# Rating of amazon movies

## Final Project Report – 2017 Spring

Submitted by: Chenshi Liu, Minghao Guo, Xuan Yang, Yinchen Jiang
BIA 658 - Social Network Analysis
Instructor: Prof. Mai Feng

# Abstract

The recent boom of online media supplier triggered the enthusiasm of commenting and rating movies (including video types, played in cinemas as well as on TVs) online, from 2013 the number of ratings increased dramatically. Some media suppliers such as Netflix have provided a huge amount of data about different movies' ratings, the ratings reflect people's thoughts about the movies, therefore, there is a necessity for us to do a research about the network inside the commenting and the rating. The purpose of this paper is to do clustering to a group of movies on Amazon, analyzing on the ratings in different clusters, and to propose directions for future research.



Key words: Amazon, movies, clustering, rating

## 1. Introduction

Viewing ratings and comments are beneficial for us before watching films. There are several benefits, for example, viewing the comments is helpful for consumers to have general ideas about the films, usually, good movies always get high marks from people, if the marks about the films are low, people will not choose to go to watch the films in the cinemas or buying the videos, it will help customers to save money and time. What is more, viewing movies' comments allow people to have a deep understanding of films. Sometimes, the directors prefer to express their feelings in some small details, it is not easy for viewers to find all the details, in this way, the ratings of movies from different people are beneficial for understanding the films.

Although analyzing these data are helpful for people when choosing watching which movies, there are also several limitations of this research. Some types of movies such as historical or scientific movies are not popular, so the viewers are few or people tend to give low comments to these movies, the comments of these movies may not valuable to people. In addition, because small items ID which is the comments of films equal or less than 100 accounts for the large part of the data (about 96%), maybe the comments and ratings are not so persuasive.

The purpose of this paper is to perform an analysis using Excel, R and Python on the rating of Amazon movies, the movies include video types, played in the cinemas as well as on TV programs and so on. We intend to cluster the movies based on the associativity among these movies, to analyze the relationship between ratings and the clusters, finally to bring up some advice for the Amazon movie recommendation systems and Amazon users. This research is significant to Amazon, because to predict how a user will respond to a product, it is essential to understand the views' preference and the properties of the products.

## 2. Data Observation

### 2.1 Data source and overview

Our data set was obtained from Stanford Network Analysis Platform (SNAP), a general purpose network analysis and graph mining library. It is about the ratings of all movies and

TVs on Amazon from 1997 to 2014. It includes 4607047 observations with 4 variables: UserID, ItemID, Rating and Occur time. The head of data set shows in Appendix [2.1].

- UserID stands for one specific user who rated on movie. One user could on different movies.
- ItemID stands for one specific movie which was rated. One movie could be rated by different users.
- Rating ranks from 1 to 5. 1 is the lowest and 5 is highest.
- Occur Time is a time stamp that marks when users commented on a moive.

2.2 Data Processing

- Average rating

  We grouped the data by each item_ID, then calculated the amount of ratings for each item_ID as "count", the average of the ratings as "avg_rate" and the standard deviation of the ratings as "sd". [Figure 1]

| | itemID | count | avg_rate | sd |
|---|---|---|---|---|
| 1 | 0000143502 | 1 | 5.000000 | *NA* |
| 2 | 0000143529 | 1 | 5.000000 | *NA* |
| 3 | 0000143561 | 1 | 2.000000 | *NA* |
| 4 | 0000143588 | 2 | 5.000000 | 0.0000000 |
| 5 | 0000589012 | 29 | 4.103448 | 1.4963009 |

Figure 1 count and average rating of each movie

- Group data set

  As the original data set was huge and the analysis required smaller and more precise data set. We divided the original data set into 4 groups depending on the number of ratings of each movie. We thought that a larger amount of ratings represented greater popularity of a movie.

