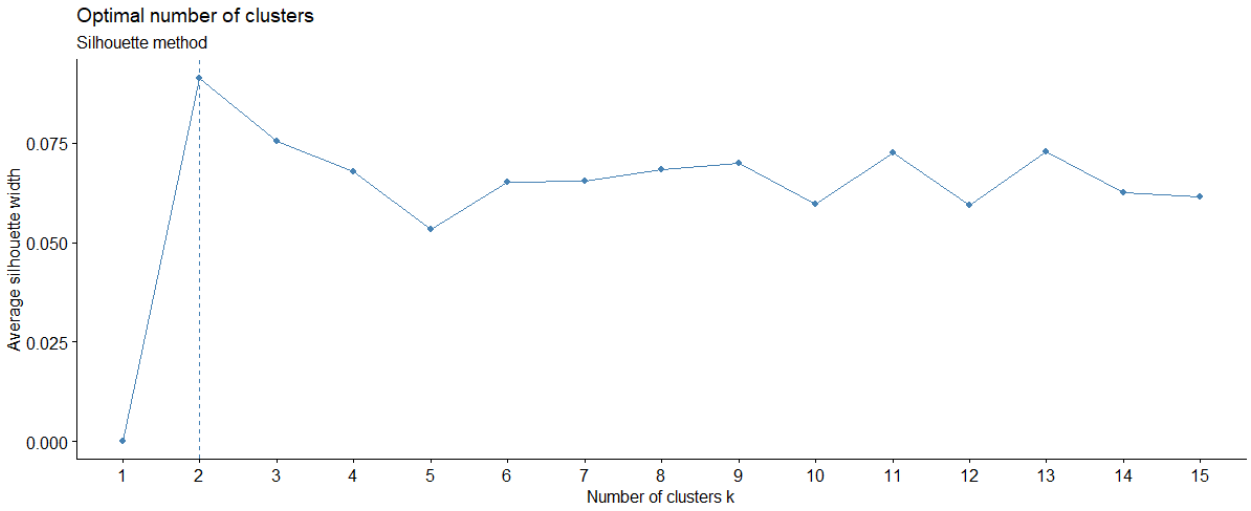
Toliau pritaikius *Elbow metodą,* gauname toki grafiką:



2 pav. Elbow metodas

Iš grafiko matome, kad nėra ryškaus linkio taško, tai iš šios diagramos įvertinti klasterių skaičių mums nepavyksta.

Paskutinis optimalaus klasterių skaičiaus įvertinimo metodas yra *vidutinio silueto metodas.*



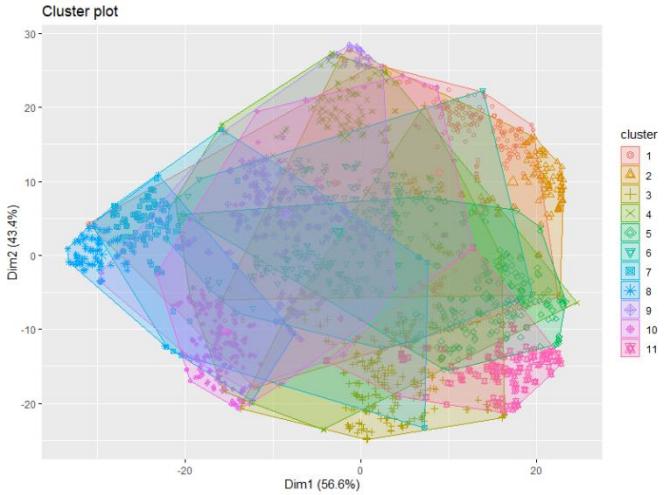
3 pav. Vidutinio silueto metodas

Iš *3 pav.* matome, kad optimalus klasterių skaičius yra 2 arba 11. Priimdami faktą, kad jau kažką žinome apie turimą duomenų aibę (arba taikydami ekspertinę nuomonę), teigiame, kad negali būti tik 2 klasteriai, todėl pasirenkame 11 klasterių.

Panaudojus visus tris metodus, nustatėme, kad naudosime 11 klasterių.

Klasterizavimas be dimensijos mažinimo

Iš pradžių klasterizavome pradinę duomenų aibę su visais 785 stulpeliais naudojant k-means klasterizavimą. Šis metodas yra dalinimu paremta klasterizacija, o klasteriai yra apibūdinami centroidais.



4 pav. Klasterių vizualizavimas taškine diagrama

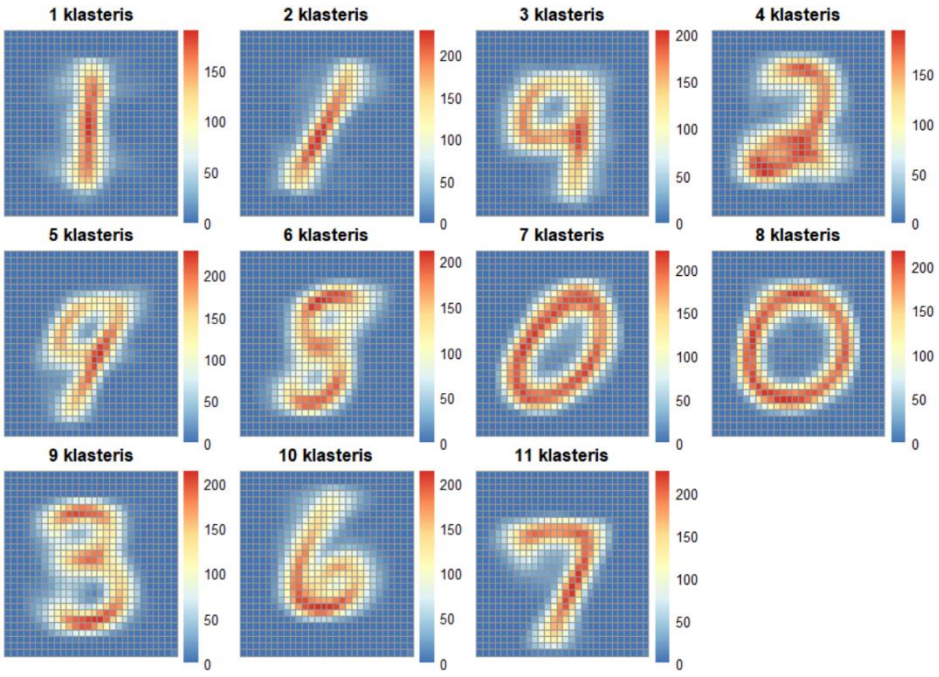
Kadangi turime labai daug stulpelių, vizualizacijai naudojome t-SNE dimensijos mažinimo algoritmą (iš ankstesnių darbų atrodė, kad t-SNE *mnist* duomenis vizualizuoja geriausiai). Tačiau vis tiek yra gana sunku kažką įžvelgti, nes yra daug išskirtinai parašytų skaičių, kurie priskiriami ne tam klasteriui. Todėl patikrinkime, kaip pasiskirstė mūsų originalūs skaitmenys (OG ...) naujuose klasteriuose (kuo mažiau skirtingų, tuo geriau turėtų būti).

