Mājas darbs 2-7: distances, teksta algoritmi

1. Uzdevums - dimensiju redukcijas algoritmi ar dažādām distances metrikām

Izmantotie algoritmi - t-sne, UMAP. Izmantotās distances metrikas: *manhattan, euclidean, chebishev* t-sne gadījumā, *manhattan, euclidean, cosine* - UMAP.

Datu apstrāde

Kā parasti, ielasa datus, nogriež klases kolonnu un izveido atsevišķu vektoru datu punktu klasēm.

```
library(foreign)
trim <- function(df){
  df[,1:length(df[1,])-1]
}
colors <- function(df) {
  sapply(df, function(x) {
    if (x == "b") { "blue" }
    else { "red" }})
}
data <- read.arff("ionosphere.arff")
trimmed <- trim(data)
col_row <- colors(data[,length(data[1,])])
```

Funkcija grafiku zīmēšanai ar t-sne:

```
library(tsne)
plot.tsne <- function(df, distance="", perp=30, iter=400, class=NULL, plt=T, k=2, ret=F) {
  transformed <- tsne(df, k = k, perplexity = perp,
                      initial_dims = length(df[1,]), max_iter = iter)
  x <- transformed[,1]
  y <- transformed[,2]
  if (plt) {
    plot(x,y,col=class,xlab="comp1",
         ylab="comp2",
         main=sprintf("Distance metric: %s", distance))
  }
  if (ret) {
    transformed
  }
}
```

Funkcija grafiku zīmēšanai ar UMAP:

```
library(umap)
plot.umap <- function(df, config, class=NULL, distance="", knn=15, plt=T, ret=T) {
  transformed <- umap(df, config)
  x <- transformed$layout[,1]
  y <- transformed$layout[,2]
  if (plt) {
    plot(x,y,col=class,xlab="comp1",
         ylab="comp2",
         main=sprintf("Distance metric %s, knn = %d", distance, knn))
  }
  if (ret) {
    transformed
  }
}
```

t-sne realizācija piedāvā variantu datu kopu saņemt distanču matricas formā - citā veidā distances metriku mainīt šai funkcijai nevar:

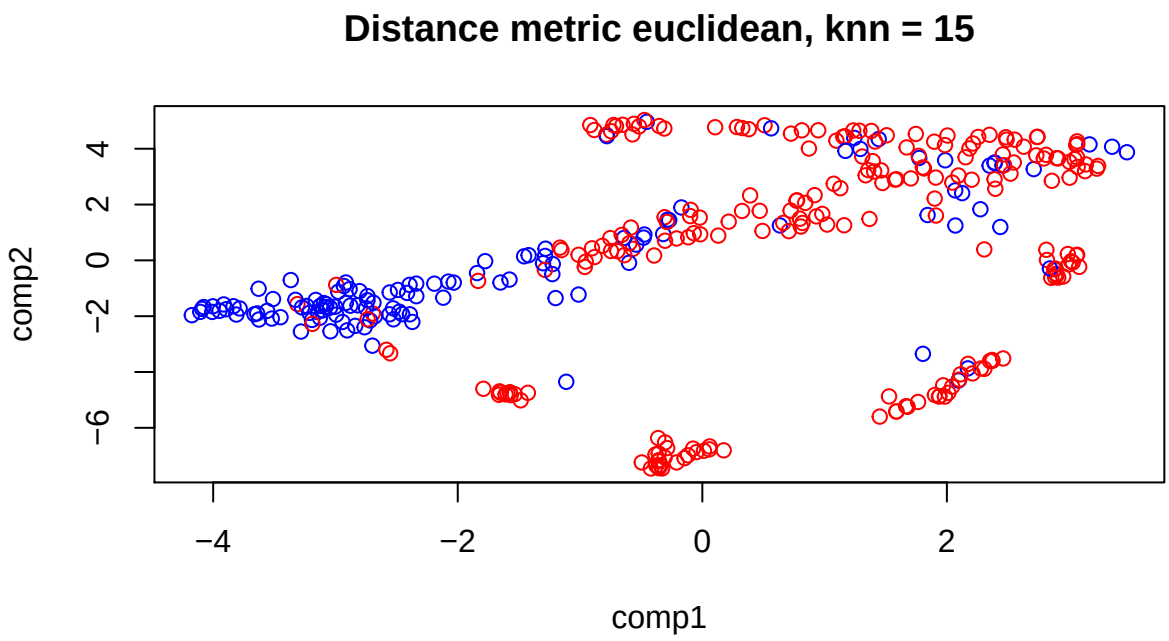
```
dist_euc <- dist(trimmed, method="euclidean")
dist_man <- dist(trimmed, method="manhattan")
dist_che <- dist(trimmed, method="maximum")
```

