ANLY501 Project2 Report

Xinyue Liu, Zikai Zhu, Tianyi Yang, Ju Huang

Key Takeaways

Introduction:

We studied the movie ratings and box office (in this report, "box office" simply means box office receipt) for the past 7 years (2010 to 2017). Our data includes different attributes of a movie such as genre, movie length, how many theaters play the movie, etc. Some key takeaways are listed below.

Key Takeaways:

- Do movie ratings on different platforms agree with each other?

Running a t-test between Rotten Tomatoes and imdb, we find that there's a significant difference between the ratings of two platforms. More specifically, people on Rotten Tomatoes tend to rate more extremely. They score 'good' movies higher and score 'poor' movies lower.

- Is there a difference for box office between different years or studios?

An anova test shows that there's no significant difference between years; meanwhile, studio does impact the box office.

- Is there a correlation between box office and rating?

By calculating the correlation, we are surprised that there are a lot of movies with very low box office but very high ratings. In addition, if the movie has a high box office, then it will also have a good rating score.

- What are the commonalities or differences for movies in the same cluster or between different clusters?

First apply different clustering algorithms on dataset to generate clustering, and then study the distribution of different attributes within each of the cluster.

For the three clustering we generated, distributions of combined rating for different clusters tend to spread on different ranges, which implies that combined rating could be one of the strong factors of formation of clustering formation. (see Appendix: boxplot for combined rating in different clusters).

- What are frequent movie genres and genre patterns?

We use Apriori algorithm to detect the frequent itemsets and frequent rules of movie genres.

- How do different factors impact the box office?

  Through linear regression, we saw that
  - If a movie runs in one more theater, the box office will increase $20.2K.
  - If a movie runs for one more day in movie theater, the box office will increase $53.8K.
  - If the movie gets one more vote on imdb, the box office will increase $186.
  - As for how the movie is rated, in general, movies rated 'PG' receive $3.96M more box office than 'Not Rated'; movies rated 'PG-13' receive $5.8M less box office than 'Not Rated'; movies rated 'R' receive $1.2M less box office than 'Not Rated'.
  - Movie rating has NO predictive power towards the box office. People do not always choose to go to a movie because of its reputation.
- What factors best predict the level of box office of a movie?

- Our best performing classification models are Random Forest, Decision Tree and SVM(with RBF Gaussian Kernel). When we select different feature combinations to construct these models, their accuracy varies. However, when we look at the accuracy of different models closely, it seems there is a upper limit of the accuracy at approximately 0.71-0.72.
  KNN model also has good performance of accuracy higher than 0.65.
  Naive Bayes model has poor performance comparing to other models, the accuracy of different Naive Bayes models are from 0.2 to 0.5 and the ROC curves are sometimes below the diagonal line.
- If we expand our feature selection range, that is, add opening box office and # of opening theaters to feature selection to predict final box office. We usually get much higher accuracy at around 0.8 (or even higher). However, that is “cheating” because we use box office (at the opening time) to predict final box office.

Data Cleaning & Description

Data Cleaning:

1. **Turn data in 'Runtime' column to minutes**: We found that in the 'Runtime' col, there are some data with 'hours' and some with 'minutes'. Thus we further cleaned this column for unity.
2. **Get the timespan the movie played in cinemas from 'close time' and 'open time' columns**: Close and Open Time are in date format, which can not be used for analysis. Thus, we take the time difference and create a new variable: Timespan.
3. **Group 'Studio' column: the '**studio' column has 506 categories, which is very hard to use in our analysis. Thus, we group studios that produced fewer than 10 movies as 'indie studios', and studios that produced 10-40 movies as 'mid-sized studios'.
4. **Delete the row if all of its rating values are NAs:** Box office and Ratings are the two most important entities of our data. Therefore we decide that if a movie has none of Metascore, Rotten Tomatoes and imdbRating values, we will delete that row (movie).
5. **Combine all the ratings into combined rating**: Because the three rating values have different distributions(they have different means and standard deviation values), we decide to combine Metascore, Rotten Tomatoes and imdbRating by calculating the mean value of each one's z-score.
6. **Fill the NA in the Timespan, All Theaters, Opening Theaters and Runtime columns**: Since we have already detected these four attributes have NA's, we would like to fill the NA values by using some logical methods. We first tried to fill all NA values by using the mean of the whole column. However, it turns out some popular movies would be filled with small values in the columns of Timespan, All Theaters and Opening Theaters and would be filled NA with the column mean value and vice versa for the unpopular/indie movies. Therefore, we group the movies by their box office rank and year and fill NA values with the mean value of each of these groups.

