Podniková analytika (Cvičenie – deskriptívne DM - riešenie)

Úlohy na zhlukovanie

1. K-means (Iris)

- Načítajte dáta **iris** do premennej s názvom **iris2**

```
iris2 <- iris
```

- Vymažte z dát **iris2** cieľový atribút s názvom **Species**

```
iris2$Species <- NULL
```

- Pomocou algoritmu k-means rozdeľte dáta **iris2** do troch zhlukov (tried) a uložte tieto výsledky do premennej s názvom **kmeans.results**

```
(kmeans.result <- kmeans(iris2, 3))
```

- Vytvorte kontingenčnú tabuľku, ktorá bude porovnávať hodnoty **Species** z dát **iris** a získané klastre (zhluky) uložené v premennej **kmeans.results**

```
table(iris$Species, kmeans.result$cluster)
```

- Vytvorte graf, v ktorom budú znázornené atribúty **Sepal.Length** a **Sepal.Width** a vykreslené body rozdeľte farebne podľa získaných zhlukov

```
plot(iris2[c("Sepal.Length", "Sepal.Width")], col = kmeans.result$cluster)
```

- Doplňte do grafu body centroidov pre jednotlivé zhluky

```
points(kmeans.result$centers[,c("Sepal.Length", "Sepal.Width")], col = 1:3, pch = 8, cex = 2)
```

2. K-means (Mtcars)

- Načítajte dáta **mtcars** do premennej s názvom **mtcars1**

```
mtcars1 = mtcars
```

- Vyberte z dát **mtcars1** len stĺpce s názvom **hp** a **drat**

```
mtcars1 = mtcars1[,c(4:5)]
```

- Pomocou algoritmu k-means rozdeľte dáta **mtcars1** do dvoch zhlukov (tried) a uložte tieto výsledky do premennej s názvom **kmeans.results1**

```
(kmeans.result1 <- kmeans(mtcars1, 2))
```

- Vytvorte kontingenčnú tabuľku, ktorá bude porovnávať hodnoty **vs** z dát **mtcars** a získané klastre (zhluky) uložené v premennej **kmeans.results1**

```
table(mtcars$vs, kmeans.result1$cluster)
```

- Vytvorte graf, v ktorom budú znázornené atribúty **hp** a **drat** a vykreslené body rozdeľte farebne podľa získaných zhlukov

```
plot(mtcars1[c("hp", "drat")], col = kmeans.result1$cluster)
```

- Doplňte do grafu body centroidov pre jednotlivé zhluky

```
points(kmeans.result1$centers[, c("hp", "drat")], col = 1:2, pch
= 8, cex = 2)
```

3. Hierarchické zhlukovanie (Iris)

- Nastavte seedovanie na hodnotu 2835

```
set.seed(2835)
```

- Vytvorte premennú **idx,** ktorá bude obsahovať indexy 40 náhodných riadkov z dát **iris**

```
idx <- sample(1:dim(iris)[1], 40)
```

- Vytvorte premennú **irisSample,** ktorá bude obsahovať iba čísla riadkov uložené v premennej **idx**

```
irisSample <- iris[idx, ]
```

- Odstráňte z dát **irisSample** cieľový atribút **Species**

```
irisSample$Species <- NULL
```

- Do premennej s názvom **hc** uložte výsledky hierarchického zhlukovania z dát **irisSample,** metódu tohto zhlukovania nastavte tak, že algoritmus bude brať vzdialenosť medzi zhlukmi ako *priemer* vzdialenosti bodov v jednom zhluku a bodov v inom zhluku

```
hc <- hclust(dist(irisSample), method = "ave")
```

- Vykreslite tieto zhluky (klastre) pomocou dendrogramu

```
plot(hc, hang = -1, labels = iris$Species[idx])
```

- Orežte vykreslený dendrogram na úrovne troch zhlukov

```
rect.hclust(hc, k = 3)
```

- Pridajte do dát **irisSample** nový stĺpec s názvom **groups**, ktorý bude obsahovať ID získaných zhlukov

```
irisSample$groups <- cutree(hc, k = 3)
```

4. Hierarchické zhlukovanie (Mtcars)

- Uložte do premennej s názvom **mtcars2** všetky riadky dát **mtcars** a stĺpce **mpg** a **qsec**

```
mtcars2 <- mtcars[,c(1,7)]
```