1 lentelė. Skaitmenų pasiskirstymas klasteriuose

	OG 0	OG 1	OG 2	OG 3	OG 4	OG 5	OG 6	OG 7	OG 8	OG 9
Klast 1	0	0,48	0,08	0,07	0,02	0,1	0,06	0,07	0,06	0,04
Klast 2	0	0,62	0,08	0	0,02	0,1	0,02	0,07	0,1	0
Klast 3	0	0	0,03	0,01	0,42	0,06	0,02	0,08	0,03	0,35
Klast 4	0,02	0	**0,8**	0,07	0	0,01	0,03	0,01	0,05	0,01
Klast 5	0,01	0	0	0	0,37	0,05	0	0,15	0,08	0,36
Klast 6	0,03	0	0	0,18	0	0,36	0,02	0	0,38	0,02
Klast 7	**0,89**	0	0,03	0	0,01	0,02	0,05	0	0,01	0
Klast 8	**0,92**	0	0,01	0	0,01	0,01	0,03	0	0,01	0
Klast 9	0,05	0	0,1	0,48	0	0,18	0,01	0	0,18	0,02
Klast 10	0,03	0,01	0,04	0,02	0,01	0,02	**0,88**	0	0	0
Klast 11	0	0	0,01	0,01	0,01	0,01	0	**0,85**	0,01	0,1

Iš *1 lentelės* matome, kad 4, 7, 8, 10, 11 klasteriuose yra $\geq 80\%$ vienodų originalių skaitmenų, kas indikuoja neblogą atskyrimą. Kita vertus, 7 ir 8 klasteriai abu daugiausiai turi 0-to skaitmens, ką galima pamatyti ir šilumos diagramose (5 pav.). Tai gali indikuoti per didelį klasterių skaičių (ta pati problema ir su 1 ir 2 bei 3 ir 5 klasteriais). Likę nepaminėti klasteriai turi sunkiai atskiriamus skaitmenis, tad tarp jų nėra dominuojančio skaitmens (pavyzdžiui 6-ame klasteryje dominuoja tokie skaitmenys: 3-tas (0,18) 5-tas (0,36) ir 8-tas (0,38). Visi jie siejasi trimis horizontaliomis linijomis (5 paveikslėlyje ryškėja 3 raudoni brūkšniai)).

Tą taip pat galime pavaizduoti naudojant šilumines diagramas. Tam mes paėmėme kiekvienam klasteriui priskirtų duomenų eilutes ir suskaičiavome kiekvieno pikselio vidurkį. Taip gauname 28 x 28 matricą su kuria mes galime sukurti šiluminę diagramą, kurioje matosi, kokie skaitmenys yra kiekviename klasteryje Mėlyna spalva reiškia, kad pikselis spalvinamas buvo retai ir / arba neryškiai, raudona – dažnai ir / arba ryškiai. Tai papildomai matome, kad išryškėja tokie skaitmenys: 1, 2, 3, 6, 7, 8, 9. Ketvertas daugiausiai priskiriamas į klasterį, kuris primena devynetą, o penketas – į klasterius, kurie primena aštuonetą arba trejetą (anksčiau minėta trijų brūkšnių problema).



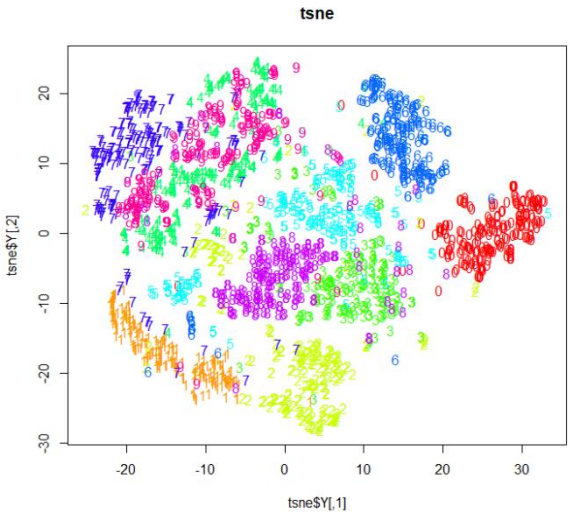
5 pav. Klasterių šiluminės diagramos

Klasterizavimas su t-SNE

Toliau panaudojome t-sne dimensijos mažinimo algoritmą, kad sumažinti duomenų dimensiją iki *dim = 2* (dabartinė dimensija buvo 784). Šie duomenys jau buvo panaudoti vizualizuojant 4 paveikslą.

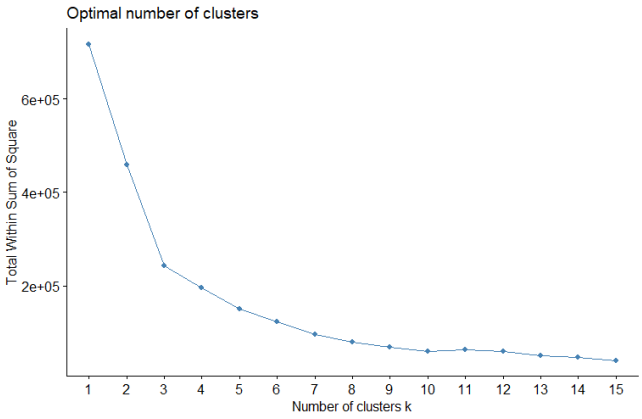
2 lentelė. Duomenys po t-SNE

V1	V2	Label
-9.475154	20.571205	0
-4.275862	27.347376	0
-12.112735	21.009255	0
-6.315152	27.890658	0
-5.802586	18.832378	0
-5.628559	26.944933	0
-6.572930	22.418048	0
-14.583289	21.990837	0
-13.057681	21.122542	0
-8.992712	22.400142	0

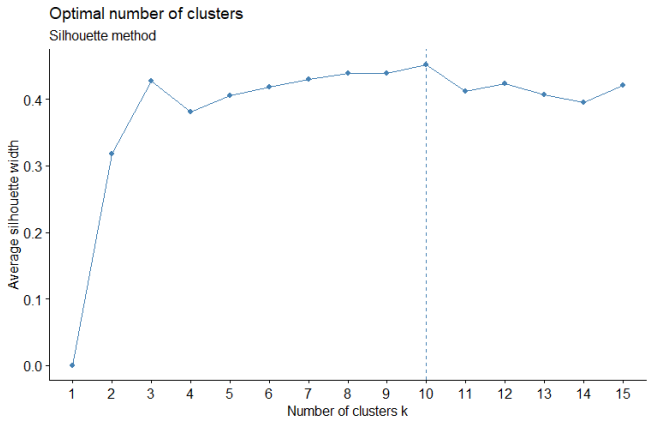


6 pav. 2D vizualizavimas

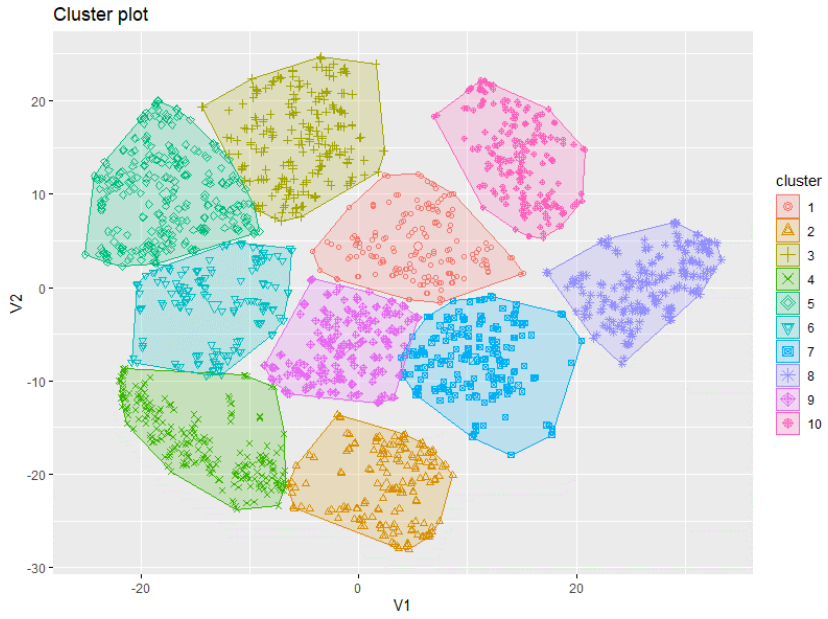
Vėl galime įvertinti optimalų klasterių kiekį. Iš *Elbow* metodo, panašiai kaip ir ankstesniame pavyzdyje, yra sunku kažką nustatyti (labiausiai atrodo, kad linkio taškas yra kai k=10), o pagal Silueto metodą gauname, kad optimalus klasterių skaičius yra lygus 10, jį ir naudosime.