  - small_itemID: ratings amount <= 100
  - middle_itemID: 100 < ratings amount <= 1000
  - large_itemID: 1000< ratings amount <= 2000
  - X_itemID: ratings amount > 2000

- Find the edges among movies

When one user rated on both movie A and movie B, we counted 1 for the "weight" between them. Therefore, "weight" between two movies defined the amount of the users who comments on both of them. By doing so, we generated weighted edges among the movies. The edge somehow represents the "associativity" between two movies. The edges show in table below [Table 2].

| | itemID_1 | itemID_2 | weight |
|---|---|---|---|
| 1 | 0310263662 | 079070546X | 24 |
| 2 | 0310263662 | 0790729628 | 46 |
| 3 | 0310263662 | 0790745399 | 61 |
| 4 | 0310263662 | 0792833171 | 39 |
| 5 | 0310263662 | 0793906091 | 83 |

Table 2 Edges among movies

2.3 Data observation

We calculated the total amounts of movies and the average of total ratings for each group: small_itemID, middle_itemID, large_itemID and X_itemID. [Table 3]

| Group | Amounts of movies | Average ratings |
|---|---|---|
| Samll_itemID | 192549 | 4.0111 |
| Middle_itemID | 8072 | 4.2207 |
| Large_itemID | 296 | 4.2584 |
| X_itemID | 106 | 4.2258 |
| All | 200941 | 4.0199 |

Table 3 Group Observation

According to the table above, the amounts of movies in Samll_itemID is much larger than the those in other groups. The average of total ratings of large_itemID group is the largest while that of large_itemID group is the smallest.

Also, we drew a histogram to show the frequencies of different average ratings (the average rating of each movie but not each group) [Figure 4].

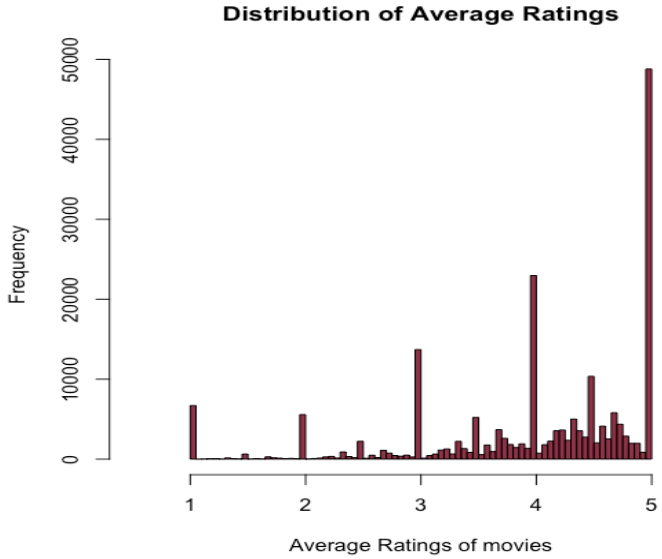



Figure 4 Frequencies of different average ratings

In the histogram, the five abnormal peeks on integers are caused by the large amounts of movies which only have one rating (occupy almost 70% of the total movies). Excluding the abnormal data, we could find that the frequencies of average ratings in range from 4 to 5 are much larger than others. That means, most users were more likely to give the rating in range from 4 to 5.

**3. Network: clustering and analysis on ratings**

Depending on the edges generated above, we drew the network of the movies (item_ID) using R. Each node in the network represents a movie (item_ID). The edges are generated in section 2.2 and each edge represents the "associativity" between each pair of movies. Because the igraph package in R could not adjust the lengths of edges by their weights, we just set a threshold to separate the weights to 0 or 1. Any weights smaller than 50 could be reset as 0 and that means the related edges do not exist. And weights larger than 50 could be reset as 1 and that means the related edges do exist.

Also, movies with different average rating have different colors. The movies with average rating higher than or equal to 4 are colored red while those with average rating smaller than 4 are colored blue.

3.1 Network of large_itemID group

Because the numbers of movies in small_itemID group and middle_itemID group are too large and the process of generating “associativity” edges costs too much time, we only produce the network of large_itemID group, X_itemID and the aggregation of them. Here we report the network of large_itemID group [Figure 5] and the other two would show in Appendix [3.1].

