1 | Introduction
1.1 Goal of the project

Throughout the course, various algorithms and properties are discussed. However, when it comes to 'real-world data', it might not work like expected and the data need to be somewhat modified due to the properties of data. In this project, data are collected from the web site using web crawling. Unlike the refined data, data crawled from the web must be preprocessed and dimension needs to be reduced before the modeling. This project focuses especially on Korea movie data, as there are already many research based on US movie data such as 'IMDB'.

In this project, I examine:

- how different clustering algorithms (K-means, Hierarchical and DBSCAN) perform differently depending on hyperparameters and the properties of data.
- how the properties of data are reflected in the performance of different algorithms.
- the way how data are acquired from the web site by crawling and preprocessed so as to make real-world movie data to adapt a model, especially clustering algorithms

2 | Data & Methodology
2.1 Data Crawling

- Site: **Daum movie annual box office rankings** (https://movie.daum.net/boxoffice/yearly)
- Target data: **850 entries with 9 columns from 2004 to 2020** shown as follows (Movie title/Box office year/Released year/Released Month/Netizen Grade/Ratings/Total Audience/Nation/Genre)
- Python Package: BeautifulSoup
- In the web site, annual Korean box office rankings are at each page. At ranking page, unique id for each movie given by web site, title, released year, month and netizen grade are collected. With 'movie_id', data of nation, genre are collected by detail page url of each movie. Total audience data of each movie are collected by json.

	year	title	release_year	release_month	nation	genre	netizen_grade	rating	total_audience
36635	2004	태극기 휘날리며	2004	02	한국	전쟁	9.1	15세관람가	11,746,135
37066	2004	트로이	2004	05	미국, 몰타, 영국	액션/로맨스/멜로	8.4	15세관람가	2,001,318
39368	2004	내 머리 속의 지우개	2004	11	한국	로맨스/멜로/드라마	8.7	12세관람가	1,887,868
39411	2004	귀신이 산다	2004	09	한국	코미디/판타지/공포	6.4	12세관람가	1,875,936
38807	2004	투모로우	2004	06	미국	액션/스릴러/SF/어드벤처/드라마	8.2	12세관람가	1,830,767

2 | Data & Methodology
2.2 Data Preprocessing, Transformation and PCA

Important to guarantee the quality of datasets through preprocessing, transformation and reducing dimensionality to fit clustering algorithms.

- **Preprocessing**

  1) Deleting null, duplicated values, redundant columns and correcting the format of data

  2) **One-hot encoding** for categorical features: The standard k-means algorithms isn't directly applicable to categorical data as the Euclidian distance function on categorical sample space is not so meaningful. Thus, categorical data must be one-hot encoded.

  3) Combining some features into one column

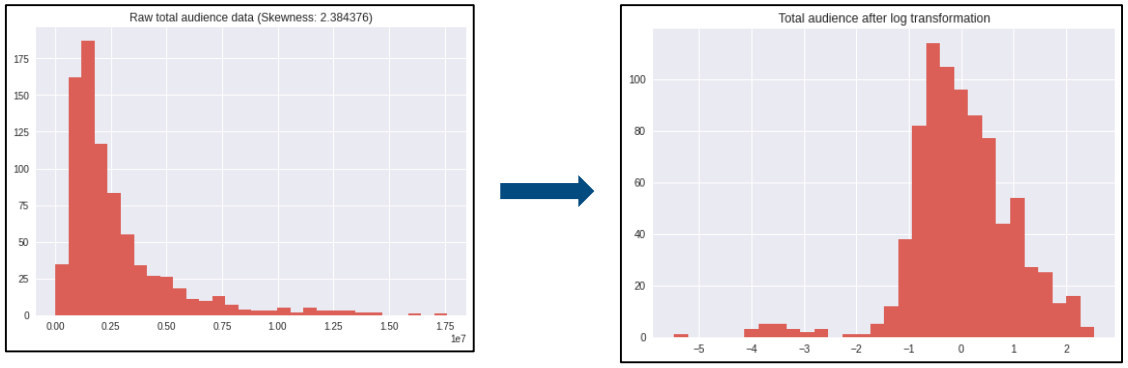
Feature	Preprocessing Details
Box office year	Deleted as it's redundant to 'released year'
Release year	One-hot encoded for each year
Release month	One-hot encoded and each month is combined to season(quarter 1~4)
Ratings	Different terms are used for same ratings like '15세관람가' and '15세 이상관람가'. These are replaced by same term and one-hot encoded.
Nation	One-hot encoded and each country is combined to larger groups such as EUR(UK, Germany etc), Asia(Japan, China etc) and so on. As more than half of the entries are made in Korea, 'Korea' is grouped by its own.
Genre	One-hot encoded and each genre is combined to larger groups based on similarity such as 'romance and melodrama', 'horror and thriller' and so on.