UMAP jāsagatavo konfigurācijas objekti:

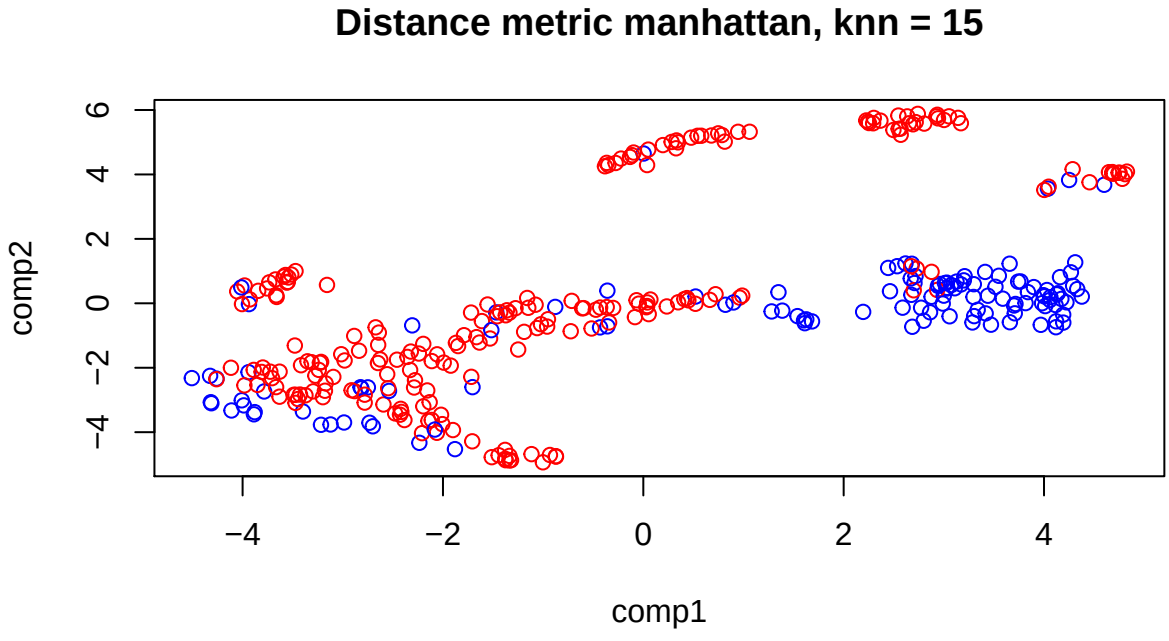
```
euclidean.config_15 <- umap.defaults
euclidean.config_30 <- umap.defaults
manhattan.config_15 <- umap.defaults
manhattan.config_30 <- umap.defaults
cosine.config_15 <- umap.defaults
cosine.config_30 <- umap.defaults
euclidean.config_15$metric <- "euclidean"
euclidean.config_30$metric <- "euclidean"
euclidean.config_30$n_neighbors <- 30
manhattan.config_15$metric <- "manhattan"
manhattan.config_30$metric <- "manhattan"
manhattan.config_30$n_neighbors <- 30
cosine.config_15$metric <- "cosine"
cosine.config_30$metric <- "cosine"
cosine.config_30$n_neighbors <- 30
```

