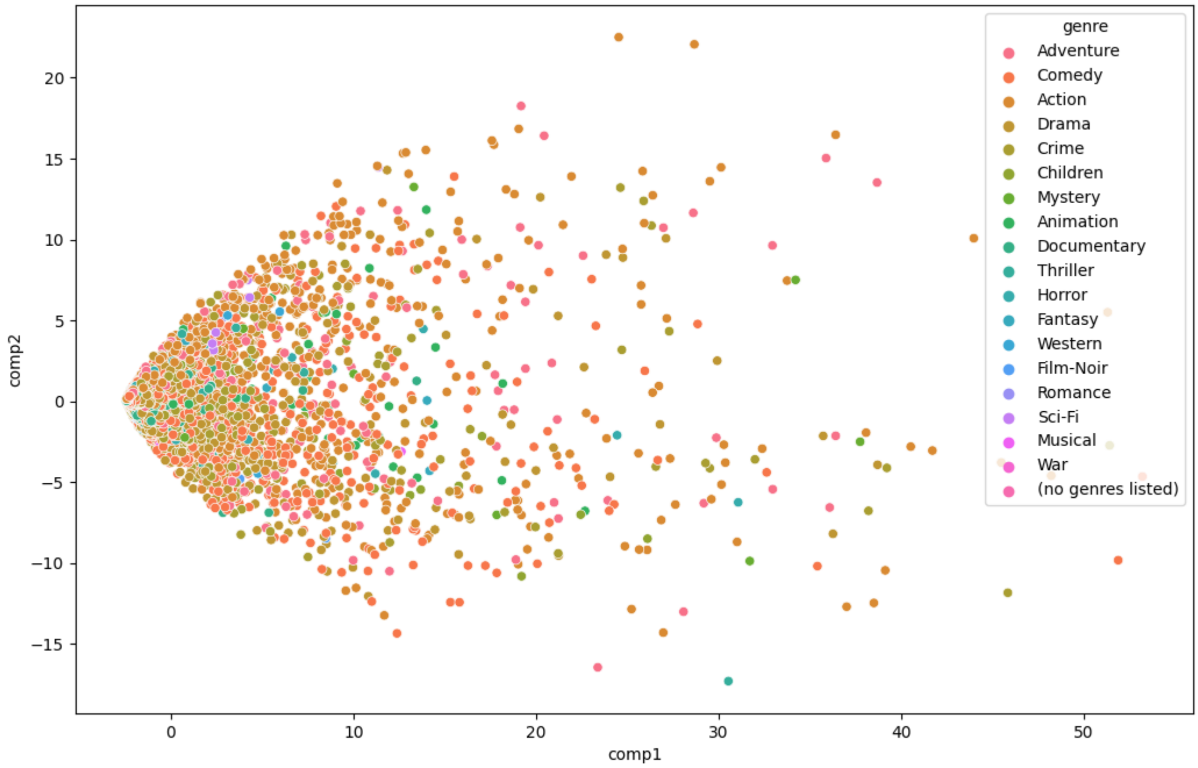
SKKU 3049-41: Project 2 2016312761 여혁수

2 - (c) Plot the results and color each movies by its genre. Discuss what patterns you see in the visualization.