Data Description:

	Rank	Total Gross	All Theaters	Opening	Opening Theaters	Timespan	Metascore	Rotten Tomatoes	imdb Rating	imdb Votes	combined_rating
Mean	327.86	1.77*10^7	719.14	5.68*10^6	674.75	71.71	5.85	6.32	6.47	42074.8	0.51
Median	319	181550	27	32524	9	69	6	7	6.6	4038.5	0.55
sd	199.66	5.27*10^7	1246.03	1.75*10^7	1221.36	55.67	1.70	2.74	1.05	106813	0.29

Numeric Columns:

Rank: It is the rank of box office each year. There are approximately 600-800 movies each year. So the mean and median is about 320.

Total Gross: It is the box office. The mean is 10 times of the median and the standard deviation is very large. This means the movie market is a winner-take-all market: a few movies have quite high box office while most movies do not.

All Theaters: It is the total number of theaters that screened the movie. Like Total Gross, the mean is much larger than the median, and the standard deviation is large. This means a few movies are very popular, and therefore many theaters would like to screen them. Others are not so successful.

Opening: It is the box office on the opening day. Like Total Gross, the mean is much larger than the median, and the standard deviation is large.

Opening Theaters: It is the number of theaters that screened the movie on the opening day. It is the same situation as All Theaters.

Timespan: It describes how many days the movie was screened in cinema. The mean and the median are close, but the standard deviation is relatively large. The timespan varies in a large range.

Metascore, Rotten Tomatoes, imdbRating: These three are movie ratings from different websites. Rescaled in 0 to 100, these three rating scores seem similar. However,Rotten Tomatoes has a larger standard deviation than the other two, which means people on this website tend to rate more extremely. They score ‘good’ movies higher and score ‘poor’ movies lower.

combined_rating: This is the average z-score of Metascore, Rotten Tomatoes, and imdbRating of each movie. It is close to a normal distribution.