UMAP rezultāti (pirmie, jo t-sne ir daudz lēnāki, un, dokumentu formatējot, praktiskāk ir likt ātrāk izpildāmus koda blokus pirmos):

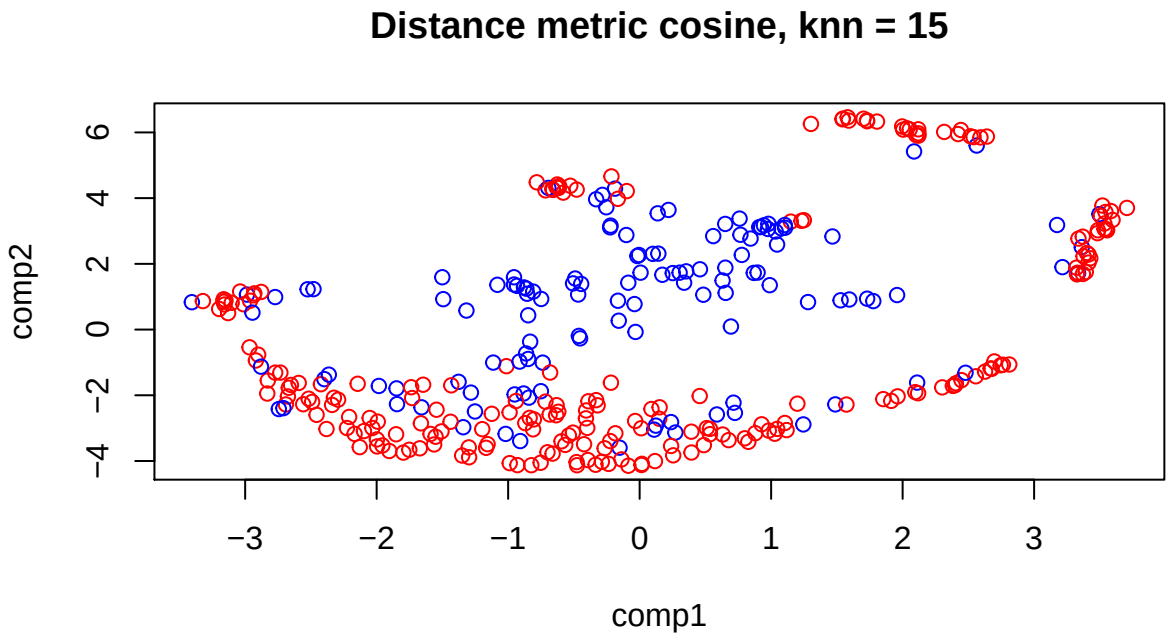
```
plot.umap(trimmed, euclidean.config_15, class=col_row, distance="euclidean")
```



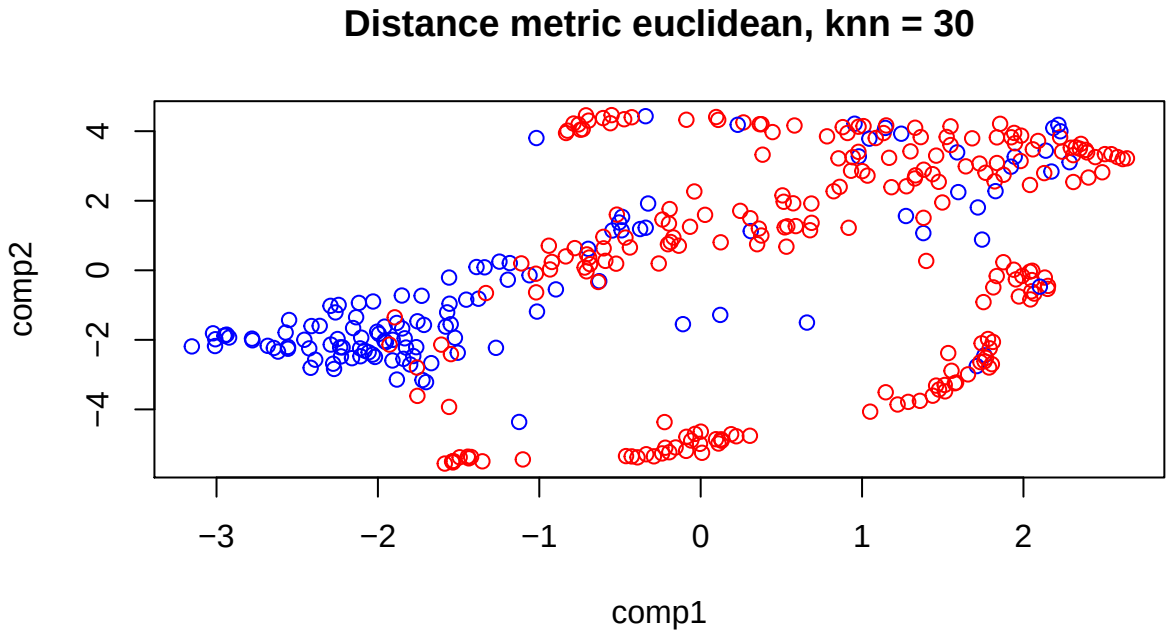

```
plot.umap(trimmed, manhattan.config_15, class=col_row, distance="manhattan")
```