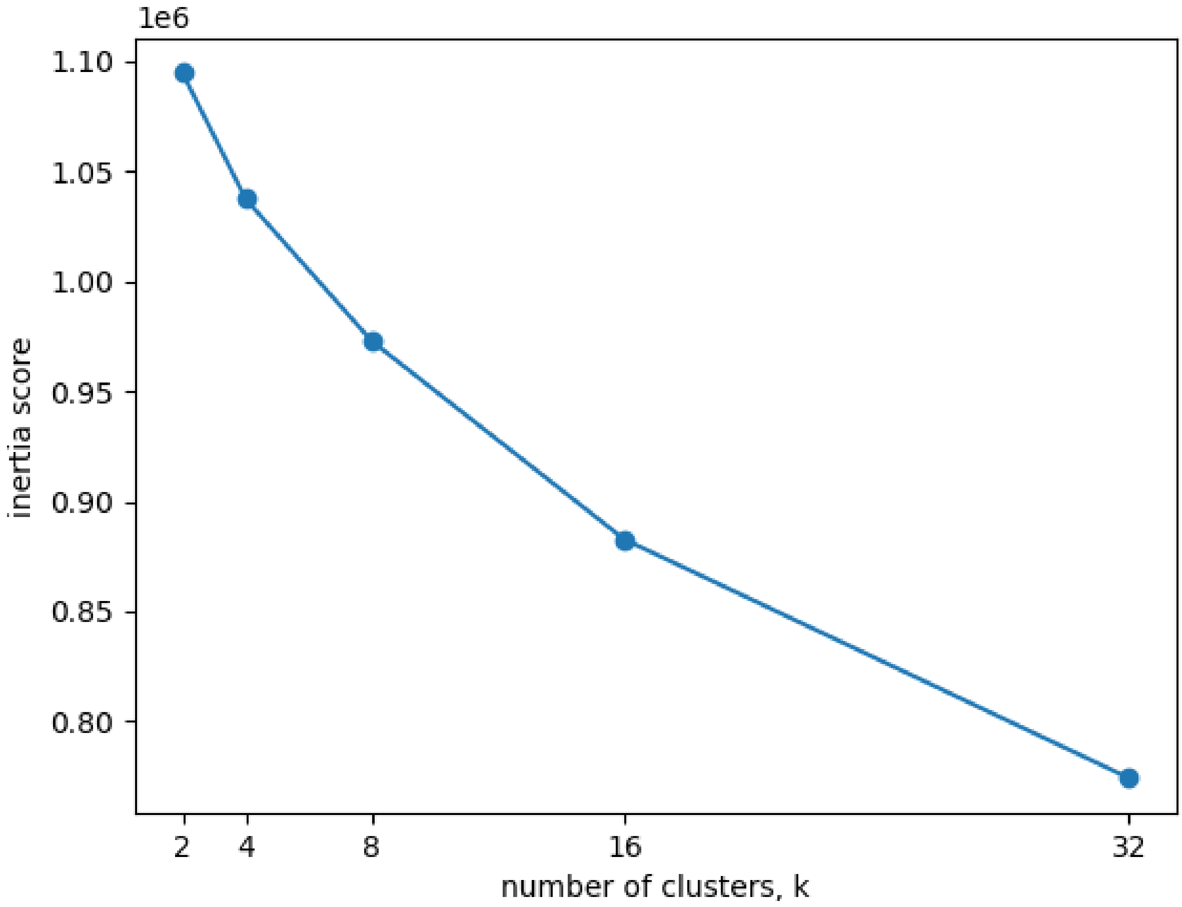


First, I can know that there are many movies of Adventure and Drama by finding the most of these colors. And it has a shape like sector form from the point near (0, 0) except some outliers. But I can't find any pattern in each group of genre. All genres are mixed up and make noisy data.

2 - (d) Determine the "intrinsic" dimensionality of the movies, by finding the number of principle components that are needed to explain 80% of the variance of the data. Discuss how this compares to k = 2 and how this may impact the quality of the visualization above.

I found the number of principle components k = 153. It will be the intrinsic dimensionality of the movies because from this number of components, it can explain more than 80% of the variance of the data. Comparing with k = 2, It can explain variance of the data almost 4 times much than the case of k = 2. If principle components k = 2, it can explain about 21.8% of the variance of the data. If we can explain more than 80% of the variance of the data, I think it can be said that we found a combination of variables that can greatly describe the data while minimizing loss of original data. So if we apply k = 153 instead of k = 2, it may incredibly increase the quality of plot graph by distinguishing and grouping movies data more clearly by movies' genre. However, It may cause some overfitting problem.

3 - (a) For values of k = [2, 4, 8, 16, 32], apply k-means and measure the inertia for each value of k. Plot the resulting inertia scores for each choice of k.



3 - (b) From the above results, choose an appropriate value of k from the plot and support your choice.