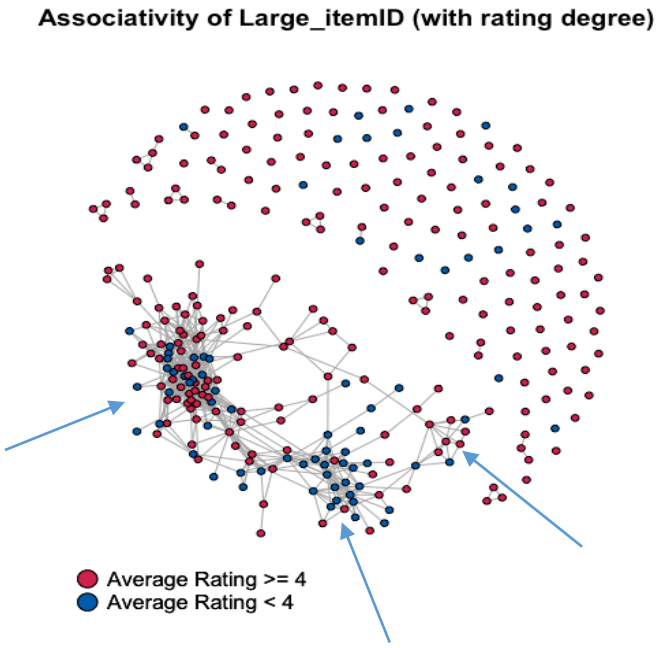



Figure 5 Network of large_itemID group

Depending on the network above, we could separate the movies in large_itemID group into two large clusters, a small cluster, many isolate nodes and some small and single-linked groups. In the left cluster, there are more high rating movies than low rating movies. In the middle cluster, low rating movies are much more than the high rating movies. And in the small cluster on the right, high rating movies are more than low rating movies. In addition, among the isolate nodes and single-linked groups, there are much more high rating movies than low rating movies.

3.2 Networks for high/low rating movies

We then produced the network diagrams for the movies with high average ratings (average rating >= 4) only and the movies with low average ratings (average rating < 4) only. [Figure 6]

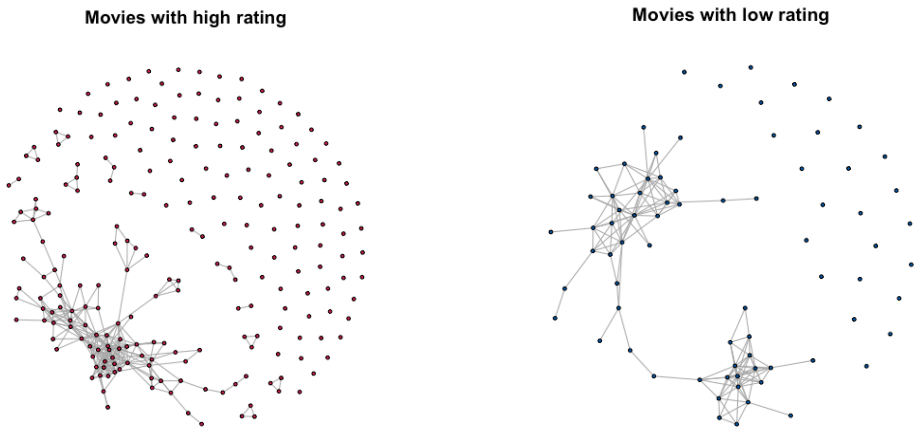



Figure 6 Networks for movies with high ratings (left) and movies with low ratings (right)

In the figure 6 above, we could find that the low rating movies could obviously be separated into two clusters while there are no distinct separated clusters among high rating movies.