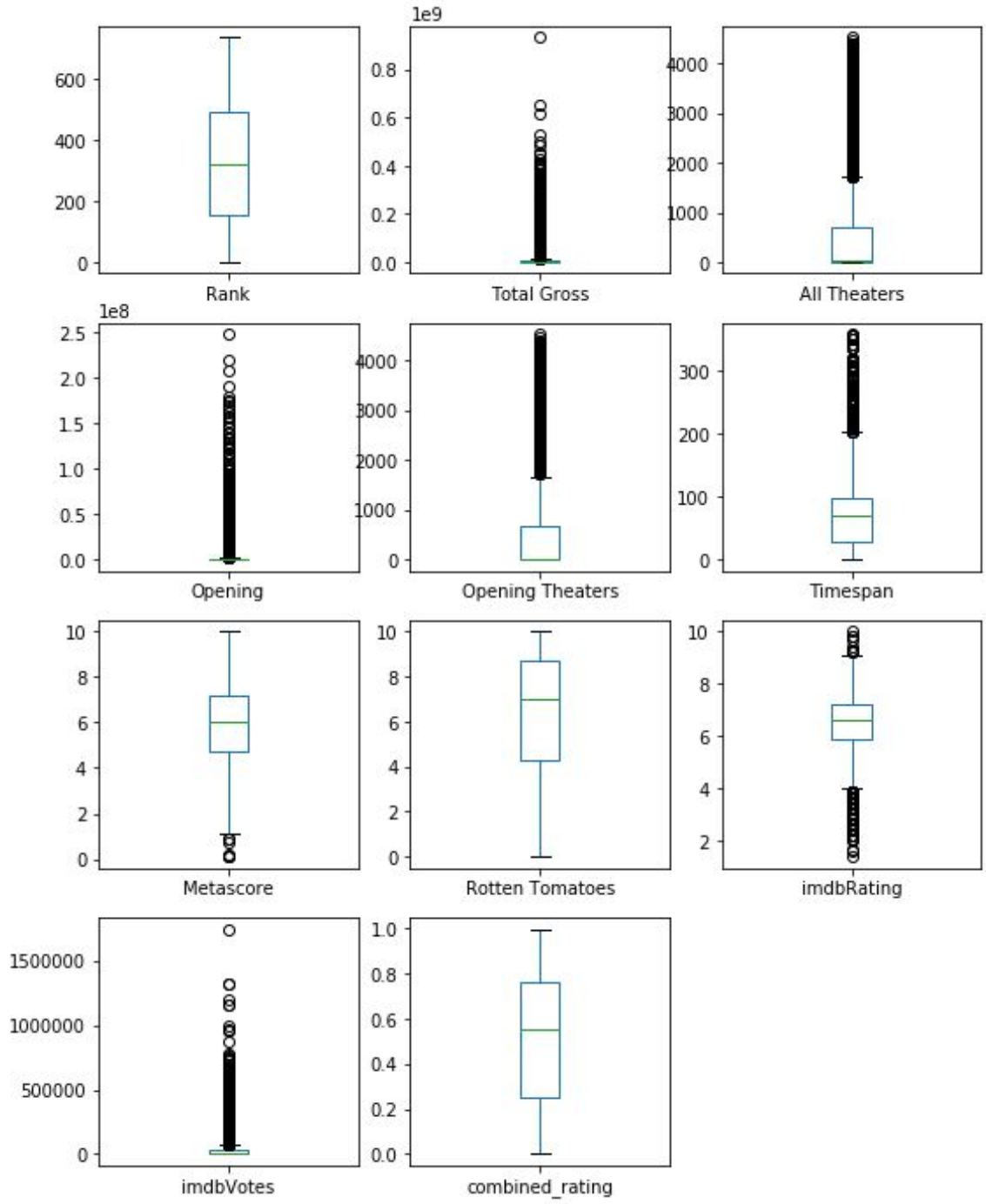
Categorical Columns:

Studio_new: The mode is ‘Mid_sized'. This means most movies are produced by studios that produced 10-40 movies over the past 7 years, which are mid-sized studios.

Rated: The mode is 'NOT RATED’. This means that most movies are not rated.

Outlier Detection:

Outliers can be detected from data description and boxplot.



Attributes that may contain outliers:

Total Gross, All Theaters, Opening, Opening Theaters, Timespan, Metascore, imdbRating, imdbVotes

Total Gross, All Theaters, Opening, Opening Theaters, imdbVotes:

As we can see from the boxplots, the "normal values" are mostly within a small range and are relevantly low, while the outliers may be quite large. The movie market is a typical "winner-take-all" market. These outliers are a reflection of what happened in the real world. For example, *Star Wars: The Force Awakens*, *Jurassic World*, and *Star Wars: The Last Jedi* are so successful that their total gross and opening exceed all other movies. *Despicable Me 3*, *The Twilight Saga: Eclipse*, and *The Dark Knight Rises* listed as the top 3 in all theaters and opening theaters. This means they are expected to be outstanding movies, and therefore a great many of theaters would like to screen them. *Inception*, *The Dark Knight Rises*, and *The Dark Knight (2012 re-release)* are three movies with the most votes. These are all excellent movies that the audience would like to watch and comment on. Thus, we keep the original values of these 'outliers'.

Metascore and imdbRating:

The outliers are those movies that really scored really low. We think it's reasonable to assume movies have very low score because of poor quality. Thus, we recognize them as natural outliers and we keep them.

