```r
url = "http://www2.informatik.uni-freiburg.de/~cziegler/BX/BX-CSV-Dump.zip"
download.file(url, destfile = "data.zip")
dir.create("data")
unzip("data.zip",exdir = "data")

files = paste0("data/",list.files("data"))

ratings = read.csv(files[1], sep = ";")
books = read.csv(files[2], sep = ";")
users = read.csv(files[3], sep = ";")

rm(files, url)

library(dplyr)
glimpse(books)

set.seed(1234)
categories = c("Action and Adventure","Classic","Detective and Mystery","Fantasy")
books$category = sample( categories, nrow(books), replace=TRUE, prob=c(0.25, 0.3, 0.25, 0.20))
books$category = as.factor(books$category)

rm(categories)

books$ISBN = paste0("Isbn.",books$ISBN)
users$User.ID = paste0("User.",users$User.ID)
ratings$ISBN = paste0("Isbn.",ratings$ISBN)
ratings$User.ID = paste0("User.",ratings$User.ID)

library(ggplot2)

ratings %>%
  group_by(Book.Rating) %>%
  summarize(cases = n()) %>%
  ggplot(aes(Book.Rating, cases)) + geom_col() +
  theme_minimal() + scale_x_continuous(breaks = 0:10)

ratings = ratings[ratings$Book.Rating!= 0, ]

ratings %>%
  group_by(Book.Rating) %>%
  summarize(cases = n()) %>%
  ggplot(aes(Book.Rating, cases)) + geom_col() +
  theme_minimal() + scale_x_continuous(breaks = 0:10)

ratings_sum = ratings %>%
  group_by(User.ID) %>%
  count()

summary(ratings_sum$n)

user_index = ratings_sum$User.ID[ratings_sum$n>4]

users = users[users$User.ID %in% user_index, ]
ratings = ratings[ratings$User.ID %in% user_index, ]
books = books[books$ISBN %in% ratings$ISBN,]
```

```
rm(ratings_sum, user_index)

library(cluster)

books_distance = books[,c("ISBN","Book.Author","Publisher")]

# Convert variables to
factors
books_distance[,1] <- as.factor(books_distance[,1])
books_distance[,2] <- as.factor(books_distance[,2])
books_distance[,3] <- as.factor(books_distance[,3])

# Calculate Gower Distance
dissimilarity = daisy(books_distance, metric = "gower")

matrix(ncol = 115246, nrow = 115246)

library(dplyr)

book_feature = books[1:10000,c("Book.Author","Publisher","category")]

# convert to factors
book_feature[,1] <- as.factor(book_feature[,1])
book_feature[,2] <- as.factor(book_feature[,2])
book_feature[,3] <- as.factor(book_feature[,3])

dissimilarity = daisy(book_feature, metric = "gower", weights = c(2,0.5,1))
dissimilarity = as.matrix(dissimilarity)

row.names(dissimilarity)<-  books$ISBN[1:10000]
colnames(dissimilarity)<- books$ISBN[1:10000]

dissimilarity[15:20,15:20]

user_id = "User.1167"

user_books = ratings %>%
  filter(User.ID == user_id & ISBN %in% books$ISBN[1:10000]) %>%
  arrange(desc(Book.Rating))

head(user_books,10)

library(tidyr)

books$ISBN = as.character(books$ISBN)
selected_books = user_books[ ,c("ISBN", "Book.Rating")]

recomendar = function(selected_books, dissimilarity_matrix,
                      books, n_recommendations = 5){

  selected_book_indexes = which(colnames(dissimilarity_matrix) %in% selected_books$ISBN)


  results = data.frame(dissimilarity_matrix[, selected_book_indexes],
                      recommended_book = row.names(dissimilarity_matrix),
                      stringsAsFactors = FALSE)


  recomendaciones = results %>%
    pivot_longer(cols = c(-"recommended_book") , names_to = "readed_book",
                 values_to = "dissimilarity") %>%
      left_join(selected_books, by = c("recommended_book" = "ISBN"))%>%
    arrange(desc(dissimilarity)) %>%
    filter(recommended_book != readed_book) %>%
    filter(!is.na(Book.Rating) ) %>%
    mutate(
      similarity = 1 - dissimilarity,
      weighted_score = similarity * Book.Rating) %>%
    arrange(desc(weighted_score)) %>%
    filter(weighted_score>0) %>%
    group_by(recommended_book) %>% slice(1) %>%
    top_n(n_recommendations, weighted_score)  %>%