3.3 Some attributes of the three networks

We calculated the average degree, average eccentricity, associativity degree and some other attributes of the three networks above using R. [Table 7]

| | All movies in large_itemID group | High rating movies | Low rating movies |
|---|---|---|---|
| Average degree | 4.939189 | 2.823529 | 3.893333 |
| Max degree | 48 | 30 | 16 |
| Min degree | 0 | 0 | 0 |
| Average eccentricity | 3.949324 | 3.334842 | 5.96 |
| Max eccentricity | 10 | 11 | 11 |
| Min eccentricity | 0 | 0 | 0 |

| | | | |
|---|---|---|---|
| Transitivity | 0.4997347 | 0.5665116 | 0.4850669 |

Table 7 Some attributes of the three networks

Some basic concepts:

- Degree: In graph theory, the degree of a vertex of a graph is the number of edges incident to the vertex, with loops counted twice.
- Average degree: Average degree is closely related to the density of a network.
- Eccentricity: The eccentricity of a vertex is the greatest geodesic distance between and any other vertex. It can be thought of as how far a node is from the node most distant from it in the graph.
- Transitivity: The transitivity of a graph is closely related to the clustering coefficient of a graph, as both measure the relative frequency of triangles. In graph science, a clustering coefficient is a measure of the degree to which nodes in a graph tend to cluster together.

Conclusions of these attributes

According to the table, we could find that:

- The max degree of high rating movies is larger than that of low rating movies. And the average degree of high rating movies is smaller than that of low rating movies. Considering the Figure 6, the reason might be that the number of high rating movies is much larger than that of low rating movies. (high rating movies: 221, low rating movies: 75). There are much more isolate nodes in the network of high rating movies. However, at the same time, in the cluster of high rating movies, nodes are more connected with each other than that in the clusters of low rating movies.
- The average eccentricity of low rating movies is larger than that of high rating movies. The high rating movies are more likely to centralize.
- The transitivity of high rating movies is a little larger than that of low rating movies. That means, the high rating movies are more likely to cluster together.

## 4. K-Means Clustering

In section 3, we transformed the weight to 1 or 0 and clustered the movies in large_itemID group. However, we still wanted to try to do the clustering by the original data of weight. Because the weight varied in a large range and transforming them to 1 or 0 might have a bad effect. On another hand, clustering in section 3 could not help us obtain the specific "item_IDs" of each cluster.

Then, we decided to use K-Means clustering to cluster the movies.

4.1 Basic concepts

- K-means Clustering: k-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. It clusters by minimizing the within-cluster sum of squares between the centers and other nodes in each cluster (WCSS).
- K-Mean Center: The center of the cluster. In the cluster, each feature of the center is calculated as the mean of the corresponding features of all the nodes in the cluster.
- Euclidean distance (The distance between two nodes in this projection):
  Calculated as

$$D_{xy} = ||\{x_1, x_2, x_3,.....x_n\} - \{y_1, y_2, y_3,.....y_n\}||$$
$$= [(x_1-y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 ......+(x_n - y_n)^2]^{0.5}$$

4.2 Preparation of features

We generated the similarity matrix of all 296 movies (item_ID) in the large_itemID group. By doing so, we obtained 296 features for each movie, those are, the associativity weights between this movie and all movies in the group (including itself). Then, we changed the numbers on the diagonal to the max number in the matrix but not on the diagonal. Because the numbers on the diagonal represent the associativity weights between the movies and themselves (the same as the numbers of the ratings for the movies). The numbers could be much larger than the other numbers and have a bad effect on the K-Meaning cluster. In addition, we added the features together for each movie and excluded the movies whose sum is 3 standard deviations away from the mean. By doing so we excluded one outliner data. After all, the data was transformed to a matrix which was appropriate for the K-Means Clustering. A part of the data after processing shows in the Appendix [4.1].

4.3 Result of the clustering

We set the K = 3 and separated the movies into 3 clusters. The numbers of movies in the three clusters are 124, 87 and 84. The specific item_IDs in the clusters would show in the Appendix [4.2].

4.4 TSNE and visualization

T-Distributed Stochastic Neighbor Embedding (TSNE) is a technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets. Compared with PCA and MDS, TSNE could guarantee that closest neighbors of each data point remains its neighbors after the projection.

