Clustering

Authors:

Jack Asaad
Andrew Sen
Atmin Sheth
Neo Zhao

Date:

10/9/2022

Introduction

Clustering algorithms are unsupervised machine learning algorithms whose goal is to create/discover groupings in data. By finding groupings in a dataset, it is possible to get new insights into the nature of the data.

In this notebook, we will demonstrate three different clustering algorithms. The dataset we will use describes shoppers' webpage activity in a given online session and if that activity was ultimately converted into a purchase.

Attributes of dataset:

- Administrative: number of administrative page visits in the session
- Administrative_Duration: total seconds spent on administrative pages in the session
- Informational: number of informational page visits in the session
- Information_Duration: total seconds spent on informational pages in the session
- ProductRelated: number of related product pages visited in the session
- ProductRelated_Duration: total seconds spend on related product pages in the session
- BounceRates: average bounce rate of all pages visited in the session (bounce rate is the percentage of visitors that enter the site through that page and immediately exit)
- ExitRates: average exit rate of all pages visited in the session (exit rate is the percentage of visits to a page that were the last page visited in a session)
- PageValues: average page value of all pages visited in the session (page value is the average transaction value sessions that visited that page)
- SpecialDay: closeness of day on which session took place to a holiday (range is [0, 1])
- Month: month in which session tooke place
- OperatingSystems: number indicating operating system of user
- Browser: number indicating browser of user
- Region: number indicating user's region
- TrafficType: number indicating type of traffic
- VisitorType: whether user is new (New_Visitor) or returning (Returning_Visitor)
- Weekend: indicates whether session was on a weekend (TRUE or FALSE)
- Revenue: indicates whether session ended in a transaction (TRUE or FALSE)

Data citation: Sakar, C.O., Polat, S.O., Katircioglu, M. et al. Neural Comput & Applic (2018). [Web Link] (https://doi.org/10.1007/s00521-018-3523-0)

Reading and Cleaning Data

```
# reading data
df <- read.csv("data/online_shoppers_intention.csv")

# converting text columns to numerical columns

# VisitorType
# new = 0; returning = 1
df$VisitorType[df$VisitorType == "New_Visitor"] <- 0
df$VisitorType[df$VisitorType == "Returning_Visitor"] <- 1

# Month
df$Month <- match(df$Month, month.abb)

# Weekend
df$Weekend[df$Weekend == "FALSE"] <- 0
df$Weekend[df$Weekend == "TRUE"] <- 1
```

To get a better idea of the value generated by each session, we will create a new column GeneratedPageValue. The values of this column will be PageValues multiplied by the total number of pages visited if the session ended with a transaction, and 0 otherwise.

```
# Replacing TRUE/FALSE with 1/0
df$Revenue[df$Revenue == "FALSE"] <- 0
df$Revenue[df$Revenue == "TRUE"] <- 1

# calculating page value generated
df$GeneratedPageValue <- df$Revenue * df$PageValues * (df$Administrative + df$Informational +
df$ProductRelated)
```

