**IMDb Rating System**

Anil Poonai

Question

How can I group movies together in order to have a better recommendation system?

Summary

I wanted to find a way to visualize and model how movies are rated and see what type of movies are similarly rated to others. I used a K means clustering method in order to explore the dataset of an IMDb movie ratings sheet. It ended up revealing that 70% of the ratings could be correctly grouped and judged off of factors that do not involve any people or studios.

Data Description

The dataset is from Kaggle and has over 14,000 observations, it includes columns such as the title of the movie, IMDb Rating, duration, year release, number of nominations and wins, number of photos and news articles and number of genres. It also has binary variables for each type of genre such as western, action, drama etc. I treated these as dummy variables to make the dataset more readable.

Data Manipulation

I changed the column data types to numeric for ones that were mismatched as well as deleted some empty columns that existed in the dataset. I then filtered out the dataset by the type of observation (movie, tv show, game, etc.) to movies. After this I made a replicated data frame and adjusted this one by setting all of the columns with non-numeric data types NULL, effectively making them deleted. As for data that had missing values, I just set them to be omitted from everything. I also set the number of user reviews, rating count and the rating columns to NULL as the rating is what I need to match to the clusters and the numbers shouldn’t be relevant as they are too variable with every movie and don’t have an effect of the rating that movie gets ideally. I also scaled all of the values so that they could be in the same range as other columns and not have a specific column have too much of an influence. I also made two more replicated data frames with one only having the column of IMDb Ratings and the other including the Rating, user reviews and rating count.

Data Summary

I used two clusters that had the lowest total within-cluster sum of squares as they had the least difference with in the clusters and each other cluster, which goes hand in hand with the proportion of the sum squares between the means of the clusters to the global means of the data frame over the sum squares of each point to the global mean of the data frame. Which was approximately 85% so both clusters are organized well with this data frame. Also had a rand index of about .7 for both clusters with one of them being slightly higher than the other.

Model Building

I found my model by realizing that I did not want to make a prediction but just wanted to see how the dataset can be arranged as it is. I wanted to be able categorize them into certain categories and see how accurate the ratings were to the clusters. I ended up on K means as most supervised methods were meant to predict future values and K means clusters are easy to scale and implement as well fairly accurate. Since this isn’t a prediction there won’t be a need for a training and testing set.

Final Model

My final model is going to be a k means clustering with a center of 90 as that had the best response from the rand index when compared to the ratings. It also has less of ranged variability when compared to my other k means clustering model with 240 centers as shown below. The clusplot function says 100% but that’s because the function has a higher variability of points than what’s needed.

K means with 90 centers:

A screenshot of a cell phone

Description automatically generated

K means with 240 centers:

A screenshot of a cell phone

Description automatically generated

The K means clustering with 90 centers actually includes the rating, user reviews and review counts, which actually ended up being the opposite of what I thought earlier in the Data Manipulation section. I should also mention that those two graphs has component 2 as just the IMDb rating but when it is the entire dataset the variability ends up dropping significantly.

K means with 90 centers:

A close up of a map

Description automatically generated

K means with 240 centers:

A close up of a map

Description automatically generated

Model Fit

I am fairly satisfied with my model as I wasn’t expecting it to be insanely accurate as some obvious variables are missing that would have to be incorporated for that, such as studios and people involved as well as budget, etc. This model really seems to fail on grouping movies with low rating counts. I’m also disappointed that it didn’t have a higher rand index score, as it doesn’t seem to useful for grouping the movies now.

Model Interpretation

This model helps finds similar movies 70% of the time, which is useful but not good enough to be utilized.

Conclusion

If needed it can work as a recommendation system but more data is needed in order to have a higher reliability. This will work 70% of the time according to how accurate it rates the clusters to the rating but I wonder how high it can get with more data than just the extrinsic characteristics.

Appendix

1. Data Source: <https://www.kaggle.com/orgesleka/imdbmovies> or you can go to <https://github.com/DevonARP/STA-3000--IMDb-Rec-System> and click on the imdb.csv file.
2. The R code is also there in the STA3000\_IMDb.r file but I will list it here as well:

library(tidyverse)

library(plyr)

library(factoextra)

library(NbClust)

options(max.print = 8000)

#Replace your file location here

imdb <- read\_csv("C:/Users/poona/Desktop/3000/imdb.csv")

View(imdb)

dim(imdb)

str(imdb)

#Changeing Datatypes

imdb$nrOfPhotos = as.numeric(imdb$nrOfPhotos)

imdb$nrOfNominations = as.numeric(imdb$nrOfNominations)

imdb$nrOfWins = as.numeric(imdb$nrOfWins)

imdb$duration = as.numeric(imdb$duration)

imdb$ratingCount = as.numeric(imdb$ratingCount)

imdb$imdbRating = as.numeric(imdb$imdbRating)

#Getting rid of empty columns

imdb$X45 = NULL

imdb$X46 = NULL

imdb$X47 = NULL

imdb$X48 = NULL

str(imdb)

#Getting movies only

unique(imdb$type)

practical = filter(imdb,imdb$type=='video.movie')

#Getting rid of characteristic columns and omitting empty data

practical$type=NULL

practical$url=NULL

practical$wordsInTitle=NULL

practical$fn=NULL

practical$title=NULL

practical$tid=NULL

practical=practical[0:10]

practical = na.omit(practical)

#Scaled it

practical=scale(practical)

practical = as.data.frame(practical)

view(practical)

summary(practical)

#Another replicated data frame with three less columns

practical1=practical

practical1$imdbRating=NULL

practical1$nrOfUserReviews=NULL

practical1$ratingCount=NULL

#Kmeans with 90 centers

p1 = kmeans(practical, centers=90,iter.max = 30)

summary(p1)

str(p1)

p1$cluster

p1$centers

p1$totss

p1$withinss

p1$tot.withinss

p1$betweenss

p1$betweenss/p1$totss

p1

count(practical$imdbRating)

table(practical$imdbRating,p1$cluster)

#Graph on Entire Dataset

library(cluster)

clusplot(practical,p1$cluster)

#Kmeans with 240 centers

p2= kmeans(practical1,center=240, iter.max=30)

p2$tot.withinss

clusplot(practical1,p2$cluster)

clusplot(practical1,p1$cluster)

#Rand Index

library(fossil)

rand.index(practical$imdbRating,p2$cluster)

rand.index(practical$imdbRating,p1$cluster)

#Data frame with one column

practical2=practical

practical2$ratingCount=NULL

practical2$duration=NULL

practical2$year=NULL

practical2$nrOfWins=NULL

practical2$nrOfNominations=NULL

practical2$nrOfPhotos=NULL

practical2$nrOfNewsArticles=NULL

practical2$nrOfUserReviews=NULL

practical2$nrOfGenre=NULL

#Graph of only IMDb rating

clusplot(practical2,p1$cluster)

clusplot(practical2,p2$cluster)