- **Transformation**

1) Total audience: Extremely positive skewed (Skewness: 2.384) → **Log transformation**
2) Difference of scale between total audience and netizen grade → **Scaling:** Since clustering such as K-means and DBSCAN is a distance-based algorithm, numerical data must be standardized to eliminate the effects of the scale.

- **Principal Component Analysis:** 39 → 2 features

For displaying clustering results in two dimensions and for preventing the curse of dimensionality, the data are orthogonally projected to the principal subspace such that the variance of the projected data is maximized.



↓ The overall skewness of total audience before and after log transformation

```
standard_scaler = StandardScaler()
scale_df = df.copy()

# Scaling log transformed audience and netizen grade
col_names = ['total_audience_log', 'netizen_grade']
features = scale_df[col_names]
scaler = standard_scaler.fit(features.values)
features = scaler.transform(features.values)
scale_df[col_names] = features
```

→ Parts of code that scaling the features 'total audience' and 'netizen grade'

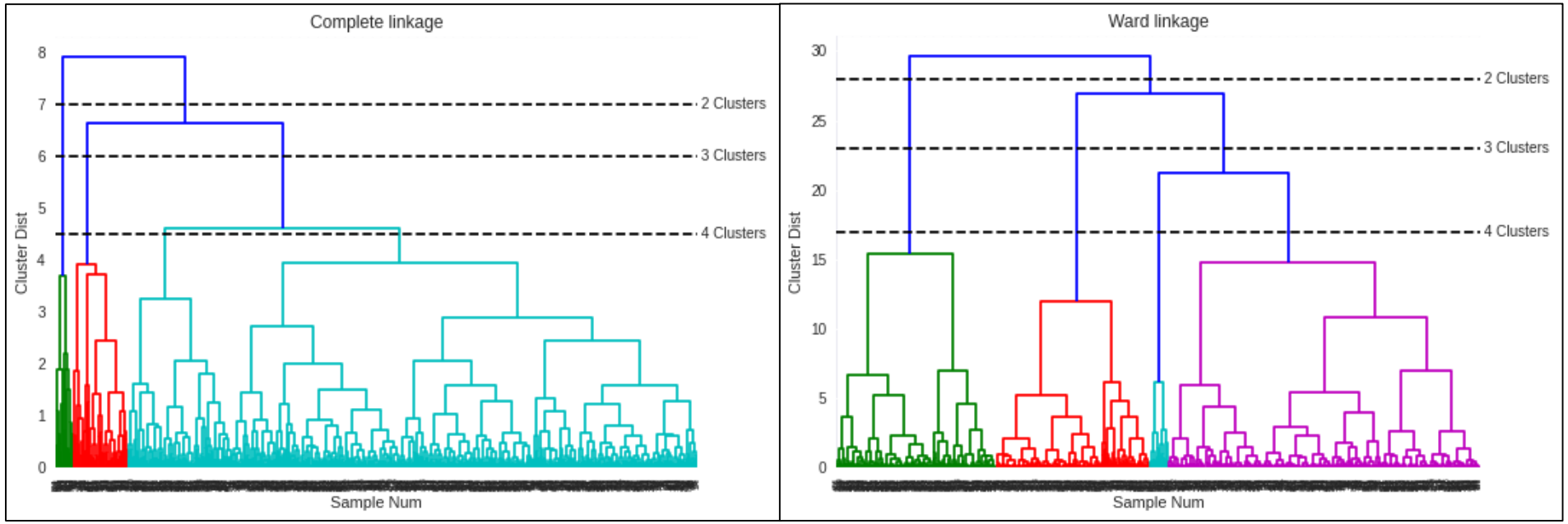
	weights	features	abs_weights
40	-0.711234	total_audience_log	0.711234
0	-0.677107	netizen_grade	0.677107
8	-0.078238	Drama	0.078238
39	0.075098	2020	0.075098
18	0.057158	Q1	0.057158
14	-0.054426	rating_12	0.054426
15	0.053945	rating_15	0.053945
13	0.053118	Thriller	0.053118