7 pav. Elbow metodas (t-SNE)



8 pav. Silueto metodas (t-SNE)

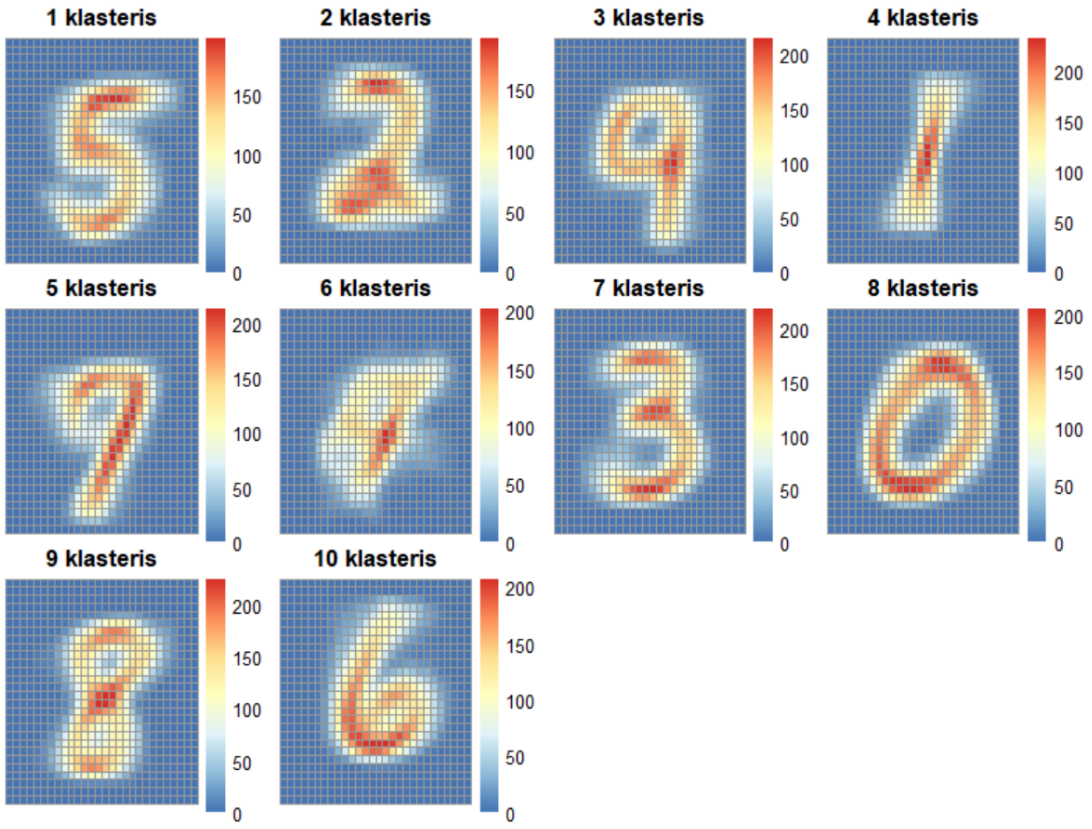


9 pav. Klasterių vizualizavimas taškine diagrama (t-SNE)

Matome, kad kai kurie klasteriai yra susitelkę labai glaudžiai (kaip 7 ir 9), kai kurie atsiskiria geriau. Dalyje klasterių matome nevienodus tankius, kas gali nulemti prastesnius rezultatus.

3 lentelė. Skaitmenų pasiskirstymas klasteriuose

	OG 0	OG 1	OG 2	OG 3	OG 4	OG 5	OG 6	OG 7	OG 8	OG 9
Klast 1	0	0	0,01	0,13	0,01	**0,79**	0	0	0,05	0,02
Klast 2	0	0,04	**0,92**	0,01	0	0	0	0,02	0,01	0
Klast 3	0	0	0,01	0	0,44	0,01	0	0,06	0,02	0,45
Klast 4	0	**0,86**	0,03	0	0	0,02	0,04	0,04	0	0,01
Klast 5	0	0	0	0	0,14	0	0	0,56	0,02	0,27
Klast 6	0,01	0,01	0,17	0,01	0,35	0,23	0	0,13	0,01	0,08
Klast 7	0,01	0	0,04	**0,78**	0	0,08	0,01	0	0,08	0,01
Klast 8	**0,97**	0	0,02	0	0	0,01	0,01	0	0	0
Klast 9	0	0	0,01	0,09	0	0	0	0	**0,89**	0,01
Klast 10	0,01	0	0,01	0	0,01	0,01	**0,94**	0	0	0



10 pav. Klasterių šiluminės diagramos (t-sne duomenys)

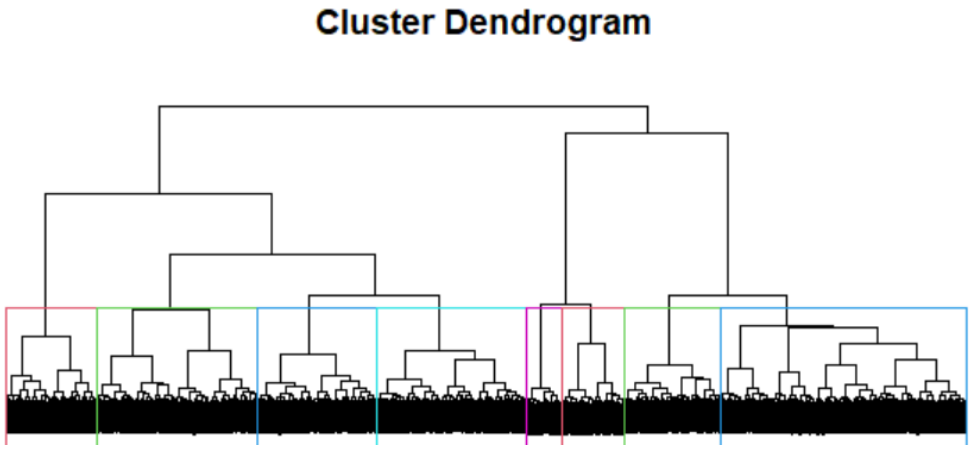
Sprendžiant pagal šilumines diagramas atrodo, kad t-sne transformacija padėjo geriau parinkti teisingą klasterių skaičių (matome mažiau dublikatinių porų). 1, 2, 4, 7, 8, 9, 10 klasteriai turėjo $\geq 78\%$ dominuojančių skaitmenų (nebuvo išsibarstę). Daugiausia problemų toliau kėlė ketverto ir devyneto atskyrimas vienas nuo kito (3 klasteris). Atsirado nauja problema – sunkiai atskiriamais skaitmenimis tapo ketvertas ir penketas (6 klasteris). Tačiau sumažinus dimensiją gavome žymiai daugiau neišsibarsčiusių klasterių (dvejais daugiau).