We used TSNE to reduce the 296 features to 2 features dimension and then use K-Means Clustering again and generate the visualization of the clustering. [Figure 8]

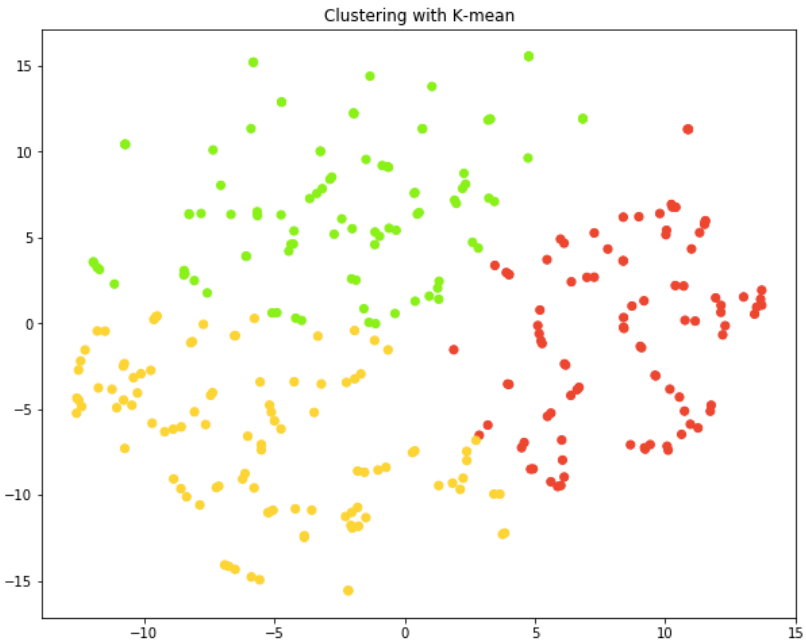



Figure 8 Visualization of K-Mean Clustering

Although we made full use of the original data and could obtained the specific "item_IDs" of each cluster, it seemed that the clustering here didn't perform well.

**5. Conclusion**

In the paper we focused on the movies whose numbers of ratings are in range from 1000 to 2000. Such movies have enough viewers and ratings which means they are qualified enough and would attract certain audiences. At the same time, they are not the top popular and there might be a lot of audiences who haven't watched them. Thus we think these movies deserves to be analyzed on and then recommended.

According to the networks in section 3, for movies in large_itemID group, we concluded that:

- Those movies could be clustered into three clusters and there is a lot of isolate items or single-linked groups at the same time.
- Among the isolate movies, there are much more high rating movies than low rating movies. Thus for audiences, it means that the most popular movies are not necessarily the best. Some less popular one would be a supervise.
- High rating movies are more likely to cluster together while low rating movies could be separated into two distinct clusters. For Amazon, it could recommend movies to users by what they have watched before. Then, it would be more likely to recommend the high rating ones. However, at the same time, it has better avoid the low rating clusters.

According to the networks in section 4, we could make full use of the associativity weight among movies and obtain specific movie IDs by K-Means Clustering. However, the cluster s obtained by this method were not distinct enough.

**6. Deficiencies**

- After spending a lot of time, we still haven't found the way to produce the network in which the length of the edges could be decided by the weight. As a result, we could not do the clustering very well with visualization.
- We haven't studied on the movies in the small_itemID group and the middle_itemID, which occupy the largest part of all movies.
- In the K-Means Clustering, we haven't gone into details and found the best K (how many clusters should we generate)

- As the clusters are not quite distinct and we cannot figure out why the movies are clustered in that way, we could not give direct and effective recommendations to Amazon and its users

**7. Future Work**

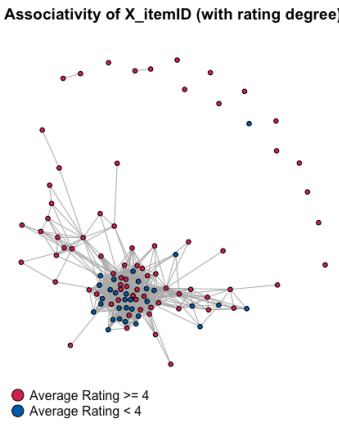
- Optimize the algorithm and try to obtain the associativity weights among movies in small_itemID group and the middle_itemID.
- On SNAP there is a related dataset which contains the comments for each rating. In the next step we could dig out more information from the comments.
- We could spend more time on the K-Means Clustering and get a clear understanding of the algorithm. Then we could find the appropriate value of K and then optimize the clustering.
- We could try to make it clear why the movies are clustered in that way. And by doing so we could optimize our methods of analysis and clustering and then give better recommendation.