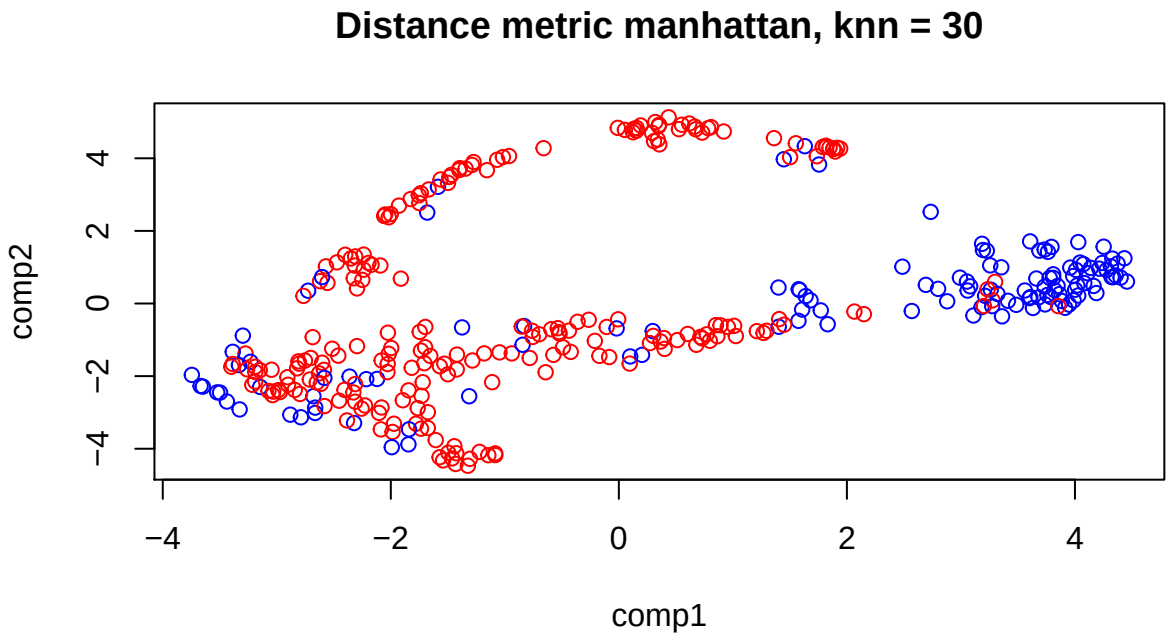
```
plot.umap(trimmed, cosine.config_15, class=col_row, distance="cosine")
```