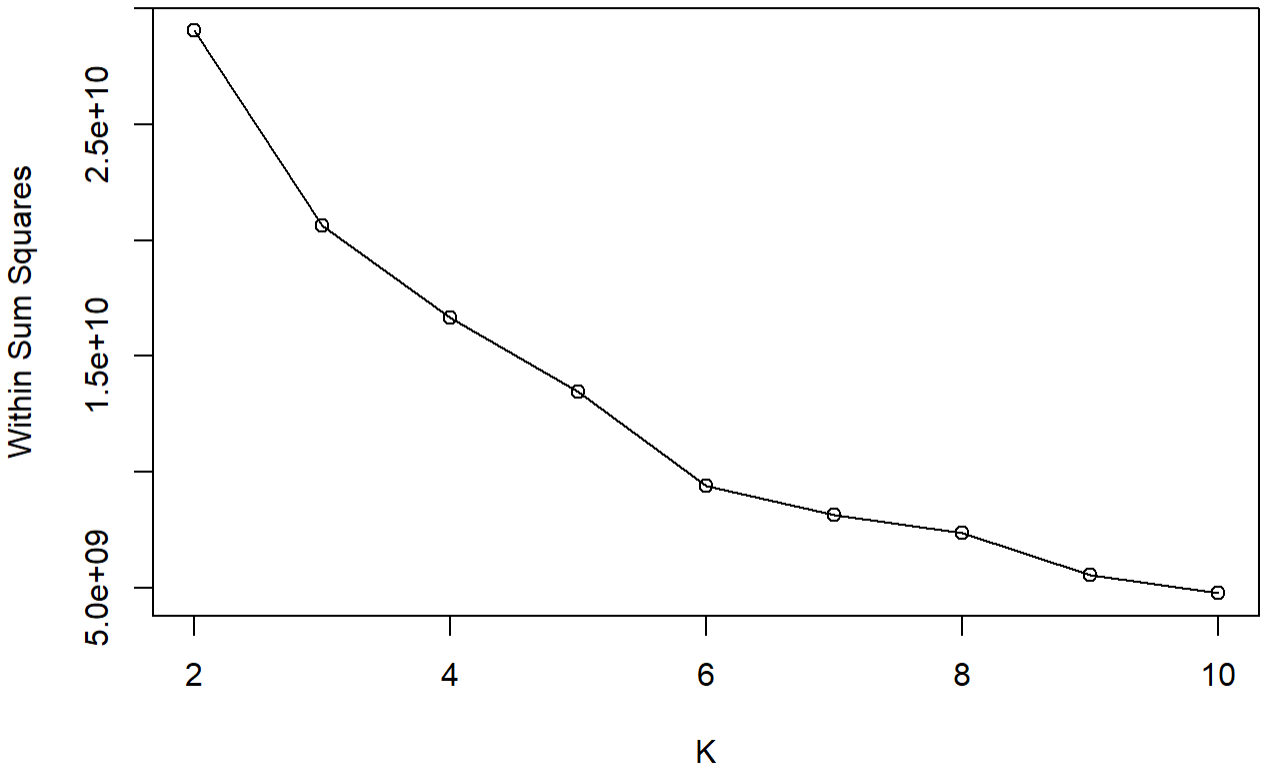
For each of the following clustering demonstrations, we will form clusters based on the ProductRelated_Duration and GeneratedPageValue columns. We can use the sizes of the resulting clusters to speculate how time spent browsing products may translate to actual purchases.

Finding Optimal Number of Clusters

Before demonstrating K-Means and Hierarchical clustering, we will need to choose how many clusters we want to make from the data. This choice can be arbitrary, but we can also use K-Means with varying numbers of clusters to determine an optimal amount.

```
plot_withinss <- function(df, max_clusters){
  withinss <- rep(0, max_clusters - 1)
  for (i in 2:max_clusters){
    set.seed(1234)
    withinss[i] <- sum(kmeans(df, i)$withinss)
  }
  plot(2:max_clusters, withinss[2:max_clusters], type="o", xlab="K", ylab="Within Sum Square
s")
}

# passing only ProductRelated_Duration and Revenue columns
plot_withinss(df[, c(6, 19)], 10)
```



We want to find an “elbow” in the graph where increasing the number of clusters does not significantly change the results. In this case, there appears to be an elbow at k=6, so that is the number of cluster we will choose to make.