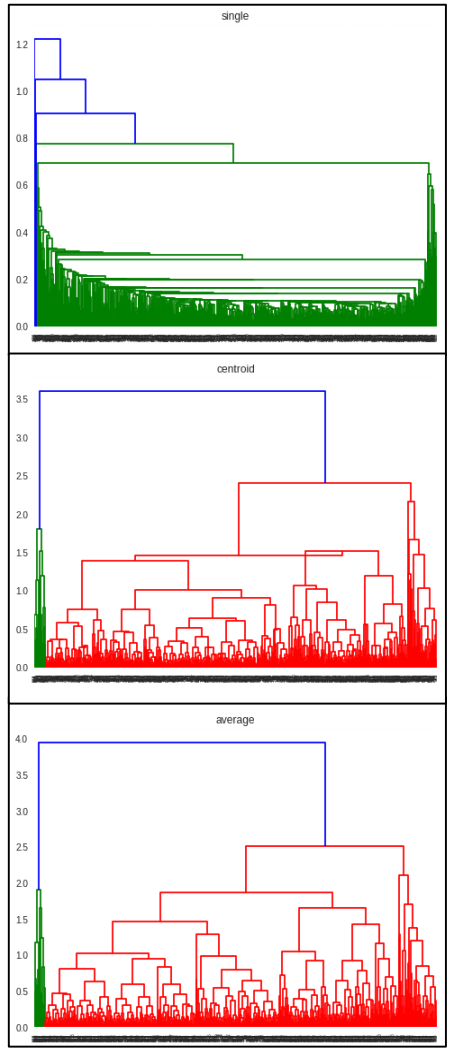
```
pca = PCA(n_components=2)
reduced_scale_df = pca.fit_transform(scale_df)
reduced_scale_df.shape

(822, 2)
```

→ PCA code and the most important variables to the first component (Total audience, netizen grade, genre "Drama" etc)