I chose value of k to 8 because I think this point is the most proper point to be the "elbow" of the plot. Before k = 8, inertia score goes better quite a lot. After k = 8, slope of the graph becomes gentle, and the width of the change of inertia score is not very large considering the increase of the number of clusters, k. I think it doesn't matter to choose k to 16 because it has quite lower inertia score than k = 8. However, if we do clustering by 16 clusters, we may get some clusters which are composed of only one or two users. It feels like overfitting and it is uncomfortable while finding the top three movies that are highest rated in the cluster.

3 - (c) Cluster the data again with your chosen value of k. For each of the resulting clusters, find the top three movies that are highest rated (on average) by the users in the cluster. Report the movie titles and discuss whether the results seem reasonable (i.e., do the top-rated movies in each cluster seem to correspond to recognizable groups).

in cluster 0)

1. Four Rooms (1995)

Persuasion (1995)

Lamerica (1994)

3 movies in this cluster do not have common genre. But Movie Persuasion and Lamerica are both Drama genre. Also all 3 movies are released in 1994~1995. So they can be grouped by the year of the movie's release.

in cluster 1)

1. Eat Drink Man Woman (Yin shi nan nu) (1994) 2. Spellbound (1945) 3. Affair to Remember, An (1957)

3 movies in this cluster are Romance genre. So they can be grouped with 'Romance'.

in cluster 2)

1. Pulp Fiction (1994) 2. Dr. Strangelove or: How I Learned to Stop Worr... 3. Vertigo (1958)

This cluster is quite unreasonable. Although first movie and second movie are both Comedy genre, and also first and third movie are Thriller movie, there is not common genre. Release years also do not overlap.

in cluster 3)

1. Three Colors: Red (Trois couleurs: Rouge) (1994) 2. Shallow Grave (1994) 3. Night of the Living Dead (1968)

This cluster is also quite unreasonable. Second and third movie are Thriller movie, so very relative. However, first movie has no relation with others.

in cluster 4)

1. Antonia's Line (Antonia) (1995) 2. Hate (Haine, La) (1995) 3. Heidi Fleiss: Hollywood Madam (1995)

Genre of movies in cluster is different, but these 3 movies are released in 1995. They can be grouped by the year of the movie's release.

in cluster 5)

1. Lamerica (1994) 2. Anne Frank Remembered (1995) 3. Chungking Express (Chung Hing sam lam) (1994)

Genre of movies in cluster is different, but these 3 movies are released in 1994~1995. They can be grouped by the movie's release year.

in cluster 6)

1. Sense and Sensibility (1995) 2. Dead Man Walking (1995) 3. Shawshank Redemption, The (1994)

3 movies in this cluster are Drama genre. So they can be grouped with 'Drama'. Also they can be grouped by the movie's release year.

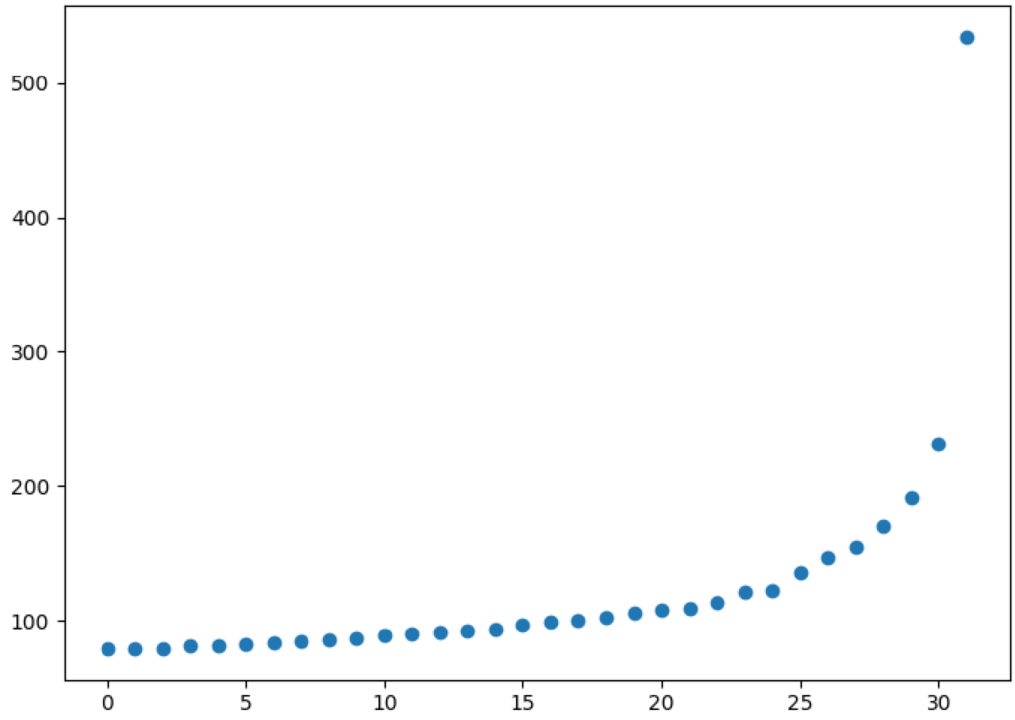
in cluster 7)