Timespan:

1. We calculate this column from subtracting the 'Opening Date' from the 'Close Date' in the original data. However, in the raw data,the 'Opening Date' and 'Close Date' columns only provide month-date information, without 'year'. So a few movies that are screened for over one year may have mistakes in its timespan. We detect them and other outliers through following method:Select movies that have a timespan less than or equal to 5 and ranked as top 200. Find news and information on each movie and deal with the timespan one by one.
2. The *Dark Knight (2012 re-release)* only released 1 day but it has a really large total gross. We check it and find out that this is a re-released film released by AMC, and it only screened for one week. The original one was so successful that the re-released one also has a large audience in such a short time.Thus, we change the timespan to 7 from 1. The movie *20 Feet from Stardom* is a really successful documentary movie that won the 86th Academic Awards and the 2015 Grammy Awards. It was on the screen for over one year, so we add 365 to the timespan.The movie *Batman: The Killing Joke* has some problem with its timespan. It was actually on the screen for over one year. So we add 365 to the timespan.
3. The movie *Newsies: The Broadway Musical* is a music drama. It is screened for limited time -- 5 days. Before the screening, there were a slew of advertisements, so the opening is up to 1 million USD. Within just 5 days, it ended up with a total gross of $2,545,060. Thus we change the timespan to 5 from 1.

Binning Strategy:

We bin Total Gross, All Theaters, Opening, Opening Theaters, Timespan, Metascore, Rotten Tomatoes, imdbRating, combined_rating, imdbVotes, that is, we bin every numeric column except the rank column. For 'Total Gross', 'All Theaters', 'Opening', 'Opening Theaters', and 'imdbVotes', the distributions of these attributes are extremely skewed, so we use exponential binning strategy:

- Total Gross: [0, 10**4, 10**5, 10**6, 10**7, 10**8, 10**9]
- All Theaters: [0, 10, 100, 1000, 10000]
- Opening: [0, 3*10**3, 3*10**4, 3*10**5, 3*10**6, 3*10**7, 3*10**8]
- Opening Theaters: [0, 5, 50, 500, 5000]
- imdbVotes: [0, 5*10**2, 5*10**3, 5*10**4, 5*10**5, 5*10**6]

For ‘Timespan’, ‘Metascore’, ‘Rotten Tomatoes’, ‘imdbRating’, ‘combined_rating’, we create bins that make intuitive sense:

- Timespan: [0,7, 30, 60, 90, 120, 500]

We bin it as the extreme-short movie, short movie, medium-short movie, medium length movie, medium-long movie, long movie, extreme-long movie.

- Metascore, Rotten Tomatoes, imdbRating: [0,3,6,8,10]

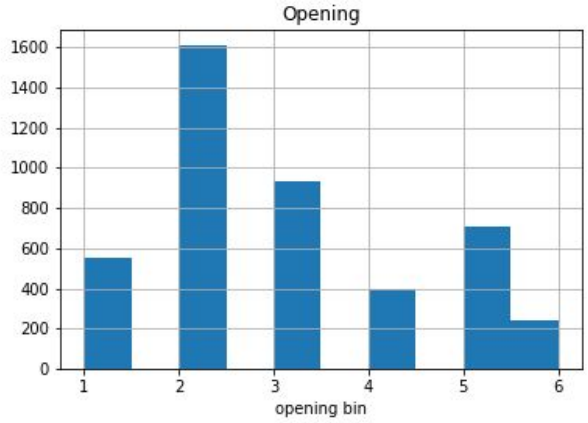
We bin them as the very-low-rated movie, low-rated movie, high-rated movie, very-high-rated movie.