Klasterizavimas naudojant Hierarchinį metodą

Pagrindinė šio tipo klasterizavimo algoritmo idėja- sukurti hierarchinį ryšį tarp duomenų, kad būtų galima sudaryti klasterį. (Plačiau žr. čia)

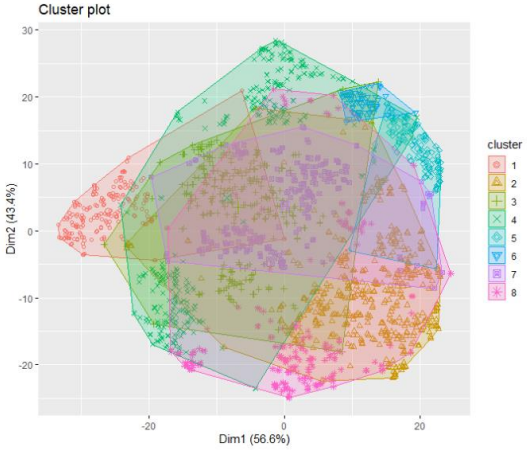
Klasterizavimas be dimensijos mažinimo

Toliau naudojome hierarchinį klasterizavimą, šis algoritmas jungia objektus į klasterius remiantis atstumais. Pirma sudaroma atstumų matrica tarp taškų, kiekvienas taškas yra priskiriamas į atskirą klasterį. Tada yra sujungiami 2 artimiausi klasteriai į naują klasterį, perskaičiuojama atstumų matrica. Šis metodas kartojamas iki kol lieka vienas klasteris. Įprastai klasteriai vizualizuojami dendogramomis:

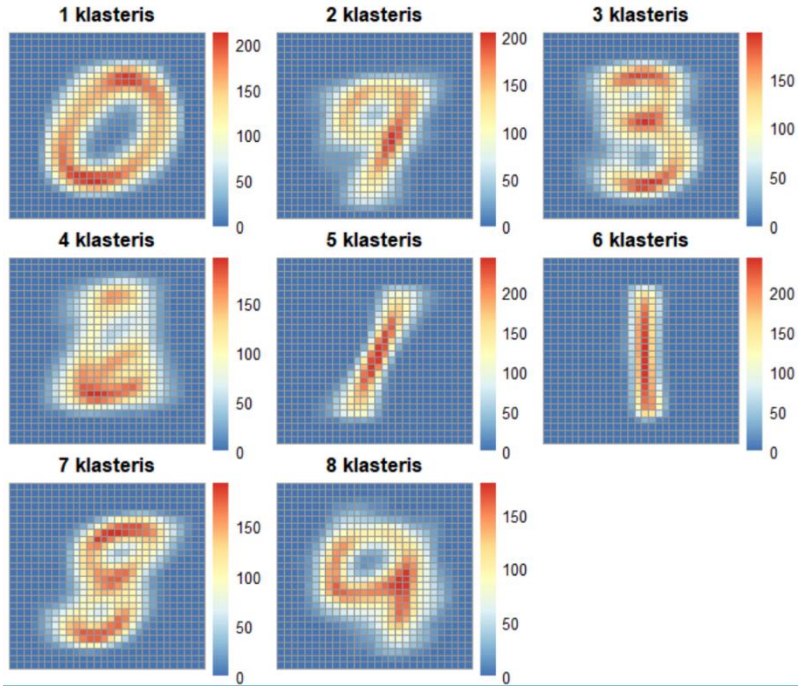


11 pav. Hierarchinio klasterizavimo dendograma su 10 klasterių

Vienas iš hierarchinio klasterizavimo metodo privalumų – nereikia iškart žinoti klasterių skaičiaus, jis parenkamas intuityviai, paprastai pagal dendogramą. Šiuo atveju vizualizavome dendogramą su 8 klasteriais (vėliau pabandysime vizualizuoti ir su trimis klasteriais). Tačiau pagal 12 pav. nematome gero klasterių atsiskyrimo, beveik visi klasteriai (išskyrus 5 ir 6) yra išsibarstę per visą plokštumą).



12 pav. Hierarchinio klasterizavimo taškinė diagrama



13 pav. Klasterių šiluminės diagramos (hierarchinis metodas)

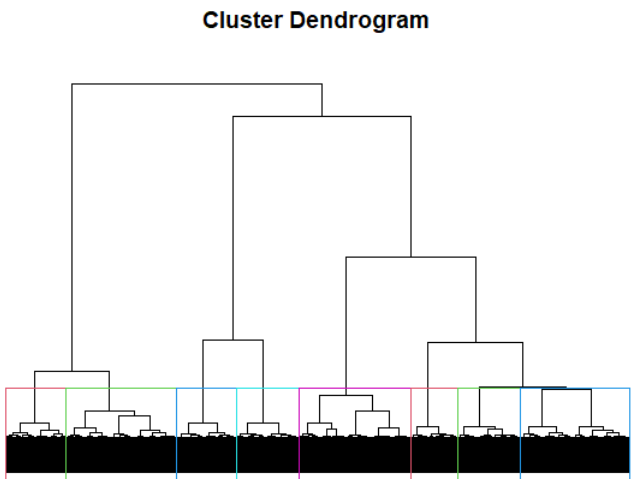
4 lentelė. Skaitmenų pasiskirstymas klasteriuose (hierarchinis metodas)

	OG 0	OG 1	OG 2	OG 3	OG 4	OG 5	OG 6	OG 7	OG 8	OG 9
Klast 1	**0,96**	0	0,02	0	0	0,02	0,01	0	0	0
Klast 2	0,02	0	0,01	0	0,28	0,09	0	0,32	0,01	0,26
Klast 3	0,02	0	0,01	0,49	0	0,18	0,02	0	0,26	0,02
Klast 4	0,01	0,01	0,44	0,02	0	0,01	0,49	0	0,01	0,01
Klast 5	0	**0,93**	0	0	0,01	0	0,01	0,05	0	0
Klast 6	0	**0,96**	0	0	0	0	0,01	0	0	0,03
Klast 7	0	0	0,03	0,15	0	0,35	0	0,04	0,42	0
Klast 8	0	0	0,17	0,01	0,28	0,01	0,14	0,09	0,03	0,28

Iš lentelės matome, kad 1, 5, 6, klasteriai turi daug vieno tipo skaitmenų. Tačiau įdomu tai, kad 5 ir 6 klasteriai (abiejuose dominuoja vienetai) yra atskiruose klasteriuose (nors iš 13 pav. matosi, kad vienetai skiriasi palinkimu). Kaip ir visur, problematiškiausi skaitmenys toliau liko 3, 4, 5, 8 ir 9. Taip pat darant šiuo metodu atsirado ketvirtas klasteris, kuriame sunku nustatyti kas yra vizualizuota (sujungta daug dvejeto ir šešeto skaitmenų, ko anksčiau dar niekada nebuvo).