3 | Clustering
3.1 Hierarchical Agglomerative Clustering

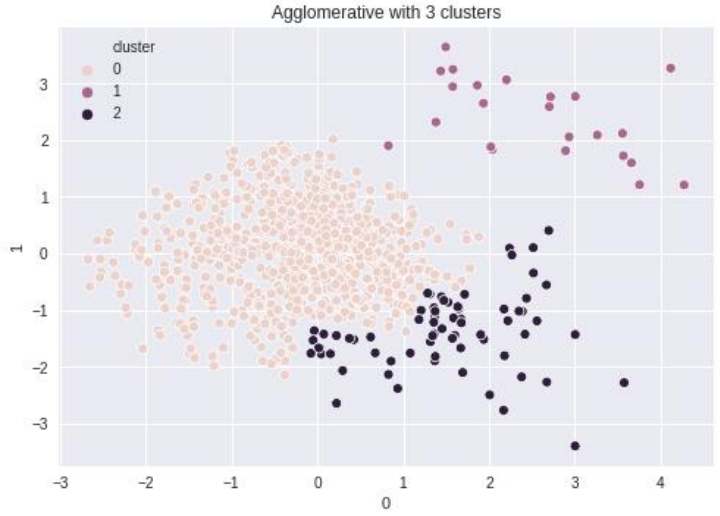


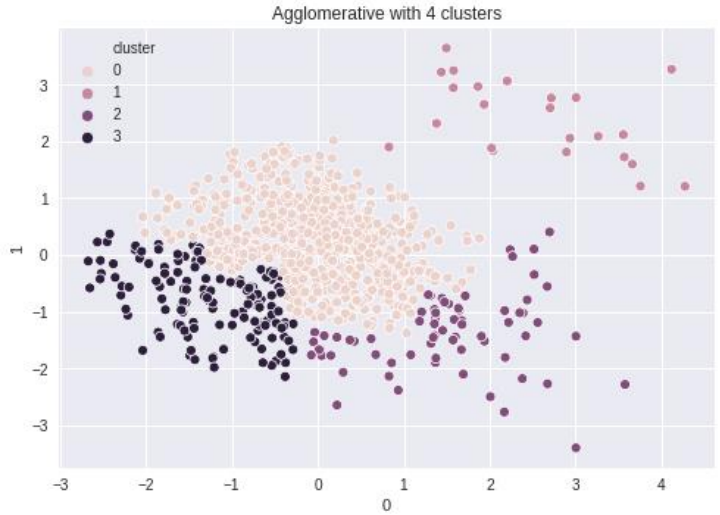


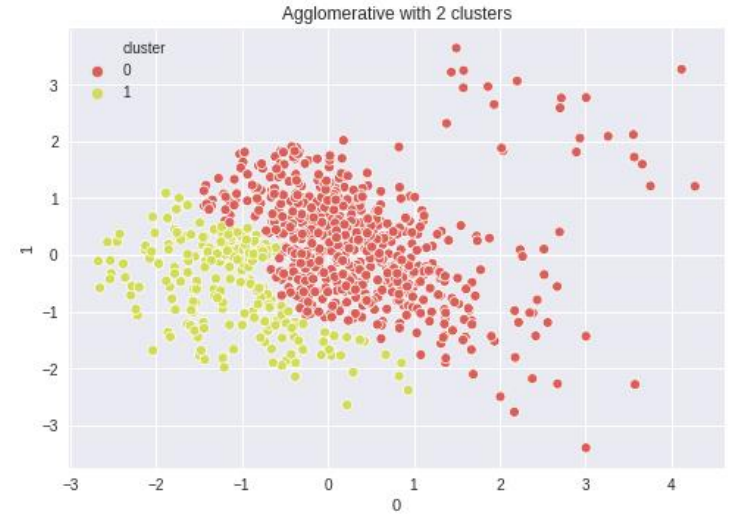
- Comparisons of **measures of the computation of the proximity of two clusters** (e.g. single, complete, average, centroid and ward linkage)

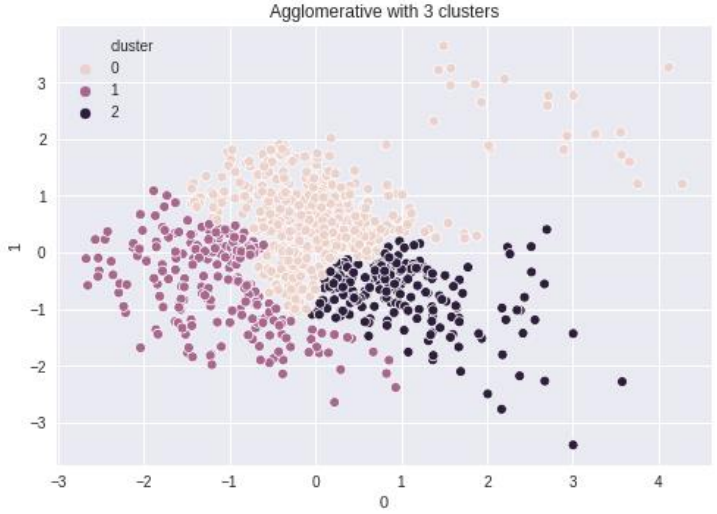
→ The data are clustered well when using '**ward linkage**' and the distance between clusters are relatively larger than other measures. The '**complete linkage**' also works well.