1. Beautiful Girls (1996) 2. Smoke (1995) 3. Once Were Warriors (1994)

3 movies in this cluster are Drama genre. So they can be grouped with 'Drama lovers'. Also they can be grouped by the movie's release year.

Overall result of clustering seems unreasonable. Because the standards of grouping of each cluster are overlapped each other. For example, there are several clusters which can be grouped by such as 'released in 1994~1995' or 'Drama lovers'.

4 – (a)



4 - (b) For each of the values of k = [2, 4, 8, 16, 32] considered above, report the sum of the explained variance ratio. Discuss how the results compare to the inertia values above and whether it supports your choice of k.

sum of the explained variance ratio: 0.17282699355904324 (k = 2)

0.22186799704913987 (k = 4)

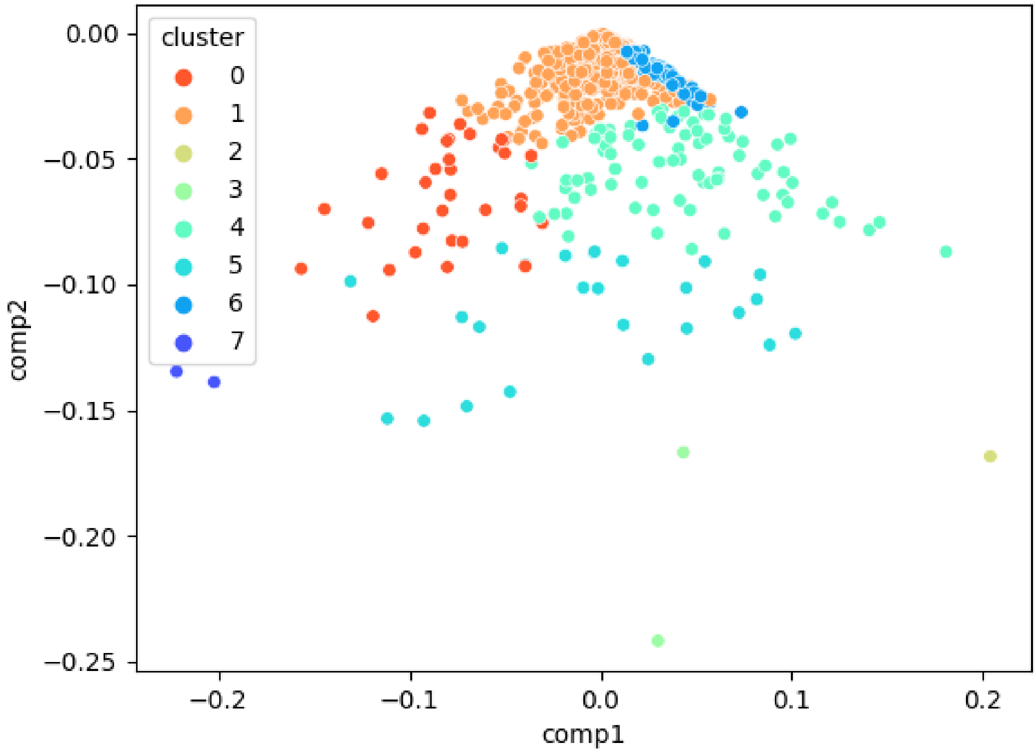
0.28702886824615936 (k = 8)

0.36343102347287404 (k = 16)

0.4606019741780782 (k = 32)

Explained variance ratio is inversely proportional relation with inertia values. Because both values indicate quite opposite meaning. Lower inertia value it is, higher the degree of cohesion between components. However, higher explained variance ratio it is, the more components we can explain, thus cohesion between components also go high. I think my choice of k = 8 can't be supported by sum of the explained variance ratio. Because the value about 0.29 is quite low to be a proper explained variance ratio.