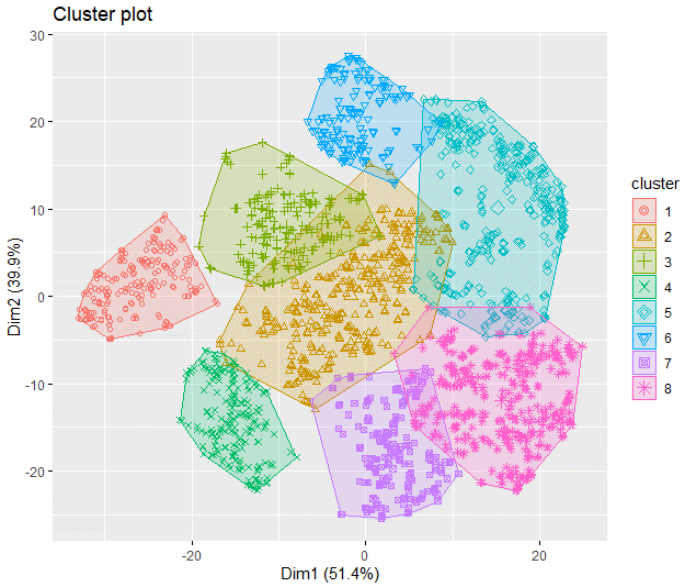
Klasterizavimas su t-SNE

Naudojame tuos pačius duomenis, gautus taikant k-means metodą sumažintos dimensijos duomenims. Vėl braižome dendogramą.



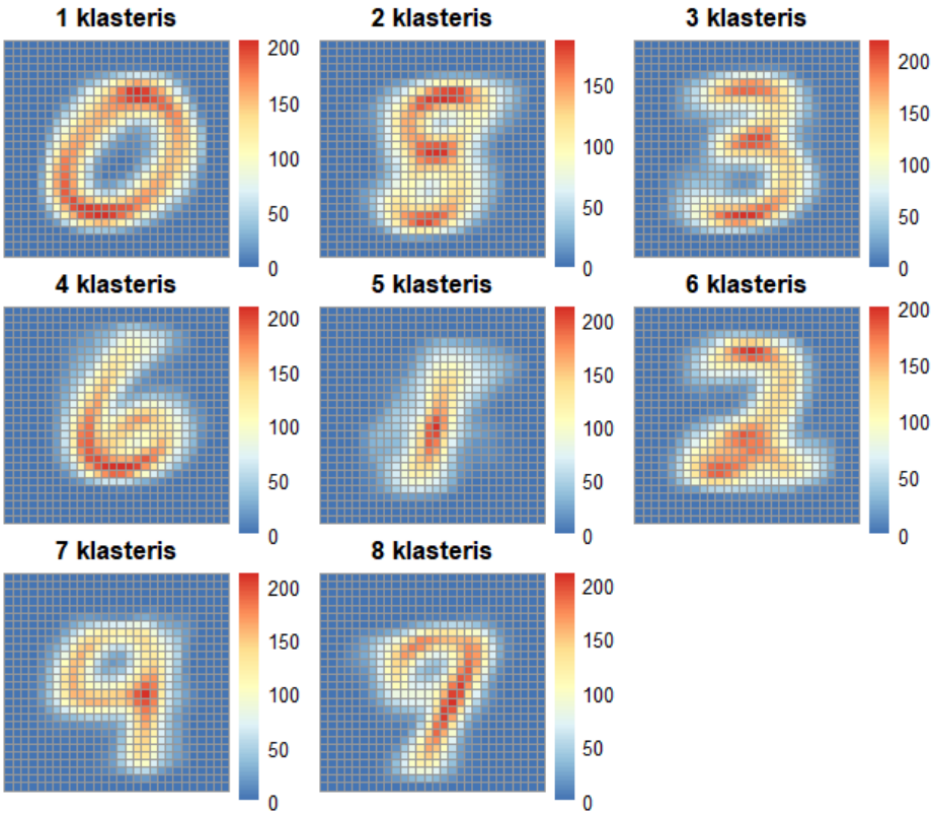
14 pav. Hierarchinio metodo dendograma su 10 klasterių (t-sne duomenims)

Šį kartą atrodo, jog reikia imti mažesnį klasterių skaičių. Bet tam, kad būtų galima palyginti paprastų ir sumažintos dimensijos duomenų klasterizavimo pastovumą, imame 8 klasterius.



15 pav. Hierarchinio klasterizavimo taškinė diagrama

Matome, kad daug klasterių persidengia, ypač 3-čias (pliusai) su 2-tu (trikampiai) (trejeto ir penketo skaitmenų problema). Gerai atsiskiria 1-as, 4-as klasteriai (atitinkamai skaitmenys 0 ir 6).



16 pav. Klasterių šiluminės diagramos (t-sne duomenys) hierarchinis metodas

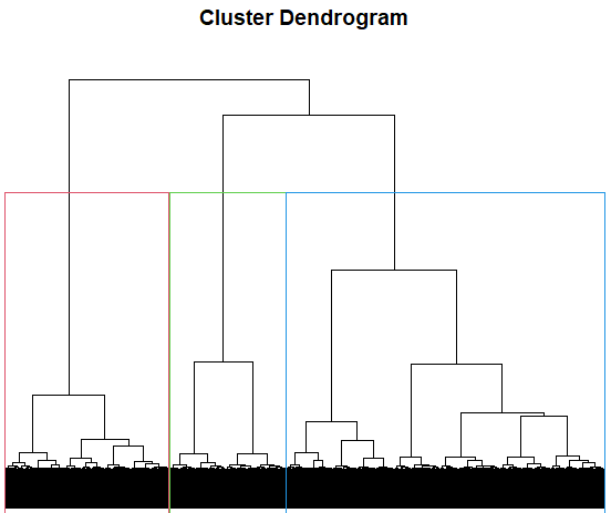
5 lentelė. Skaitmenų pasiskirstymas klasteriuose

	OG 0	OG 1	OG 2	OG 3	OG 4	OG 5	OG 6	OG 7	OG 8	OG 9
Klast 1	**0,97**	0	0,02	0	0	0,01	0,01	0	0	0
Klast 2	0,01	0	0,01	0,06	0	0,39	0,01	0,01	0,5	0,01
Klast 3	0,01	0	0,04	**0,82**	0	0,07	0	0	0,05	0
Klast 4	0,01	0	0,01	0	0,02	0,01	**0,96**	0	0,01	0
Klast 5	0,01	0,55	0,1	0,01	0,14	0,11	0,03	0,04	0	0,01
Klast 6	0	0,01	**0,97**	0,01	0	0	0	0,01	0,01	0
Klast 7	0	0	0,01	0,04	0,48	0,03	0	0,02	0,02	0,41
Klast 8	0	0	0	0	0,14	0	0	0,5	0,03	0,32

Įdomu tai, kad šį kartą į 5 klasterį (kuris labiausiai panašus į skaitmenį vienetą) pateko daug visokių skaitmenų (anksčiau vienetas kaip skaitmuo atsiskirdavo neblogai). Gerai atsiskyrė 1, 3 (dauguma trejetai), 4 (dauguma šešetai) ir 6 (dauguma dvejetai) klasteriai.

Hierarchinis algoritmas su t-sne ir mažesniu klasterių skaičiumi

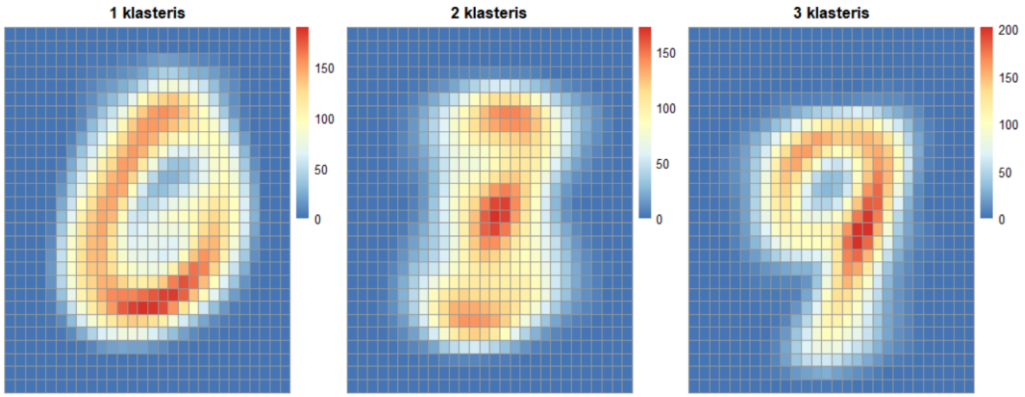
Iš prieš tai atlikto bandymo matėme, kad pagal dendogramą galima imti ir žymiai mažesnį klasterių skaičių. Panašu, kad didžiausios vertikalios linijos yra kai imame tris klasterius. Tą ir padarykime.