    left_join(books, by = c("recommended_book" = "ISBN"))

  return(recomendaciones)
}

recomendaciones = recomendar(selected_books, dissimilarity, books)
recomendaciones

visualizar_recomendacion = function(recomendation,
                                    recommended_book, image, n_books = 5){

  if(n_books > nrow(recomendation)) {n_books = nrow(recomendation)}

  plot = list()

  dir.create("content_recommended_images")
  for(i in 1:n_books){
    # Create dir & Download the images
    img = pull(recomendation[i,which(colnames(recomendation) == image)])
    name = paste0("content_recommended_images/",i,".jpg")
    suppressMessages(
      download.file(as.character(img), destfile = name ,mode = "wb")
    )

    # Assign Objetc
    plot[[i]] = rasterGrob(readJPEG(name))
  }

    do.call(marrangeGrob, args = list(plot, ncol = n_books, nrow = 1, top=""))

}

visualizar_recomendacion(recomendaciones, "recommended_book","Image.URL.M")


user_item = ratings %>%
  top_n(10000) %>%
  pivot_wider(names_from = ISBN,values_from = Book.Rating) %>%
  as.data.frame()

row.names(user_item) = user_item$User.ID
user_item$User.ID = NULL

user_item = as.matrix(user_item)

user_item[1:5,1:5]

sum(is.na(user_item)) /  ( ncol(user_item) * nrow(user_item) )

cos_similarity = function(A,B){
  num = sum(A *B, na.rm = T)
  den = sqrt(sum(A^2, na.rm = T)) * sqrt(sum(B^2, na.rm = T))
  result = num/den

  return(result)
}

item_recommendation = function(book_id, rating_matrix = user_item, n_recommendations = 5){

  book_index = which(colnames(rating_matrix) == book_id)

  similarity = apply(rating_matrix, 2, FUN = function(y)
                     cos_similarity(rating_matrix[,book_index], y))

  recommendations = tibble(ISBN = names(similarity),
                              similarity = similarity) %>%
    filter(ISBN != book_id) %>%
    top_n(n_recommendations, similarity) %>%
    arrange(desc(similarity))

  return(recommendations)

}

recom_cf_item = item_recommendation("Isbn.0446677450")
recom_cf_item

recom_cf_item = recom_cf_item %>%
  left_join(books, by = c("ISBN" = "ISBN"))

visualizar_recomendacion(recom_cf_item[!is.na(recom_cf_item$Book.Title),],
                         "ISBN",
                         "Image.URL.M"
                         )

user_recommendation = function(user_id, user_item_matrix = user_item,
                              ratings_matrix = ratings,
                              n_recommendations = 5,
                              threshold = 1,
                              nearest_neighbors = 10){

  user_index = which(rownames(user_item_matrix) == user_id)
  similarity = apply(user_item_matrix, 1, FUN = function(y)
                     cos_similarity(user_item_matrix[user_index,], y))

  similar_users = tibble(User.ID = names(similarity),
                         similarity = similarity) %>%
    filter(User.ID != user_id) %>%
    arrange(desc(similarity)) %>%
    top_n(nearest_neighbors, similarity)


  readed_books_user = ratings_matrix$ISBN[ratings_matrix$User.ID == user_id]

  recommendations = ratings_matrix %>%
    filter(
      User.ID %in% similar_users$User.ID &
      !(ISBN %in% readed_books_user)) %>%
    group_by(ISBN) %>%
    summarise(
      count = n(),
      Book.Rating = mean(Book.Rating)
    ) %>%
    filter(count > threshold) %>%
    arrange(desc(Book.Rating), desc(count)) %>%
    head(n_recommendations)

  return(recommendations)

}

recom_cf_user = user_recommendation("User.99", n_recommendations = 20)
recom_cf_user

recom_cf_user = recom_cf_user %>%
  left_join(books, by = c("ISBN" = "ISBN"))

visualizar_recomendacion(recom_cf_user[!is.na(recom_cf_user$Book.Title),],
                         "ISBN","Image.URL.M")
```