SPRAWOZDANIE 5

Klasteryzacja

Krzyszczuk Michał

7 stycznia 2018

Wczytanie danych oraz ich podział

```
data <- read.csv("songs.csv", header = TRUE)
data$year <- NULL
data$songtitle <- NULL
data$artistname <- NULL
data$songID <- NULL
data$artistID <- NULL
data$timesignature <- NULL
data$timesignature_confidence <- NULL
temp <- split(data, sample(rep(1:2, c(5049,2525))))
learning_data <- temp$`1`
test_data <- temp$`2`
learning_data_copy <- learning_data
test_data_copy <- test_data
learning_data_copy$Top10 <- NULL
test_data_copy$Top10 <- NULL
preproc <- preProcess(learning_data_copy)
learning_data_norm <- predict(preproc, learning_data_copy)
test_data_norm <- predict(preproc, test_data_copy)
```

Klasteryzacja

```
centers = 3
km <- kmeans(learning_data_norm, centers)
kmkcca <- as.kcca(km, learning_data_norm)
```

```
## Found more than one class "kcca" in cache; using the first, from namespace 'flexclust'
```

```
## Also defined by 'kernlab'
```

```
## Found more than one class "kcca" in cache; using the first, from namespace 'flexclust'
```

```
## Also defined by 'kernlab'
```

```
clustertrain <- predict(kmkcca)
```

```
## Found more than one class "kcca" in cache; using the first, from namespace 'flexclust'
## Also defined by 'kernlab'
```

```
clustertest <- predict(kmkcca, newdata=test_data_norm)
datalist = c()
x <- c(1:centers)
for (i in x)

{
regtrain <- subset(learning_data, clustertrain==i)
regtest <- subset(test_data, clustertest==i)
model <- glm(Top10 ~ loudness + tempo + tempo_confidence + key + key_confidence +
```

```
                      energy + pitch + timbre_0_min + timbre_0_max + timbre_1_min + timbre_1_max + timbr
                      timbre_2_max + timbre_3_min + timbre_3_max + timbre_4_min + timbre_4_max + timbre_
                      timbre_5_max + timbre_6_min + timbre_6_max + timbre_7_min + timbre_7_max + timbre_
                      timbre_8_max + timbre_9_min + timbre_9_max + timbre_10_min + timbre_10_max + timbr
                      timbre_11_max, data = regtrain, family = 'binomial')
prediction1 <- predict(model, regtest, type="response")
accuracy1 <- table(ifelse(prediction1 > 0.75, 1, 0), regtest$Top10)
cluster <- sum(diag(accuracy1))/sum(accuracy1)
datalist[[i]] <-cluster
}
prob3 <-mean(datalist)

centers_list <- c(2,4,5)
propabilities <- c()
for(j in centers_list)
{
  km <- kmeans(learning_data_norm, j)
kmkcca <- as.kcca(km, learning_data_norm)
clustertrain <- predict(kmkcca)
clustertest <- predict(kmkcca, newdata=test_data_norm)
datalist = c()
x <- c(1:j)
for (i in x)

{
regtrain <- subset(learning_data, clustertrain==i)
regtest <- subset(test_data, clustertest==i)
model <- glm(Top10 ~ loudness + tempo + tempo_confidence + key + key_confidence +
                      energy + pitch + timbre_0_min + timbre_0_max + timbre_1_min + timbre_1_max + timbr
                      timbre_2_max + timbre_3_min + timbre_3_max + timbre_4_min + timbre_4_max + timbre_
                      timbre_5_max + timbre_6_min + timbre_6_max + timbre_7_min + timbre_7_max + timbre_
                      timbre_8_max + timbre_9_min + timbre_9_max + timbre_10_min + timbre_10_max + timbr
                      timbre_11_max, data = regtrain, family = 'binomial')
prediction1 <- predict(model, regtest, type="response")
accuracy1 <- table(ifelse(prediction1 > 0.75, 1, 0), regtest$Top10)
cluster <- sum(diag(accuracy1))/sum(accuracy1)
datalist[[i]] <-cluster
propabilities[[j]]<-mean(datalist)
}
}
propabilities[[2]]
```

```
## [1] 0.8525399
```

```
prob3
```

```
## [1] 0.8493175
```

```
propabilities[[4]]
```

```
## [1] 0.8539954
```

```
propabilities[[5]]
```

```
## [1] 0.8415013
```

Klaster z największą wartością prawdopodobieństwa zwracanego przez funkcję predict zostanie utworzony dla liczby klastrów równej 4. Prawdopodobieństwo to wynosi:

```
propabilities[[4]]
```

```
## [1] 0.8539954
```