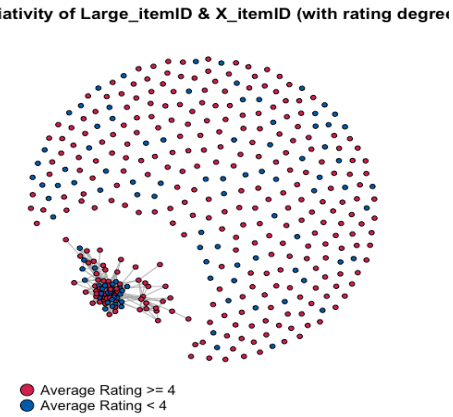
Appendix

2.1 Part of the original data

| UserID | ItemID | Rating | Occur Time |
|---|---|---|---|
| A3R5OBKS7OM2IR | 143502 | 5 | 1358380800 |
| A3R5OBKS7OM2IR | 143529 | 5 | 1380672000 |
| AH3QC2PC1VTGP | 143561 | 2 | 1216252800 |
| A3LKP6WPMP9UKX | 143588 | 5 | 1236902400 |
| AVIY68KEPQ5ZD | 143588 | 5 | 1232236800 |
| A1CV1WROP5KTTW | 589012 | 5 | 1309651200 |
| AP57WZ2X4G0AA | 589012 | 2 | 1366675200 |
| A3NMBJ2LCRCATT | 589012 | 5 | 1393804800 |
| A5Y15SAOMX6XA | 589012 | 2 | 1307404800 |
| A3P671HJ32TCSF | 589012 | 5 | 1393718400 |

3.1 Networks of X_itemID group and the aggregation of large_itemID group and X_itemID group.




4.1 A part of the data after processing in section 4.

| | 0767002652 | 0780623134 | 0782002064 | 0783211856 | 0783216084 | 0783222734 | 0783222955 | 0783239408 | 0783241038 | 0784010188 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0780623134** | 1 | 283 | 37 | 39 | 47 | 27 | 33 | 57 | 19 | 36 | ... |
| **0782002064** | 3 | 37 | 283 | 23 | 29 | 22 | 20 | 24 | 14 | 15 | ... |
| **0783211856** | 0 | 39 | 23 | 283 | 48 | 30 | 27 | 29 | 17 | 28 | ... |
| **0783216084** | 0 | 47 | 29 | 48 | 283 | 71 | 25 | 39 | 31 | 52 | ... |
| **0783222734** | 0 | 27 | 22 | 30 | 71 | 283 | 17 | 28 | 24 | 32 | ... |
| **0783222955** | 0 | 33 | 20 | 27 | 25 | 17 | 283 | 21 | 13 | 14 | ... |
| **0783239408** | 0 | 57 | 24 | 29 | 39 | 28 | 21 | 283 | 24 | 29 | ... |
| **0783241038** | 3 | 19 | 14 | 17 | 31 | 24 | 13 | 24 | 283 | 22 | ... |
| **0784010188** | 0 | 36 | 15 | 28 | 52 | 32 | 14 | 29 | 22 | 283 | ... |