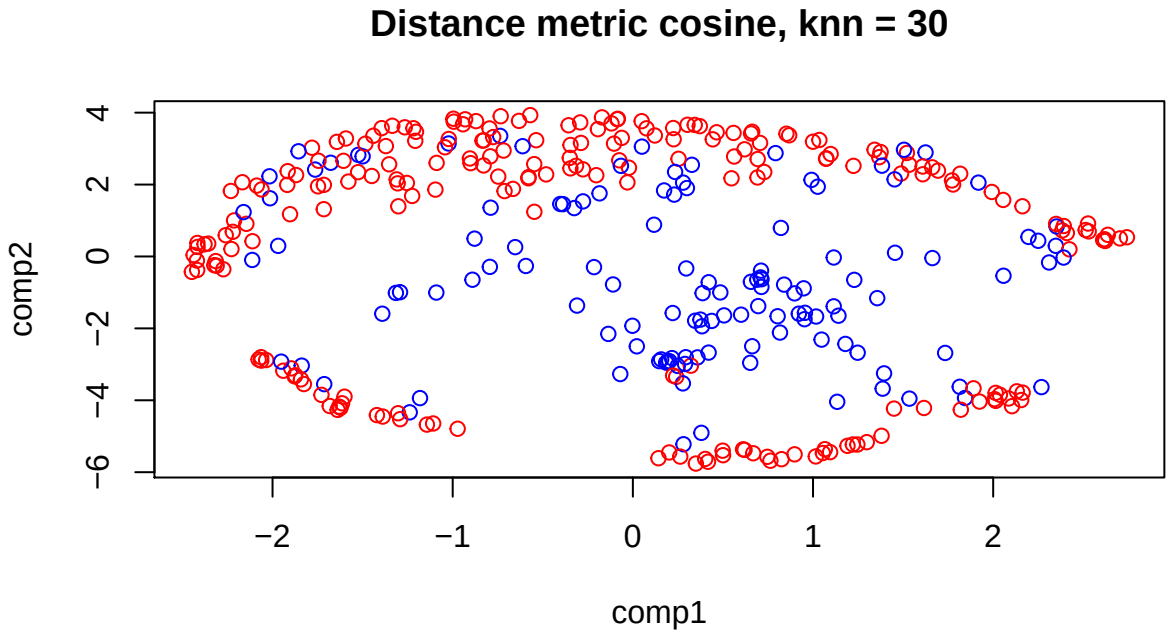

```
plot.umap(trimmed, euclidean.config_30, class=col_row, distance="euclidean", knn=30)
```



```
plot.umap(trimmed, manhattan.config_30, class=col_row, distance="manhattan", knn=30)
```

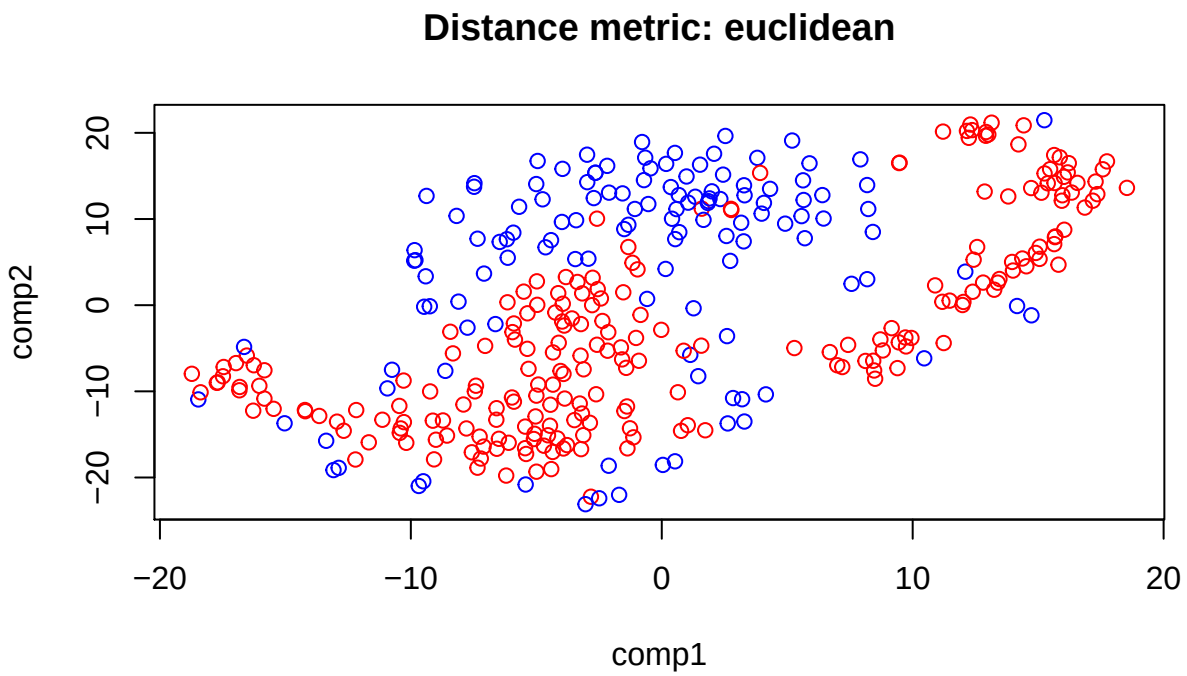



```
plot.umap(trimmed, cosine.config_30, class=col_row, distance="cosine", knn=30)
```