(Right Dendrograms: Single, centroid, average linkage from upper one)







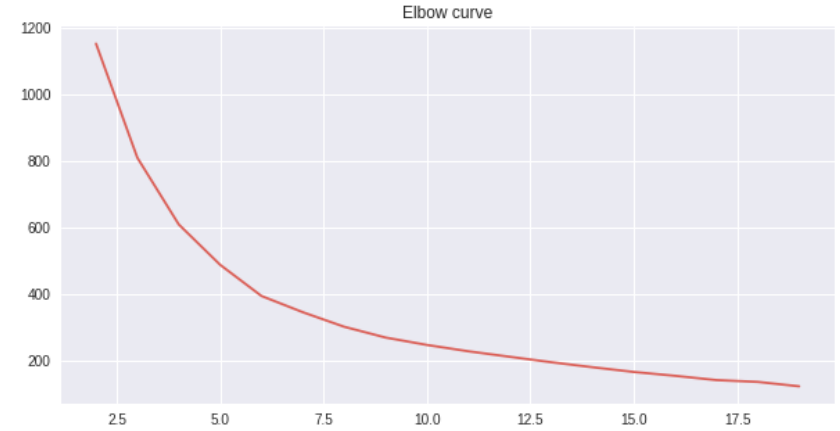


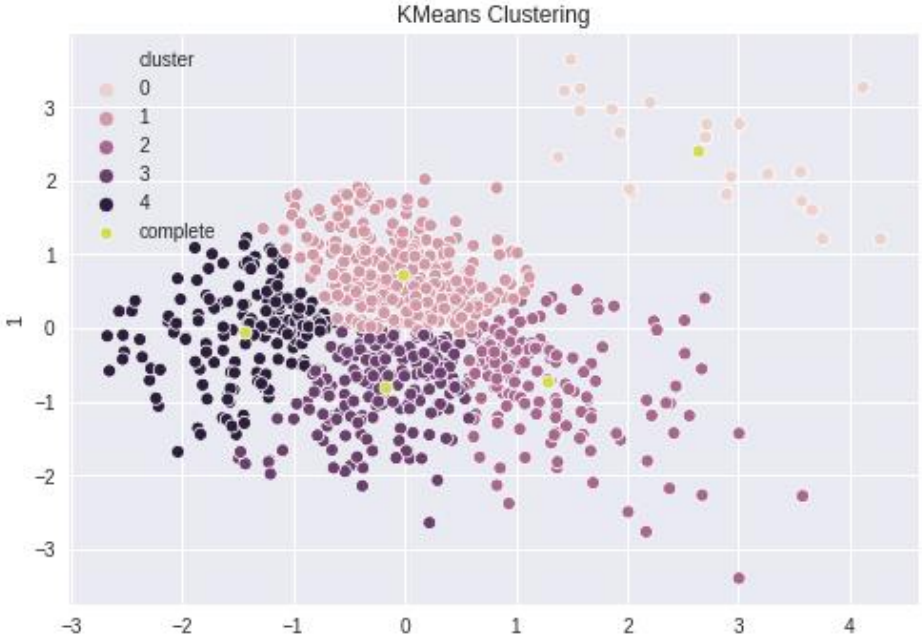
- First row: Complete linkage with 2, 3, 4 clusters / Second row: Ward linkage with 2, 3, 4 clusters