9 rows × 296 columns

4.2 Specific item_IDs in three clusters of K-Means Clustering

```
print('itemIDs in Cluster 1: ', labels_1.values)
```

```
itemIDs in Cluster 1:  ['0767002652' '0783222955' '0784010218' '0788802194' '0788812408'
 '0790733226' '079074404X' '0792148061' '1417054069' '141983049X'
 '1424810248' '1424819253' '1569494126' '1582704414' '6300215695'
 '6300274187' '6301442962' '6301996135' '6302320488' '6302595916'
 '6302968143' '6303118240' '6303965415' '6304176287' '6305368139'
 'B00003CXOO' 'B00005JKVZ' 'B00005JLET' 'B00005JLF3' 'B00005JNOG'
 'B00005JPAR' 'B00006RCNW' 'B000083C6V' 'B00008YGRU' 'B0000INU6S'
 'B0000VCZK2' 'B0000WN1WW' 'B0000YTP02' 'B0001WTWDI' 'B0002ERXC2'
 'B0002I831S' 'B0002PYS7Y' 'B0006FO5LO' 'B0009AK57Y' 'B000A6T1X6'
 'B000AE4QD8' 'B000E5LEXS' 'B000FL7CAK' 'B000K7VHOG' 'B000KX0HIC'
 'B000MQC9H4' 'B000N4SHOE' 'B000NOIX48' 'B000NQRE9Q' 'B000QDLSR0'
 'B000QUEQ4U' 'B000S6LS66' 'B000WGWQG8' 'B001GKJ2DO' 'B001HN68ZU'
 'B001NFNFMQ' 'B001NFNFN0' 'B001TH16DI' 'B002BWP2IK' 'B002JVWRD6'
 'B002VWOMZ4' 'B002ZCY7SW' 'B0031RAOVY' 'B0031XYLWG' 'B0032JTV38'
 'B00395ATT0' 'B003FBNJ4U' 'B003L77G10' 'B003L77G2Y' 'B003R4ZMQI'
 'B003Y5HWJU' 'B0041G683W' 'B00466HN86' 'B004C45AX2' 'B004EPYZRG'
 'B004G5Z0B4' 'B004HW7JNS' 'B004IK30PA' 'B0053O89YM' 'B0053O8A46'
 'B0058YPG1G' 'B0058YPGSY' 'B005A1GS00' 'B005LAIIB2' 'B005LAJ16I'
 'B005LAJ22Q' 'B005LAJ23A' 'B005N4DMMG' 'B005S9ELCG' 'B006U1J5ZY'
 'B0079ILHRG' 'B008JFUS8M' 'B008KEQM3W' 'B008SOS7MS' 'B00915G6R6'
 'B00979JV00' 'B009JBZH54' 'B009LDCWWG' 'B009LDCXNY' 'B009NNM77E'
 'B00AIBZJLG' 'B00AZMFO6I' 'B00AZMFONG' 'B00B2YH7BS' 'B00BC5FN2C'
 'B00BEEKN26' 'B00BEIYGK2' 'B00BEIYMIS' 'B00BEIYMTW' 'B00BUUAV08'
 'B00C888LOO' 'B00C8CQRQ4' 'B00CS5RBPM' 'B00DZP1BZ0' 'B00E3UN44W'
 'B00FPPQYXM' 'B00FQY4C86' 'B00G2P79BU' 'B00H7KJTCG']
```

```
print('itemIDs in Cluster 2: ', labels_2.values)
```

```
itemIDs in Cluster 2:  ['6302219205' 'B001OQCV56' 'B001UV4XFG' 'B001UV4XHY' 'B001UV4XIS'
 'B00275EHJG' 'B002BWP49C' 'B002VECM6S' 'B002ZG97YM' 'B002ZG98J6'
 'B002ZG999U' 'B0034G4OYA' 'B0034G4P3O' 'B0034G4P4O' 'B003UESJH4'
 'B003Y5H53S' 'B003Y5H5H4' 'B00466HN7M' 'B004A8ZWVK' 'B004BDOEZO'
 'B004EPYZP8' 'B004EPYZPS' 'B004EPYZUS' 'B004EPZ07K' 'B004HO6I4M'
 'B004LWZW42' 'B004LWZWFQ' 'B00542PSY2' 'B005IZLPKQ' 'B005LAIH4A'
 'B005LAIHKY' 'B005LAIHPE' 'B005LAIHQS' 'B005LAIHR2' 'B005LAIHUO'