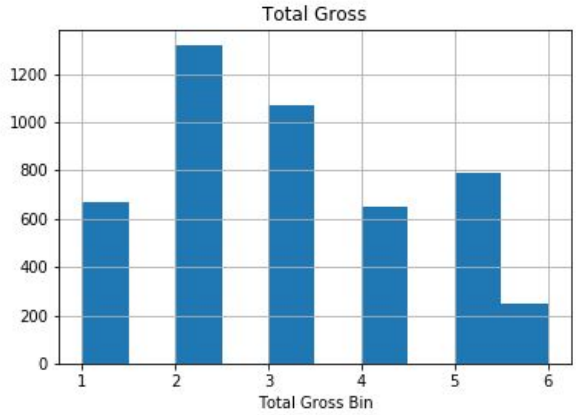
- Combined_rating: [0, 0.3, 0.6, 0.8, 1]

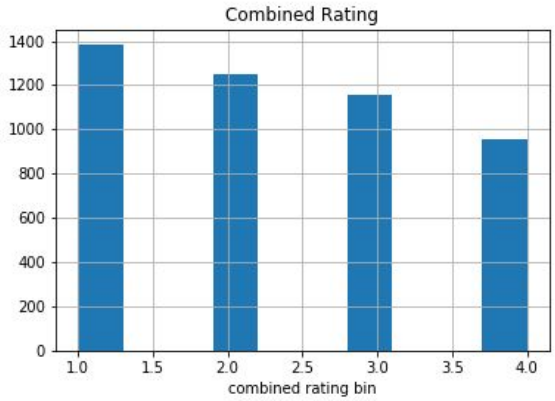
The binning strategy is the same as the previous one. The only difference is the data scale.

Exploratory Analysis

Histogram and Correlations:



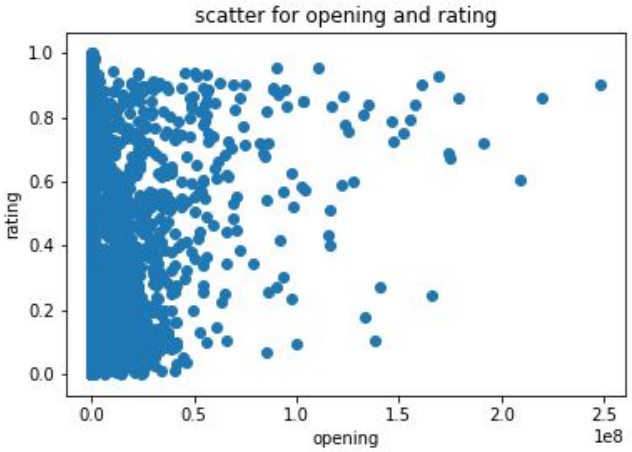


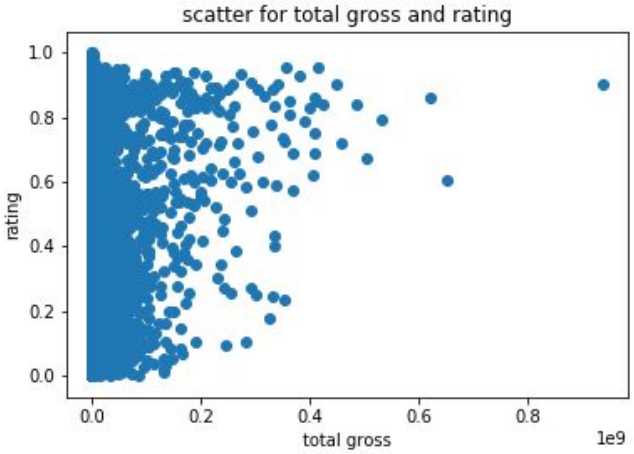


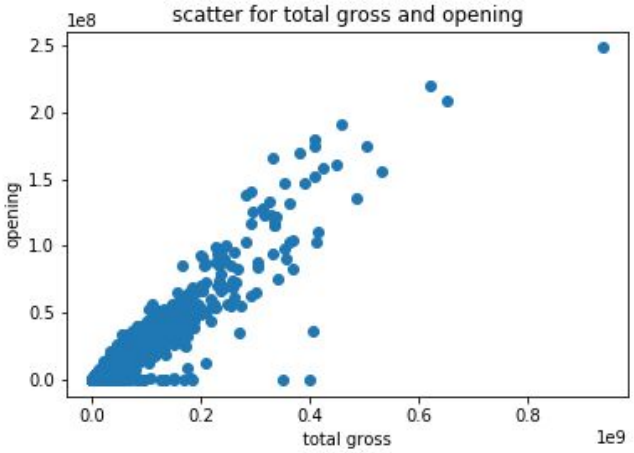
These are three histograms for attributes: total gross bin, opening bin, and combined rating bin. The first two histograms are both right skewed. They share a similar pattern in distribution. This finding intuitively makes sense because opening box office should be a strong indicator of final box office (total gross). For

the histogram of rating, the histogram shows that there are more movies with lower rating than higher rating.