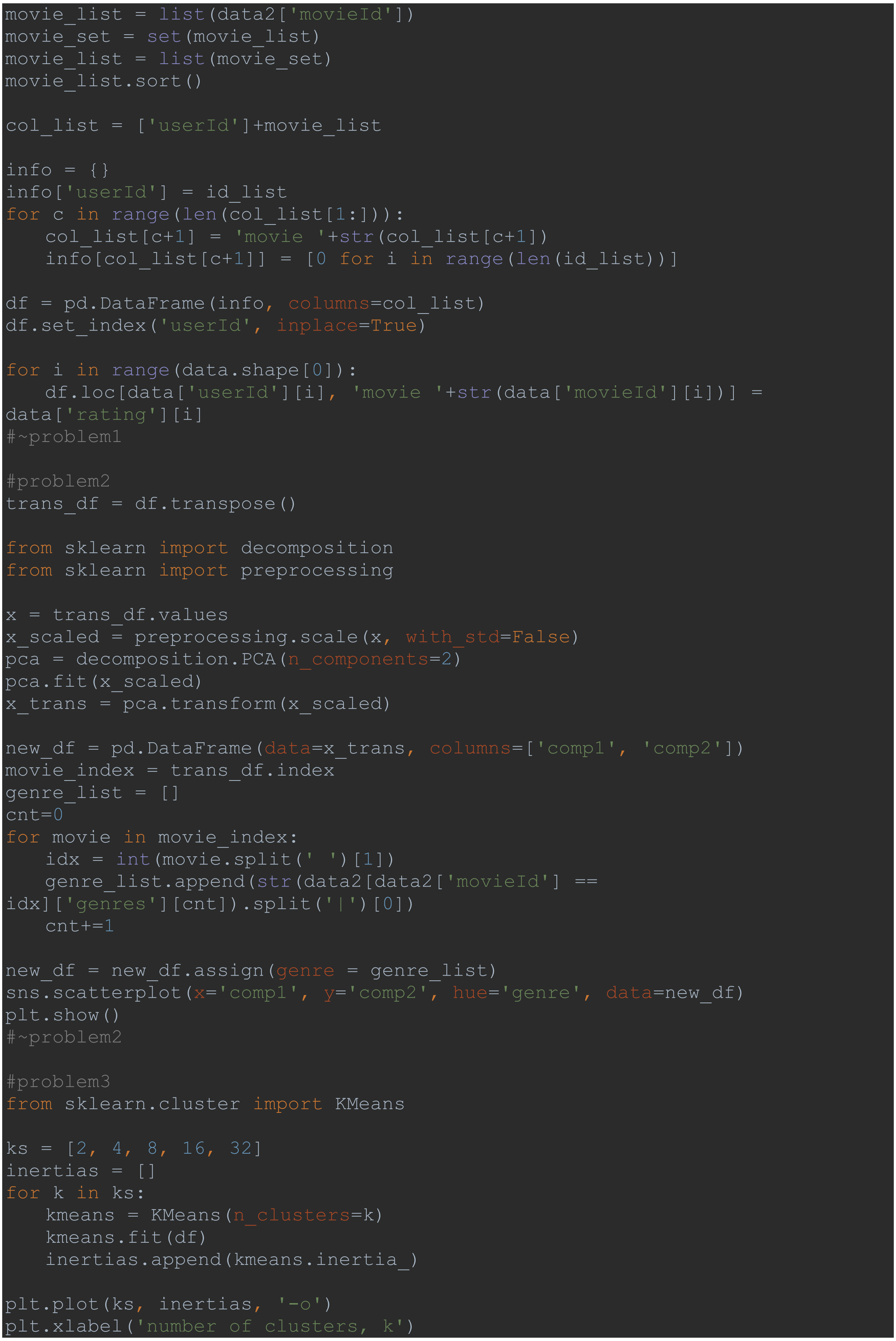
4 - (d) Plot the results (for k = 2) and color the users by the cluster memberships you found above. Discuss any patterns you can see and compare them to the previous analysis (either from clustering or PCA).