17 pav. Hierarchinio metodo dendograma su t-SNE duomenimis ir trimis klasteriais

18 pav. Hierarchinio metodo taškinė diagrama su t-SNE duomenimis ir trimis klasteriais

Matome, kad klasteriai atskirti neblogai, nors yra persidengimų (pagal t-SNE vizualizaciją). Patikrinkime šilumines diagramas ir pasiskirstymo lentelę. Tikėtina, kad joje bus daugiausia sukombinuoti skaitmenys, kurie iki šiol dažnai patekdavo į vieną klasterį (kaip skaitmenys 3, 5, 8 arba 4, 7, 9).



19 pav. Šiluminė diagrama

6 lentelė. Skaitmenų pasiskirstymas klasteriuose

	OG 0	OG 1	OG 2	OG 3	OG 4	OG 5	OG 6	OG 7	OG 8	OG 9
Klast 1	**0,5**	0	0,01	0	0,01	0,01	**0,48**	0	0	0
Klast 2	0,01	**0,19**	**0,18**	**0,18**	0,05	**0,18**	0,01	0,02	**0,18**	0,01
Klast 3	0	0	0,01	0,01	**0,26**	0,01	0	**0,33**	0,02	**0,35**

Matome, kad į pirmąjį klasterį pateko daug 0 ir 6 skaitmenų, į antrą, kaip ir tikėtasi – trejetas, penketas ir aštuonetas. Tačiau prisidėjo ir dauguma vienetų ir dvejetų. Trečiajame klasteryje pateko kaip ir prognozuota: ketvertas, septynetas ir devynetas.

Išvados

Matėme, kad su visais metodais ir duomenimis nebuvo išskirtinai gero klasterizavimo būdo. Į klasterius patekdavo daug skirtingų skaitmenų, kai kurie klasteriai buvo beveik identiški ir prastai atsiskirdavo nuo kitų (pavyzdžiui klasteriai, kuriuose atvaizduoti skaitmenys iš šiluminių diagramų primena devynetus arba vienetus). Taip pat ypač problematiškas tapo 3, 5, 8 ir 4, 7, 9 skaitmenų atskyrimas. Tačiau tą galima pateisinti, nes sprendžiant pagal duomenis, kai kurie skaitmenys būdavo sunkiai atskiriami net sprendžiant žmogui (nes kiekvieno žmogaus rašyba skiriasi, vieni daugiau pasuka, kiti – padeda brūkšnį kur nereikia ar atvirkščiai). K-means metodo atžvilgiu, klasteriai buvo nepastovūs, nes netgi buvo parinktas skirtingas optimalus klasterių skaičius. Tačiau lyginant originalių ir sumažintos dimensijos duomenų klasterizavimą gavome, kad k-means metodas geriau klasterizuoja mažesnės dimensijos duomenis, nes iš šiluminių diagramų buvo galima įžvelgti daugiau skaitmenų, o ir pačių klasterių, su dominuojančiais skaitmenimis buvo daugiau. Su hierarchiniu metodu rezultatai nepralenkė k-means metodo, buvo daug dublikatų ar prastai atsiskiriančių klasterių. Sumažinus dimensiją, rezultatai pagerėjo nežymiai. Imant tris klasterius, susidarė klasteriai iš skaitmenų, kurie dažnai buvo maišomi ankstesniuose algoritmuose.

Šaltiniai

K-means

„Unsupervised K-Means Clustering Algorithm“ žurnalas: „IEEE Access“. Autoriai: Kristina P. Sinaga, Miin-Shen Yang. 2020 balandis. Nuoroda: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9072123

„K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks”. Autoriai – Imad Dabbura, 2018 rugsėjis. Nuoroda: https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a

Hierarchinis

„Introduction to HPC with MPI for Data Science“. Autoriai – Frank Nielsen. 8 skyrius. : Nuoroda: https://link.springer.com/content/pdf/10.1007/978-3-319-21903-5.pdf

„Understanding the concept of Hierarchical clustering Technique“. Autoriai – Chaitanya Reddy Patlolla. 2018 gruodis. Nuoroda: https://towardsdatascience.com/understanding-the-concept-of-hierarchical-clustering-technique-c6e8243758ec