K-Means Clustering

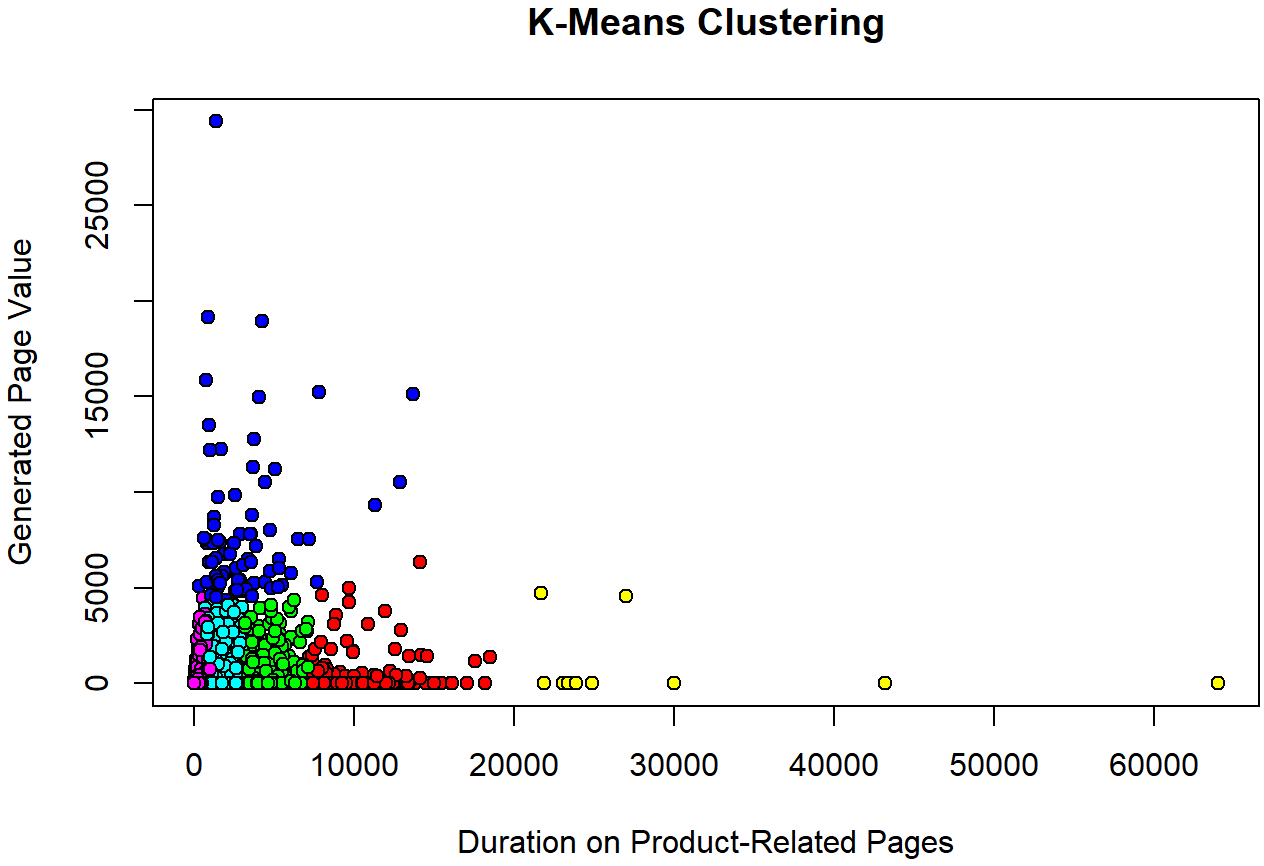
```
set.seed(1234)

# making 6 clusters from ProductRelated_Duration and Revenue
km <- kmeans(df[, c(6, 19)], 6, nstart=20)
summary(km)
```

```
##              Length Class  Mode
## cluster      12330  -none- numeric
## centers         12  -none- numeric
## totss            1  -none- numeric
## withinss         6  -none- numeric
## tot.withinss     1  -none- numeric
## betweenss        1  -none- numeric
## size             6  -none- numeric
## iter             1  -none- numeric
## ifault           1  -none- numeric
```

```
# making list of colors for scatterplot
color_palette <- c("red", "green", "blue", "yellow", "magenta", "cyan", "orange", "white", "b
lack")

# coloring scatterlplot with cluster classifications
plot(df$ProductRelated_Duration, df$GeneratedPageValue, pch=21, bg=color_palette [unclass(k
m$cluster)], main="K-Means Clustering", xlab="Duration on Product-Related Pages", ylab="Gener
ated Page Value")
```