- Do premennej s názvom **hc2** uložte výsledky hierarchického zhlukovania z dát **mtcars2,** metódu tohto zhlukovania nastavte tak, že algoritmus bude brať vzdialenosť medzi zhlukmi ako *priemer* vzdialenosti bodov v jednom zhluku a bodov v inom zhluku

```
hc2 <- hclust(dist(mtcars2), method = "ave")
```

- Vykreslite tieto zhluky (klastre) pomocou dendrogramu

```
plot(hc, hang = -1, labels = mtcars$cyl)
```

- Orežte vykreslený dendrogram na úrovne troch zhlukov

```
rect.hclust(hc, k = 3)
```

- Pridajte do dát **mtcars** nový stĺpec s názvom **groups**, ktorý bude obsahovať ID získaných zhlukov

```
mtcars$groups <- cutree(hc, k = 3)
```

Úlohy na vytvorenie asociačných pravidiel

5. **Asociačné pravidlá (Titanic)**

- Stiahnite dáta **Titanic** z webovej adresy http://web.tuke.sk/fei-cit/butka/res/titanic.csv a uložte ich do premennej s názvom **titanic**

```
download.file("http://web.tuke.sk/fei-
cit/butka/res/titanic.csv", destfile = "titanic.csv")
titanic = read.csv("titanic.csv", header = TRUE, sep = ",")
```

- Vytvorte premennú **rules,** ktorá bude obsahovať asociačné pravidlá získané z týchto dát pomocou algoritmu *apriori*, minimálnu dĺžku pravidla nastavte na 2, minimálnu podporu pravidiel na 0,005, minimálnu spoľahlivosť na 0,8, sledovaný atribút (pravá strana pravidla) bude obsahovať možnosti atribútu **Survived**

```
library(arules)
rules <- apriori(titanic, control = list(verbose=F),
         parameter = list(minlen=2, supp=0.005, conf=0.8),
         appearance = list(rhs=c("Survived=No","Survived=Yes"),
         default="lhs"))
```

- Získané pravidlá usporiadajte podľa ukazovateľa **Lift** a následne ich vypíšte

```
rules.sorted <- sort(rules, by="lift")
inspect(rules.sorted)
```

- Vymažte zo získaných pravidiel tie, ktoré sú redundantné a na záver ich vypíšte

```
subset.matrix <- is.subset(rules.sorted, rules.sorted)
subset.matrix[lower.tri(subset.matrix, diag = T)] <- NA
redundant <- colSums(subset.matrix, na.rm = T) >= 1
which(redundant)
rules.pruned <- rules.sorted[!redundant]
inspect(rules.pruned)
```

6. **Asociačné pravidlá (AdultUCI)**

- Načítajte dáta **AdultUCI**

```
data("AdultUCI")
```

- Do premennej s názvom **fact** uložte hodnoty *TRUE, FALSE* podľa toho, či daný stĺpec je dátový typ *faktor*

```
fact <- sapply(AdultUCI, is.factor)
```

- Z dát **AdultUCI** vytvorte jej podmnožinu s názvom **AdultUCI1,** ktorá bude obsahovať len stĺpce dátového typu faktor

```
AdultUCI1 = AdultUCI[ , fact]
```

- Vytvorte premennú **rules1,** ktorá bude obsahovať asociačné pravidlá získané z týchto dát pomocou algoritmu *apriori*, minimálnu dĺžku pravidla nastavte na 4, minimálnu podporu pravidiel na 0,01, minimálnu spoľahlivosť na 0,68, sledovaný atribút (pravá strana pravidla) bude obsahovať možnosti atribútu **income**

```
rules1 <- apriori(AdultUCI1, control = list(verbose=F),
          parameter  =  list(minlen=4,  supp=0.01,  conf=0.68),
          appearance=list(rhs=c("income=large","income=small"),
          default="lhs"))
```

- Získané pravidlá usporiadajte podľa ukazovateľa **Lift** a následne ich vypíšte

```
rules.sorted <- sort(rules1, by="lift")
inspect(rules.sorted)
```

- Vymažte zo získaných pravidiel tie, ktoré sú redundantné a na záver ich vypíšte

```
subset.matrix <- is.subset(rules.sorted, rules.sorted)
subset.matrix[lower.tri(subset.matrix, diag = T)] <- NA
redundant <- colSums(subset.matrix, na.rm = T) >= 1
which(redundant)
rules.pruned <- rules.sorted[!redundant]
inspect(rules.pruned)
```