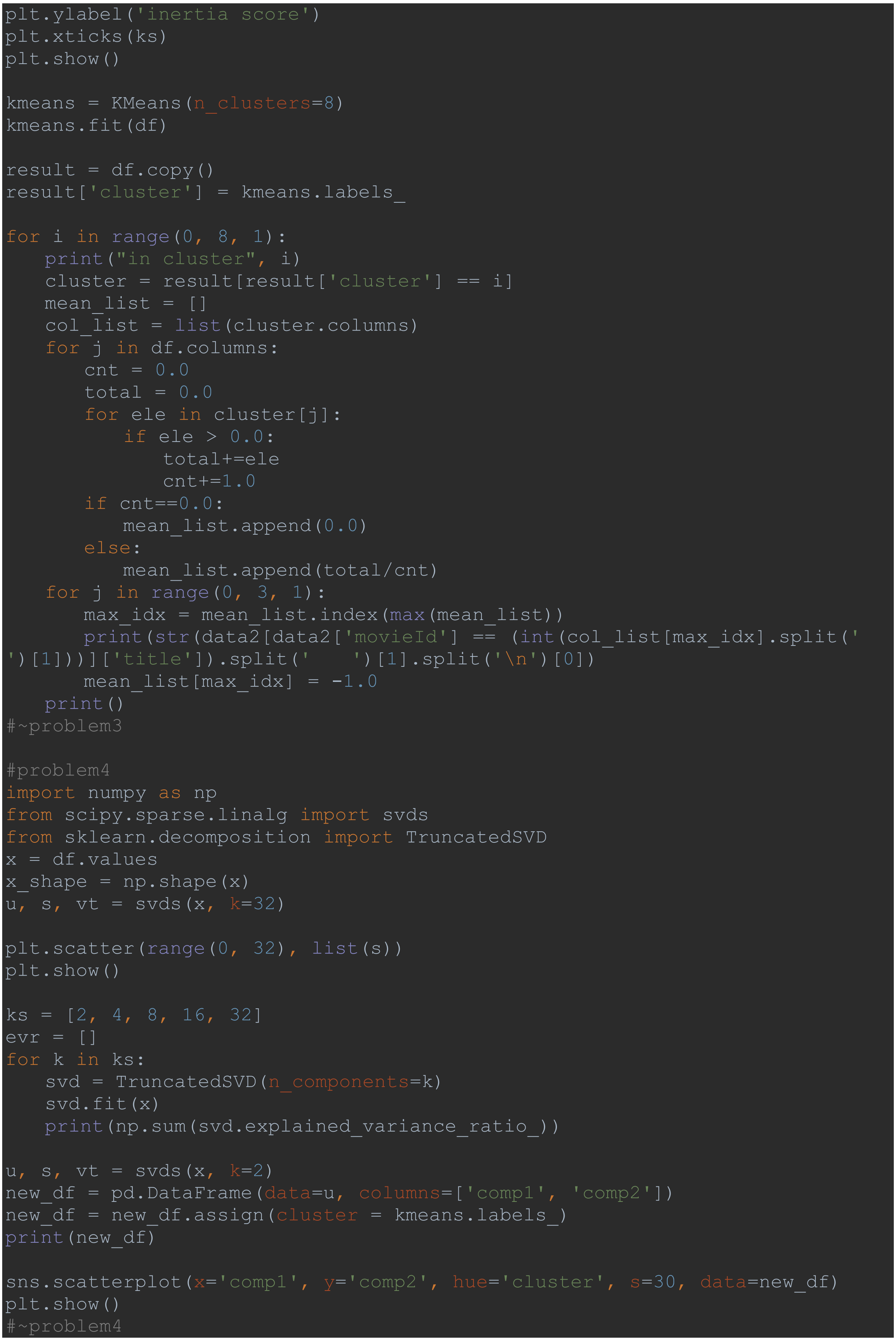
I can see patterns that components having same color are gathered together. But also I can see some outliers that distance between components is very far. Moreover, like plot of the result PCA with number of components k = 2, it has a shape like sector form. But compared with PCA plot, a kind of classification is clearly visible.