 'B005LAIHW2' 'B005LAIHYU' 'B005LAII3A' 'B005LAII7G' 'B005LAII80'
 'B005LAII8K' 'B005LAIIFS' 'B005LAIIQC' 'B005LAIISA' 'B0067EKYAY'
 'B0067EKYS6' 'B007K3JCAE' 'B007L6VR12' 'B008220AGC' 'B008220BGQ'
 'B008220BLG' 'B008220CQU' 'B008G33OOG' 'B008JFUN50' 'B008JFUOWM'
 'B008JFUP8A' 'B008JFUPOY' 'B008JFURII' 'B008WCP2KG' 'B0090SI56Y'
 'B009369Z8A' 'B0095HHM78' 'B0099114WO' 'B009AMAK54' 'B009AMAOTQ'
 'B009NNM9OA' 'B00AFEY354' 'B00AZNEW5G' 'B00B74MJOS' 'B00BEIYHO2'
 'B00BEIYLO8' 'B00BEIYSL4' 'B00BEJL69U' 'B00BUADSMQ' 'B00CHVIAS4'
 'B00CIXVAN8' 'B00CWM58WY' 'B00DL47RQ2' 'B00DL48BM6' 'B00DV1XYTO'
 'B00EV4EUT8' 'B00G15MDI0' 'B00GMV8LIO' 'B00GSBMNOQ' 'B00GST8U4U'
 'B00H9L26AA' 'B00HEPE6MM']
```

```
print('itemIDs in Cluster 3: ', labels_3.values)
```

```
itemIDs in Cluster 3:  ['0780623134' '0782002064' '0783211856' '0783216084' '0783222734'
 '0783239408' '0783241038' '0784010188' '0788828126' '078886047X'
 '0788860704' '0788882988' '0790705141' '0790729725' '0790743132'
 '0792140923' '0792151712' '1415724784' '1417030321' '1419838830'
 '1558908242' '6302526574' '6302610702' '6302760046' '6303122647'
 '6304117752' '6304994540' '6305123616' '6305568901' '630573240X'
 '7883704591' '7883706837' 'B00000K3AM' 'B00003CWQ2' 'B00003CXE6'
 'B00003CXI0' 'B00003CXP1' 'B00003CXR3' 'B00003CXRM' 'B00003CXTF'
 'B00003CXXI' 'B00003CXXO' 'B00005JKCH' 'B00005JKDQ' 'B00005JKZY'
 'B00005JL3T' 'B00005JLWN' 'B00005JLXH' 'B00005JM02' 'B00005JM5E'
 'B00005JMAH' 'B00005JMEW' 'B00005JMFQ' 'B00005JMJ4' 'B00005JMYI'
 'B00005JNBQ' 'B00005JNEI' 'B00005JNS0' 'B00005JNTI' 'B00005JO20'
 'B00005JPA6' 'B00005JPI2' 'B00005JPLW' 'B00005JPNO' 'B00005JPS8'
 'B00005JPTK' 'B00005QZ7U' 'B00005V3Z4' 'B0000633ZP' 'B0000640VO'
 'B00006AL1D' 'B000A3XY5A' 'B000E1MTYK' 'B000M341QE' 'B000MNP2K8'
 'B000VBJEEG' 'B000ZECQ08' 'B001FB55H6' 'B001GCUNYO' 'B001KVZ6FW'
 'B001OQCV6A' 'B0021L8V1G' 'B002RD55JE' 'B002ZG980U']
```

Reference


[1] Recommender systems with social regularization.

H. Ma, D. Zhou, C. Liu, M. Lyu, and I. King. In WSDM, 2011.]

[2] Network Science

https://en.wikipedia.org/wiki/Network_science

[3] Degree (graph theory)

https://en.wikipedia.org/wiki/Degree_(graph_theory)

[4] Transitivity

http://mathinsight.org/definition/transitivity_graph

[5] K-Means Clusterling

https://en.wikipedia.org/wiki/K-means_clustering

[6] T-SNE algorithm

https://lvdmaaten.github.io/tsne/