→ **Complete Linkage with 2 clusters** seems to be well clustered the most. Others are not appropriate as close points are divided into different clusters

(Against the goal of clustering: *Minimizing the intra-cluster variance and maximize the inter-cluster variance*)

3 | Clustering
3.2 K-means Clustering





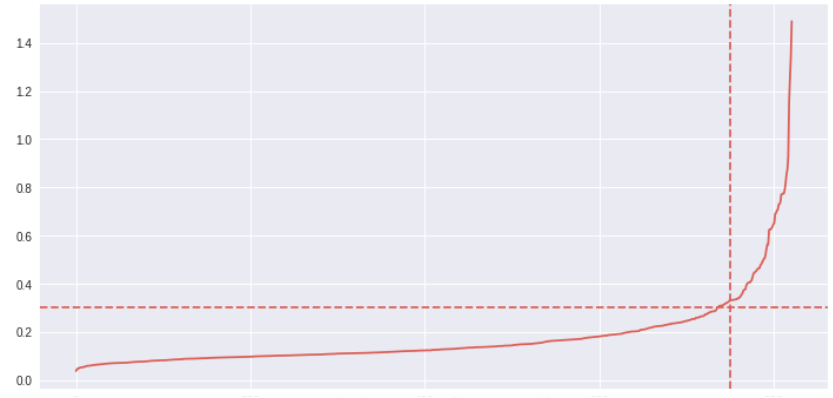
- **Hyperparameter K = 5** : decided by elbow point (Distortions decrease in a linear fashion since k=5. The optimal number of clusters for the data is 5)

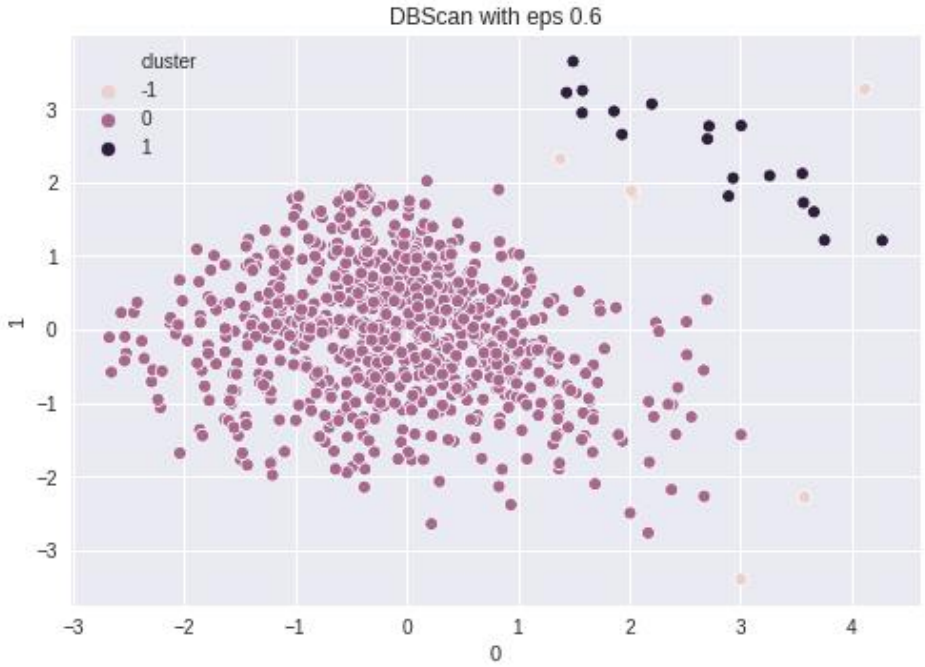
- As shown in the plot of the result of K-means clustering with 5 clusters, clusters are made in undesirable way as points gathering close together are divided into different clusters.

- It attributes to the fact that the properties of the data don't match to K-means algorithms. K-means algorithm has limitations that it can't cope with different size, densities, or non-globular shaped clusters. As K-means algorithm simply allocates clusters to near-distance points, largely dense clusters may be divided just like this case.