Hierarchical Clustering

```
library(flexclust)
```

```
## Loading required package: grid
```

```
## Loading required package: lattice
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

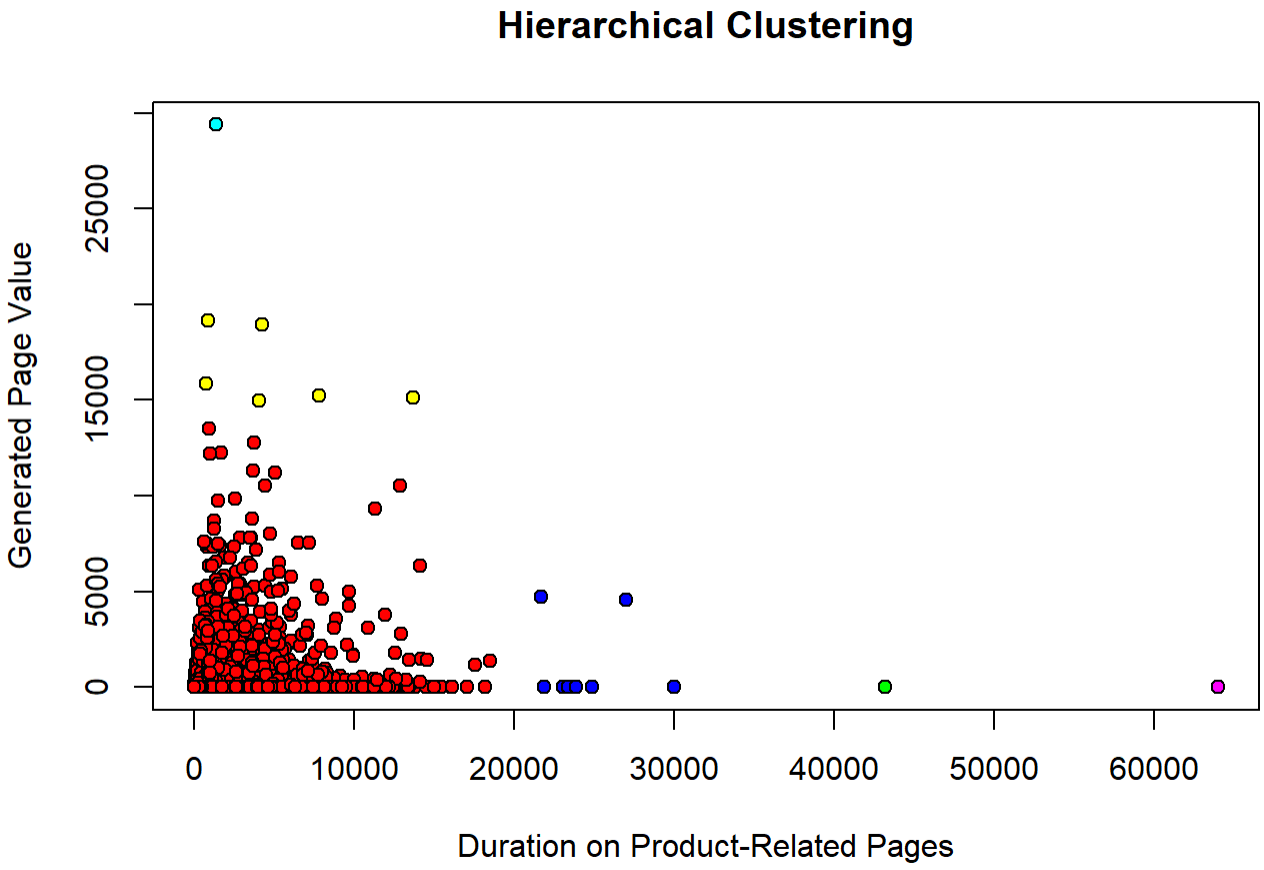
```
# creating distance function
df.scaled <- scale(df[, c(6, 19)])
d <- dist(df.scaled)

# creating hierarchy of clusters
hc <- hclust(d, method="average")
summary(hc)
```

```
##             Length Class  Mode
## merge       24658  -none- numeric
## height      12329  -none- numeric
## order       12330  -none- numeric
## labels          0  -none- NULL
## method          1  -none- character
## call            3  -none- call
## dist.method     1  -none- character
```

```
# cutting tree to get 5 clusters
memb <- cutree(hc, k=6)