Priedai

Kodas

```
library(dplyr)
library(stats)
library(ggplot2)
library(ggfortify)
# install.packages("devtools")
#devtools::install_github("jlmelville/mnist")
library(mnist)

mnist <- download_mnist()

set.seed(67)
data <- mnist %>% group_by(Label) %>% sample_n(size = 200)

############################
# K-MEANS
############################

# Duomenys be labels
data1 <- data[,-785]

## T-SNE dimensijos mazinimas (reikes tolimesniame darbe)

library(Rtsne)

# sutvarkau duomenis
labels<-data$Label
data$Label<-as.factor(data$Label)

colors = rainbow(length(unique(data$Label)))
names(colors) = unique(data$Label)

# T-SNE (2D)
set.seed(67)
```

```r
tsne <- Rtsne(data1, dims = 2, perplexity=30, verbose=TRUE, max_iter = 500)

# T-SNE vizualizacija
plot(tsne$Y, t='n', main="tsne")
text(tsne$Y, labels=data$Label, col=colors[data$Label])

data_1<- as.data.frame(tsne$Y)

data_1$Label <- data$Label


############### KLASTERIU SKAICIAI ##################

########## empyrinis metodas

empyrinis <- function(duom) {
  klast_sk <- sqrt(length(duom)/2)
  return(klast_sk)
}

# Klasteriu ne daugiau nei 19
empyrinis(data)

######### Elbow

# load required packages
library(factoextra)
library(NbClust)

# Elbow klasteriu parinkimo
fviz_nbclust(data, kmeans, method = "wss", k.max = 15)


# kitas (destytojos) variantas (leisti nebutina)
wcss = vector()
for (i in 1:10) wcss[i] = sum(kmeans(data, i)$withinss)
plot(1:10,
     wcss,
     type = 'b',
     main = paste(' Elbow metodas'),
     xlab = 'Klasteriu skaicius',
     ylab = 'WCSS')

# Panasu, kad nematome linkio tasko


######### silhouette method

fviz_nbclust(data, kmeans, method = "silhouette", k.max = 15) +
  labs(subtitle = "Silhouette method")

# 2,3 arba 11/13

######### KLASTERIZAVIMAS #############

# Imkime 11 teigiant, kad tikrai negali buti 2 klasteriai
set.seed(67)
km.res <- kmeans(x = data1, centers = 11)
y_kmeans = km.res$cluster
#km.res
y_kmeans


# Vizualizacija taskine diagrama (pagal PCA)
#fviz_cluster(km.res, data1, stand = FALSE, geom = 'point')

# Vizualizacija taskine diagrama (pagal t-SNE)
data_tsne <- data_1[,-3]
data_tsne$fill <- rep(0, nrow(data_tsne))

a<-matrix(unlist(data_tsne), nrow = 2000, ncol = 3)
```

```
fviz_cluster(list(data = a, cluster = y_kmeans), stand = F , geom = "point")


# heatmapas
data_c <- data1
data_c$cluster<- y_kmeans

library(pheatmap)

heatmapping<- function(duomenys, c) {
  data_c1<- duomenys %>% filter(cluster == c)
  means_1<-matrix(colMeans(data_c1[,-ncol(data_c1)]), ncol = sqrt(784), byrow = TRUE)
  pheatmap(means_1, cluster_rows = FALSE, cluster_cols = FALSE,
           main = paste(as.character(c),"klasteris"))
}
# load libraries
library(ggplotify)
library(pheatmap)
library(patchwork)

p1<-as.ggplot(heatmapping(data_c, 1))
p2<-as.ggplot(heatmapping(data_c, 2))
p3<-as.ggplot(heatmapping(data_c, 3))
p4<-as.ggplot(heatmapping(data_c, 4))
p5<-as.ggplot(heatmapping(data_c, 5))
p6<-as.ggplot(heatmapping(data_c, 6))
p7<-as.ggplot(heatmapping(data_c, 7))
p8<-as.ggplot(heatmapping(data_c, 8))
p9<-as.ggplot(heatmapping(data_c, 9))
p10<-as.ggplot(heatmapping(data_c, 10))
p11<-as.ggplot(heatmapping(data_c, 11))
p1 + p2 + p3 + p4 + p5 +p6 + p7 + p8 + p9 + p10 + p11

######### dazniu lentele pagal kluster

testas <- data
testas$clust <- y_kmeans

# Funkcija gauti pasiskirstyma labels naujuose klasteriuose
lent <- function(duomenys, klasteris){
  temp1 <- duomenys %>% filter(clust == klasteris) %>% group_by(Label) %>%
    summarise(kiek = n())
  n_clust <- duomenys %>% filter(clust == klasteris) %>% nrow()
  temp1$kiek <- round(temp1$kiek/n_clust,2)
  temp1
}

temp <- lent(testas, 1)
temp <- merge(temp, lent(testas,2), by="Label", all = T)
temp <- merge(temp, lent(testas,3), by="Label", all = T)
temp <- merge(temp, lent(testas,4), by="Label", all = T)
temp <- merge(temp, lent(testas,5), by="Label", all = T)
temp <- merge(temp, lent(testas,6), by="Label", all = T)
temp <- merge(temp, lent(testas,7), by="Label", all = T)
temp <- merge(temp, lent(testas,8), by="Label", all = T)
temp <- merge(temp, lent(testas,9), by="Label", all = T)
temp <- merge(temp, lent(testas,10), by="Label", all = T)
temp <- merge(temp, lent(testas,11), by="Label", all = T)

temp[is.na(temp)] <- 0
names(temp) <- c("OG", "Klast 1", "Klast 2", "Klast 3", "Klast 4", "Klast 5",
                 "Klast 6", "Klast 7", "Klast 8", "Klast 9", "Klast 10", "Klast 11")
gal_klast_pas <- as.data.frame(t(temp))
gal_klast_pas <- gal_klast_pas[-1,]
names(gal_klast_pas) <- c("OG 0", "OG 1", "OG 2", "OG 3", "OG 4", "OG 5", "OG 6",
                          "OG 7", "OG 8", "OG 9")

write.csv(gal_klast_pas, file = "lent1.csv")

########## T-SNE ir KLASTERIZAVIMAS ##############
```

```
##### KLASTERIU SKAICIAI
#ELBOW
fviz_nbclust(data_1, kmeans, method = "wss", k.max = 15)

# Siluete
fviz_nbclust(data_1, kmeans, method = "silhouette", k.max = 15) +
  labs(subtitle = "Silhouette method")


##### K-means su naujais 2D dataframe
set.seed(67)
km.res2 <- kmeans(x = data_1, centers = 10)
y_kmeans2 = km.res2$cluster

# heatmapas
data_c2 <- data1
data_c2$cluster<- y_kmeans2

p1<- as.ggplot(heatmapping(data_c2, 1))
p2<- as.ggplot(heatmapping(data_c2, 2))
p3<- as.ggplot(heatmapping(data_c2, 3))
p4<- as.ggplot(heatmapping(data_c2, 4))
p5<- as.ggplot(heatmapping(data_c2, 5))
p6<- as.ggplot(heatmapping(data_c2, 6))
p7<- as.ggplot(heatmapping(data_c2, 7))
p8<- as.ggplot(heatmapping(data_c2, 8))
p9<- as.ggplot(heatmapping(data_c2, 9))
p10<- as.ggplot(heatmapping(data_c2, 10))
p1+ p2+p3+p4+p5+p6+p7+p8+p9+p10

##### Vizualizacija taskine diagrama
data_2 <- as.data.frame(data_1)

fviz_cluster(km.res2, data_2[,-3], stand = FALSE, geom = 'point')

######### dazniu lentele pagal kluster

testas2 <- data
testas2$clust <- y_kmeans2


temp2 <- lent(testas2, 1)
temp2 <- merge(temp2, lent(testas2, 2), by="Label", all = T)
temp2 <- merge(temp2, lent(testas2, 3), by="Label", all = T)
temp2 <- merge(temp2, lent(testas2, 4), by="Label", all = T)
temp2 <- merge(temp2, lent(testas2, 5), by="Label", all = T)
temp2 <- merge(temp2, lent(testas2, 6), by="Label", all = T)
temp2 <- merge(temp2, lent(testas2, 7), by="Label", all = T)
temp2 <- merge(temp2, lent(testas2, 8), by="Label", all = T)
temp2 <- merge(temp2, lent(testas2, 9), by="Label", all = T)
temp2 <- merge(temp2, lent(testas2, 10), by="Label", all = T)

temp2[is.na(temp2)] <- 0
names(temp2) <- c("OG", "Klast 1", "Klast 2", "Klast 3", "Klast 4", "Klast 5",