- The data used in this project seem to have clusters with huge dense clusters, divided into clusters of unequal density. It does not fit well with K-means algorithms.

3 | Clustering
3.3 DBSCAN and Evaluation

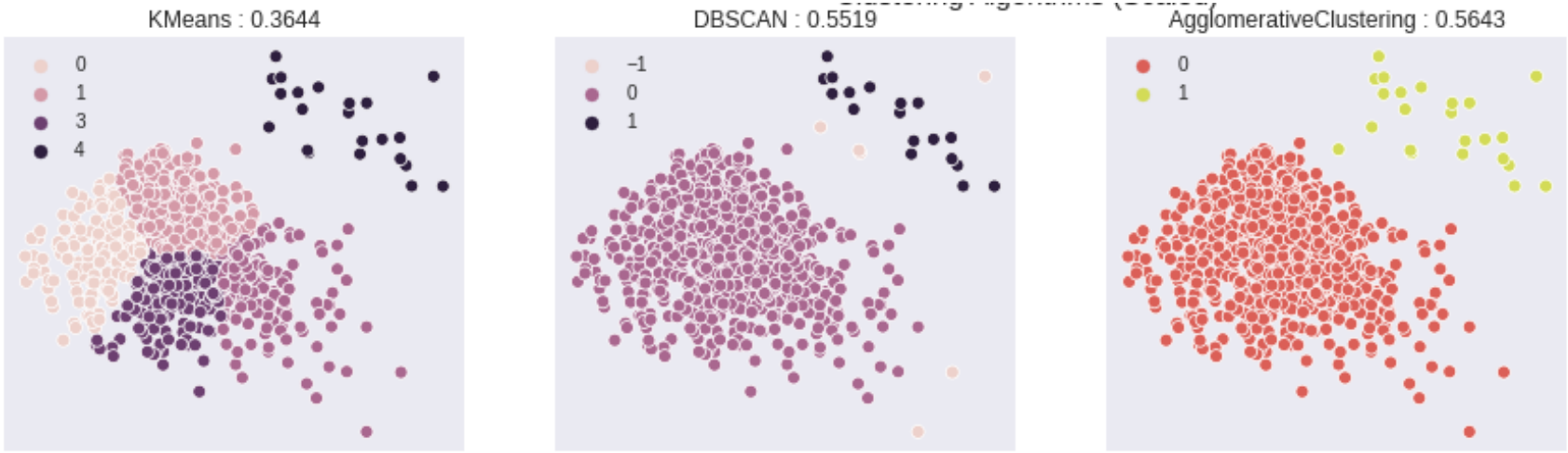




- As the K-means algorithm did not work well due to the form of the data, I experimented one of the density-based algorithm **DBSCAN(Density-Based Spatial Clustering of Applications with Noise)** that are independent of the form of the data.

- After choosing appropriate epsilon as 0.6, the clustering results are as follows. It works well.

4 | Conclusion & Reference



- As the data have properties that clustered with huge dense clusters and divided into clusters of unequal density, it does not fit to K-means algorithm, while DBSCAN and hierarchical agglomerative clustering algorithms work well. (Numbers in parentheses are Silhouette coefficients for each case) It is successfully shown that the properties of data are reflected in the performance of different algorithms and hyperparameters have dominant effects of results of clustering.
- It is expected that it will be possible to eliminate the outlier or add other features to the data in next research.

1. Aditya TS, Karthik Rajaraman, and M. Monica Subashini, '*Comparative Analysis of Clustering Techniques for Movie Recommendation*', MATEC Web of Conferences 225, 02004 (2018).
2. Jai-Ill Lee. Young-Ho Chun. Chunghun Ha, '*Clustering Analysis of Films on Box Office Performance : Based on Web Crawling*', School of Information & Computer Engineering, Hongik University(2016).