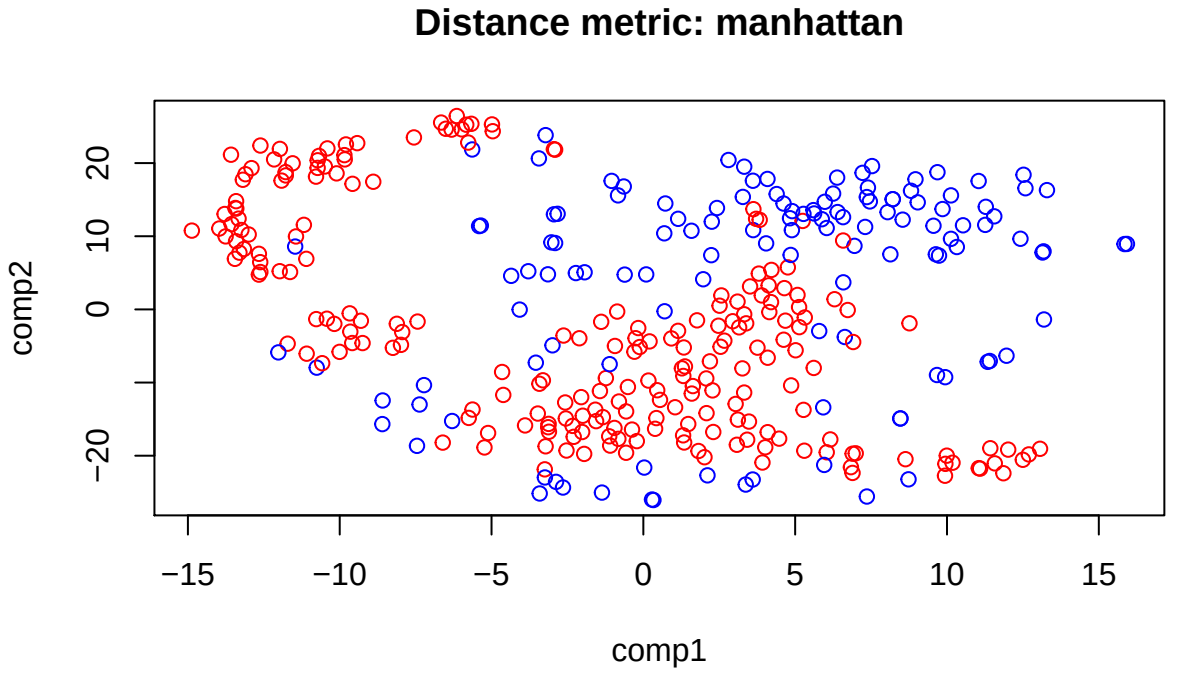


t-SNE rezultāti:

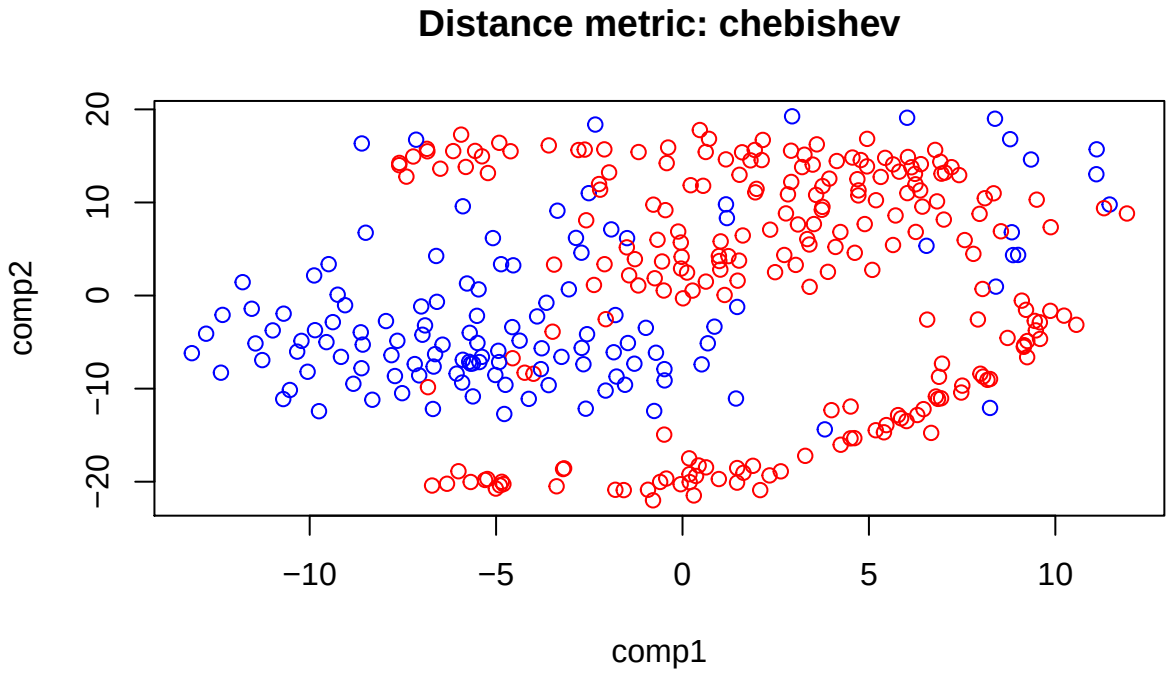
```
plot.tsne(dist_euc,30,class=col_row, distance="euclidean")
```




```
plot.tsne(dist_man,30,class=col_row, distance="manhattan")
```



```
plot.tsne(dist_che,30,class=col_row, distance="chebishev")
```



Secinājumi

t-SNE gadījumā pie Manhetenas distances zilās kopas punkti šķietami veido divus klasterus, pie Čebiševa distances - vienu difūzu, bet pie Eiklīda, kas ir savā veidā starp abiem Minkovska distances ekstrēmiem - vienu vai divus pārklājošos klasterus. Sarkanā kopa visos variantos varētu veidot divus iegarenus klasterus.

UMAP nebija pieejams Čebiševa distances mērs, taču vietā tika ņemta kosīnusu līdzība - t.i, vektoru savstarpējās projekcijas garums jeb skalārais reizinājums dalīts ar atsevišķo vektoru garumu reizinājumu. Pie diviem dažādiem tuvāko kaimiņu skaitiem novērotas līdzīgas sakarības - pēc kosinusiem zilā klase veido vienu klasteri, ko ielenc viens vai vairāki gredzenveida sarkanie klasteri. Pēc Manhetenas zilie punkti veido divus klasterus, sarkanie veido garenas grupas starp tiem. Pēc Eiklīda zilie punkti veido vienu klasteri ar daudziem izlēcējiem sarkanās grupas virzienā, sarkanā grupa veido vienu vai divus garenus klasterus.

2. Uzdevums - topic modeling praktiski pielietojumi