plot(df$ProductRelated_Duration, df$GeneratedPageValue, pch=21, bg=color_palette [unclass(mem
b)], main="Hierarchical Clustering", xlab="Duration on Product-Related Pages", ylab="Generate
d Page Value")
```



Model Based Clustering

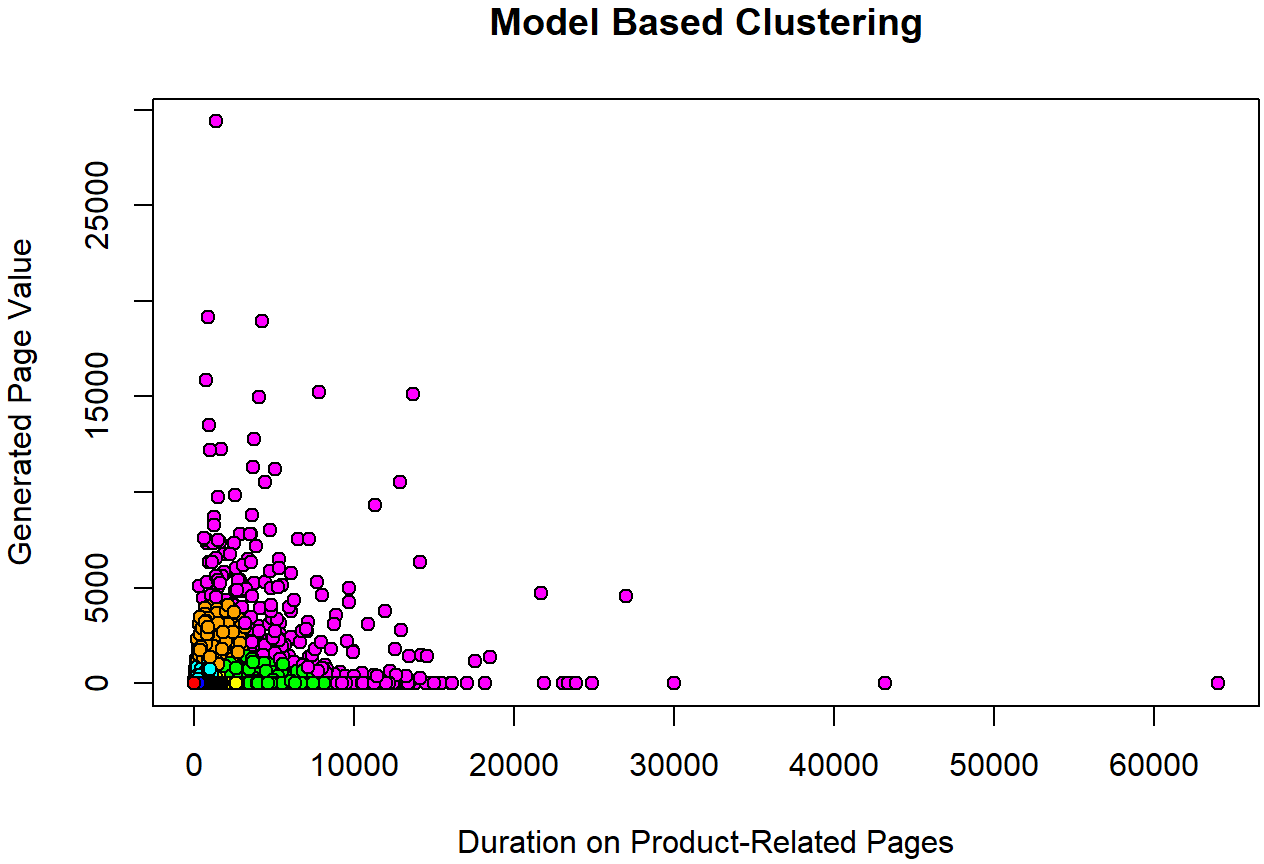
```
library(mclust)
```

```
## Package 'mclust' version 5.4.10
## Type 'citation("mclust")' for citing this R package in publications.
```

```
# finding clusters with model-based approach
mc <- Mclust(df[, c(6, 19)])
summary(mc)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVE (ellipsoidal, equal orientation) model with 9 components:
##
##  log-likelihood     n df       BIC       ICL
##         -168671 12330 45 -337765.9 -339883.4
##
## Clustering table:
##    1    2    3    4    5    6    7    8    9
## 2423  893 2695 1150  286  609  324 2248 1702
```

```
plot(df$ProductRelated_Duration, df$GeneratedPageValue, pch=21, bg=color_palette [unclass(m
c$classification)], main="Model Based Clustering", xlab="Duration on Product-Related Pages",
ylab="Generated Page Value")
```



Conclusion

Every model produced drastically different clusters. For both K-Means and Hierarchical clustering, we chose to create 6 clusters. As we can see in the scatterplots, the algorithms were very different in how they divided the data into 6 groups. K-Means appears to have combined outliers into larger clusters while the hierarchical model placed outliers in their own singleton groups. In fact, K-Means overall produced more evenly-sized groups compared to the hierarchical model.

The model based clustering algorithm determined that 9 clusters was optimal, so its graph differs from the other two in that area. We can also see that the clusters were not determined by a naive distance function because the clusters appear to be slightly mixed in the scatterplot, unlike the other two models. This makes it slightly harder to interpret. It appears that the clusters produced by the model based algorithm are more evenly sized than the one created by the hierarchical algorithm but are still less even than K-Means.

As per the hierarchical model, we can see that most users can be grouped together as spending little time browsing products before making a purchasing decision, and most of the site's page value can be attributed to this group of users. Both the K-Means and hierarchical models grouped together users that spend a lot of time browsing but never buy anything. However, the model based algorithm chose to group many of these users with the large cluster in the bottom left of the plot, indicating that perhaps both kinds of users are similar to some extent.