capstone_model_3_2 (hier)

2022-12-17

Helper packages

```
library(dplyr)       # for data manipulation
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)     # for data visualization
```

Modeling packages

```
library(cluster)     # for general clustering algorithms
library(factoextra)  # for visualizing cluster results
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

#import dataset

```
radiomics <- read.csv("radiomics_completedata.csv")
```

```
str(radiomics)
glimpse(radiomics)

# initial dimension
dim(radiomics)


#check for missing values
is.na(radiomics)
sum(is.na(radiomics))
na.omit(radiomics)
```

Scale data

```
scale(radiomics)
head(radiomics)
newdf1 = subset(radiomics, select = c(-Institution))
newdf1
```

```
#Determining Optimal Number of Clusters
set.seed(123)
```

Dissimilarity matrix

```
d <- dist(newdf1, method = "euclidean")
```

Hierarchical clustering using Complete Linkage

```
hc1 <- hclust(d, method = "complete" )
```

For reproducibility

```
set.seed(123)
```

Compute maximum or complete linkage clustering with agnes

```
hc2 <- agnes(newdf1, method = "complete")
```

Agglomerative coefficient

```
hc2$ac
## [1] 0.926775
```

methods to assess

```
m <- c( "average", "single", "complete", "ward")
names(m) <- c( "average", "single", "complete", "ward")
```

function to compute coefficient

```
ac <- function(x) {
  agnes(newdf1, method = x)$ac
}
```

get agglomerative coefficient for each linkage method

```
purrr::map_dbl(m, ac)
##   average    single  complete      ward
## 0.9139303 0.8712890 0.9267750 0.9766577
```

compute divisive hierarchical clustering

```
hc4 <- diana(newdf1)
```

Divise coefficient; amount of clustering structure found

```
hc4$dc
## [1] 0.9191094
```

Plot cluster results

```
p1 <- fviz_nbclust(newdf1, FUN = hcut, method = "wss",
                   k.max = 10) +
  ggtitle("(A) Elbow method")
p2 <- fviz_nbclust(newdf1, FUN = hcut, method = "silhouette",
                   k.max = 10) +
  ggtitle("(B) Silhouette method")
p3 <- fviz_nbclust(newdf1, FUN = hcut, method = "gap_stat",
                   k.max = 10) +
  ggtitle("(C) Gap statistic")
```

Display plots side by side

```
gridExtra::grid.arrange(p1, p2, p3, nrow = 1)
```

Construct dendorgram

```
hc5 <- hclust(d, method = "ward.D2" )
dend_plot <- fviz_dend(hc5)
dend_data <- attr(dend_plot, "dendrogram")
dend_cuts <- cut(dend_data, h = 8)
fviz_dend(dend_cuts$lower[[2]])
```

Ward's method

```
hc5 <- hclust(d, method = "ward.D2" )
```

Cut tree into 4 groups

```
sub_grp <- cutree(hc5, k = 8)
```

Number of members in each cluster

```
table(sub_grp)
```

Plot full dendogram

```
fviz_dend(
  hc5,
  k = 8,
  horiz = TRUE,
  rect = TRUE,
  rect_fill = TRUE,
  rect_border = "jco",
  k_colors = "jco",
  cex = 0.1
)


dend_plot <- fviz_dend(hc5)                # create full dendogram
dend_data <- attr(dend_plot, "dendrogram") # extract plot info
dend_cuts <- cut(dend_data, h = 70.5)      # cut the dendogram at designated height
# Create sub dendrogram plots
p1 <- fviz_dend(dend_cuts$lower[[1]])
p2 <- fviz_dend(dend_cuts$lower[[1]], type = 'circular')
```

Side by side plots

```r
gridExtra::grid.arrange(p1, p2, nrow = 1)
```