Code)

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

#problem1
data = pd.read_csv('ratings.csv')

id_list = list(data['userId'])
id_set = set(id_list)
id_list = list(id_set)
id_list.sort()

data2 = pd.read_csv('movies.csv')
```



```
movie_list = list(data2['movieId'])
movie_set = set(movie_list)
movie_list = list(movie_set)
movie_list.sort()

col_list = ['userId']+movie_list

info = {}
info['userId'] = id_list
for c in range(len(col_list[1:])):
    col_list[c+1] = 'movie '+str(col_list[c+1])
    info[col_list[c+1]] = [0 for i in range(len(id_list))]

df = pd.DataFrame(info, columns=col_list)
df.set_index('userId', inplace=True)

for i in range(data.shape[0]):
    df.loc[data['userId'][i], 'movie '+str(data['movieId'][i])] =
data['rating'][i]
#~problem1

#problem2
trans_df = df.transpose()

from sklearn import decomposition
from sklearn import preprocessing

x = trans_df.values
x_scaled = preprocessing.scale(x, with_std=False)
pca = decomposition.PCA(n_components=2)
pca.fit(x_scaled)
x_trans = pca.transform(x_scaled)

new_df = pd.DataFrame(data=x_trans, columns=['comp1', 'comp2'])
movie_index = trans_df.index
genre_list = []
cnt=0
for movie in movie_index:
    idx = int(movie.split(' ')[1])
    genre_list.append(str(data2[data2['movieId'] ==
idx]['genres'][cnt]).split('|')[0])
    cnt+=1

new_df = new_df.assign(genre = genre_list)
sns.scatterplot(x='comp1', y='comp2', hue='genre', data=new_df)
plt.show()
#~problem2

#problem3
from sklearn.cluster import KMeans

ks = [2, 4, 8, 16, 32]
inertias = []
for k in ks:
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(df)
    inertias.append(kmeans.inertia_)

plt.plot(ks, inertias, '-o')
plt.xlabel('number of clusters, k')
```



```
plt.ylabel('inertia score')
plt.xticks(ks)
plt.show()

kmeans = KMeans(n_clusters=8)
kmeans.fit(df)

result = df.copy()
result['cluster'] = kmeans.labels_

for i in range(0, 8, 1):
    print("in cluster", i)
    cluster = result[result['cluster'] == i]
    mean_list = []
    col_list = list(cluster.columns)
    for j in df.columns:
        cnt = 0.0
        total = 0.0
        for ele in cluster[j]:
            if ele > 0.0:
                total+=ele
                cnt+=1.0
        if cnt==0.0:
            mean_list.append(0.0)
        else:
            mean_list.append(total/cnt)
    for j in range(0, 3, 1):
        max_idx = mean_list.index(max(mean_list))
        print(str(data2[data2['movieId'] == (int(col_list[max_idx].split('
')[1]))]['title']).split('    ')[1].split('\n')[0])
        mean_list[max_idx] = -1.0
    print()
#~problem3

#problem4
import numpy as np
from scipy.sparse.linalg import svds
from sklearn.decomposition import TruncatedSVD
x = df.values
x_shape = np.shape(x)
u, s, vt = svds(x, k=32)

plt.scatter(range(0, 32), list(s))
plt.show()

ks = [2, 4, 8, 16, 32]
evr = []
for k in ks:
    svd = TruncatedSVD(n_components=k)
    svd.fit(x)
    print(np.sum(svd.explained_variance_ratio_))

u, s, vt = svds(x, k=2)
new_df = pd.DataFrame(data=u, columns=['comp1', 'comp2'])
new_df = new_df.assign(cluster = kmeans.labels_)
print(new_df)

sns.scatterplot(x='comp1', y='comp2', hue='cluster', s=30, data=new_df)
plt.show()
#~problem4
```