                  "Klast 6", "Klast 7", "Klast 8", "Klast 9", "Klast 10")
gal_klast_pas2 <- as.data.frame(t(temp2))
gal_klast_pas2 <- gal_klast_pas2[-1,]
names(gal_klast_pas2) <- c("OG 0", "OG 1", "OG 2", "OG 3", "OG 4", "OG 5", "OG 6",
                           "OG 7", "OG 8", "OG 9")
write.csv(x = gal_klast_pas2, file = "lent2.csv")




################################
# HIERARCHINIS METODAS
################################

############## Be t-sne ################
```

```
# Naudojant dendograma, identifikuojamas optimalus klasteriu skaicius
set.seed(67)
dendrogram = hclust(d = dist(data1, method = 'euclidean'), method = 'ward.D')
plot(dendrogram,
     main = paste('Dendograma'),
     xlab = 'Pikseliai',
     ylab = 'Euklidinis atstumas')

# Duomenys klasterizuojami hierarchiniu algoritmu
set.seed(67)
hc = hclust(d = dist(data1, method = 'euclidean'), method = 'ward.D')
y_hc = cutree(hc, 8)

# Nuspalvina dendograma
plot(hc)
rect.hclust(hc , k = 8, border = 2:6)


# heatmapai
data_c_hc <- data1
data_c_hc$cluster<- y_hc


p1<- as.ggplot(heatmapping(data_c_hc, 1))
p2<- as.ggplot(heatmapping(data_c_hc, 2))
p3<- as.ggplot(heatmapping(data_c_hc, 3))
p4<- as.ggplot(heatmapping(data_c_hc, 4))
p5<- as.ggplot(heatmapping(data_c_hc, 5))
p6<- as.ggplot(heatmapping(data_c_hc, 6))
p7<- as.ggplot(heatmapping(data_c_hc, 7))
p8<- as.ggplot(heatmapping(data_c_hc, 8))
#p9<- as.ggplot(heatmapping(data_c_hc, 9))
#p10<- as.ggplot(heatmapping(data_c_hc, 10))

p1+ p2+p3+p4+p5+p6+p7+p8#+p9+p10


### vizualizavimas taskine diagrama
#t-sne
fviz_cluster(list(data = a, cluster = y_hc), stand = F , geom = "point")


######### dazniu lentele pagal kluster

testas3 <- data
testas3$clust <- y_hc


temp3 <- lent(testas3, 1)
temp3 <- merge(temp3, lent(testas3, 2), by="Label", all = T)
temp3 <- merge(temp3, lent(testas3, 3), by="Label", all = T)
temp3 <- merge(temp3, lent(testas3, 4), by="Label", all = T)
temp3 <- merge(temp3, lent(testas3, 5), by="Label", all = T)
temp3 <- merge(temp3, lent(testas3, 6), by="Label", all = T)
temp3 <- merge(temp3, lent(testas3, 7), by="Label", all = T)
temp3 <- merge(temp3, lent(testas3, 8), by="Label", all = T)
#temp3 <- merge(temp3, lent(testas3, 9), by="Label", all = T)
#temp3 <- merge(temp3, lent(testas3, 10), by="Label", all = T)

temp3[is.na(temp3)] <- 0
names(temp3) <- c("OG", "Klast 1", "Klast 2", "Klast 3", "Klast 4", "Klast 5",
                  "Klast 6", "Klast 7", "Klast 8")#, "Klast 9", "Klast 10")
gal_klast_pas3 <- as.data.frame(t(temp3))
gal_klast_pas3 <- gal_klast_pas3[-1,]
names(gal_klast_pas3) <- c("OG 0", "OG 1", "OG 2", "OG 3", "OG 4", "OG 5", "OG 6",
                           "OG 7", "OG 8", "OG 9")
write.csv(gal_klast_pas3, "lent3.csv")

################ po t-SNE ##################
# Naudojant dendograma, identifikuojamas optimalus klasteriu skaicius
```

```
set.seed(67)
hc_tsne = hclust(d = dist(data_1, method = 'euclidean'), method = 'ward.D')
y_hc_tsne = cutree(hc_tsne, 8)

# Dendograma
# Nuspalvina dendograma
plot(hc_tsne)
rect.hclust(hc_tsne , k = 8, border = 2:6)

### APRASOMOJI
data_1_c_hc<- data1
data_1_c_hc$cluster<-y_hc_tsne

p1<- as.ggplot(heatmapping(data_1_c_hc, 1))
p2<- as.ggplot(heatmapping(data_1_c_hc, 2))
p3<- as.ggplot(heatmapping(data_1_c_hc, 3))
p4<- as.ggplot(heatmapping(data_1_c_hc, 4))
p5<- as.ggplot(heatmapping(data_1_c_hc, 5))
p6<- as.ggplot(heatmapping(data_1_c_hc, 6))
p7<- as.ggplot(heatmapping(data_1_c_hc, 7))
p8<- as.ggplot(heatmapping(data_1_c_hc, 8))
#p9<- as.ggplot(heatmapping(data_1_c_hc, 9))
#p10<- as.ggplot(heatmapping(data_1_c_hc, 10))

p1+ p2+p3+p4+p5+p6+p7+p8#+p9+p10

a_tsne<-matrix(unlist(data_1), nrow = 2000, ncol = 3)
fviz_cluster(list(data = a_tsne, cluster = y_hc_tsne), stand = FALSE, geom = "point")


######### dazniu lentele pagal kluster
testas4 <- data
testas4$clust <- y_hc_tsne

temp4 <- lent(testas4, 1)
temp4 <- merge(temp4, lent(testas4, 2), by="Label", all = T)
temp4 <- merge(temp4, lent(testas4, 3), by="Label", all = T)
temp4 <- merge(temp4, lent(testas4, 4), by="Label", all = T)
temp4 <- merge(temp4, lent(testas4, 5), by="Label", all = T)
temp4 <- merge(temp4, lent(testas4, 6), by="Label", all = T)
temp4 <- merge(temp4, lent(testas4, 7), by="Label", all = T)
temp4 <- merge(temp4, lent(testas4, 8), by="Label", all = T)
#temp4 <- merge(temp4, lent(testas4, 9), by="Label", all = T)
#temp4 <- merge(temp4, lent(testas4, 10), by="Label", all = T)

temp4[is.na(temp4)] <- 0
names(temp4) <- c("OG", "Klast 1", "Klast 2", "Klast 3", "Klast 4", "Klast 5",
                  "Klast 6", "Klast 7", "Klast 8")#, "Klast 9", "Klast 10")
gal_klast_pas4 <- as.data.frame(t(temp4))
gal_klast_pas4 <- gal_klast_pas4[-1,]
names(gal_klast_pas4) <- c("OG 0", "OG 1", "OG 2", "OG 3", "OG 4", "OG 5", "OG 6",
                           "OG 7", "OG 8", "OG 9")
write.csv(gal_klast_pas4, "lent4.csv")

############################
# HIERARCHINIS MAZAI KLASTERIU
############################
################ po t-SNE ##################
# Naudojant dendograma, identifikuojamas optimalus klasteriu skaicius
set.seed(67)
hc_tsne = hclust(d = dist(data_1, method = 'euclidean'), method = 'ward.D')
y_hc_tsne = cutree(hc_tsne, 3)

# Dendograma
# Nuspalvina dendograma
plot(hc_tsne)
rect.hclust(hc_tsne , k = 3, border = 2:6)

# Taskine diagrama
a_tsne<-matrix(unlist(data_1), nrow = 2000, ncol = 3)
fviz_cluster(list(data = a_tsne, cluster = y_hc_tsne), stand = FALSE, geom = "point")
```

```
### APRASOMOJI
data_1_c_hc<- data1
data_1_c_hc$cluster<-y_hc_tsne

p1<- as.ggplot(heatmapping(data_1_c_hc, 1))
p2<- as.ggplot(heatmapping(data_1_c_hc, 2))
p3<- as.ggplot(heatmapping(data_1_c_hc, 3))

p1+p2+p3



######### dazniu lentele pagal kluster
testas4 <- data
testas4$clust <- y_hc_tsne

temp4 <- lent(testas4, 1)
temp4 <- merge(temp4, lent(testas4, 2), by="Label", all = T)
temp4 <- merge(temp4, lent(testas4, 3), by="Label", all = T)


temp4[is.na(temp4)] <- 0
names(temp4) <- c("OG", "Klast 1", "Klast 2", "Klast 3")
gal_klast_pas4 <- as.data.frame(t(temp4))
gal_klast_pas4 <- gal_klast_pas4[-1,]
names(gal_klast_pas4) <- c("OG 0", "OG 1", "OG 2", "OG 3", "OG 4", "OG 5", "OG 6",
                           "OG 7", "OG 8", "OG 9")

write.csv(gal_klast_pas4, "lent5.csv")

############################
# GALUTINE LENTELE
############################

galut <- data[,785]
galut$kmeans <- y_kmeans
galut$kmeans_tsne <- y_kmeans2
galut$hc <- y_hc
galut$hc_tsne <- y_hc_tsne
```