Scatterplots:



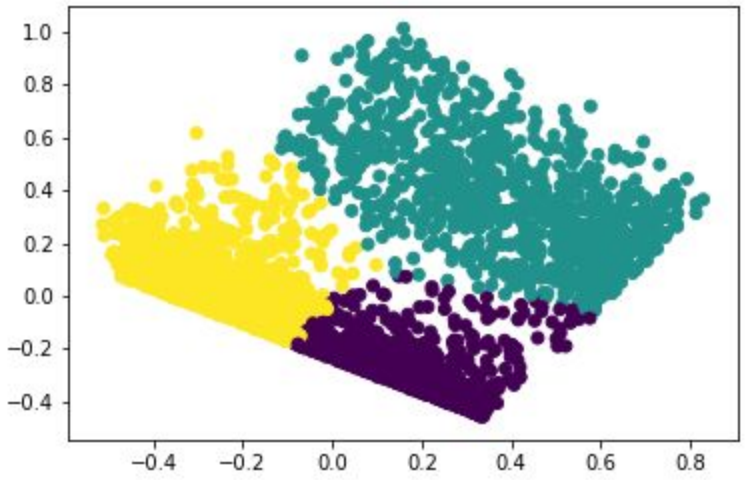




The scatterplots for total gross vs. rating and opening vs. rating both show a lot of interesting things. The graph shows that no matter how much the box office is, there will be very high rating movies. However, if the movie's box office is high, then the rating for this movie is usually high (at least not low). In addition, for the scatterplot of opening vs. total gross, we find that movies with large opening box office usually have large total box office.

Clustering:

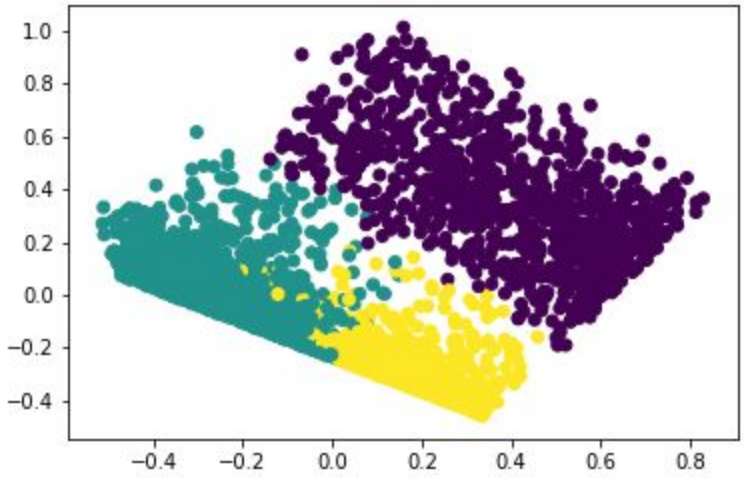
K-Means Clustering:



For k-means clustering, we choose to cluster attributes total gross, timespan, all theaters, and combined rating. We get the graph above for the 3 clusters, which performs the best. The Calinski-Harabaz score is 4782.62 for k=5, 4678.86 for k=4 and 5043.26 for k=3. Since the higher the Calinski-Harabaz score, the better quality of the clustering, we choose k=3 to do further analysis. By comparing boxplots, we find that the distribution of rating in each cluster is different.

The mean values of rating of the three clusters are 0.22, 0.40 and 0.74. We can see that those three clusters are well separated, which means this method does fine.

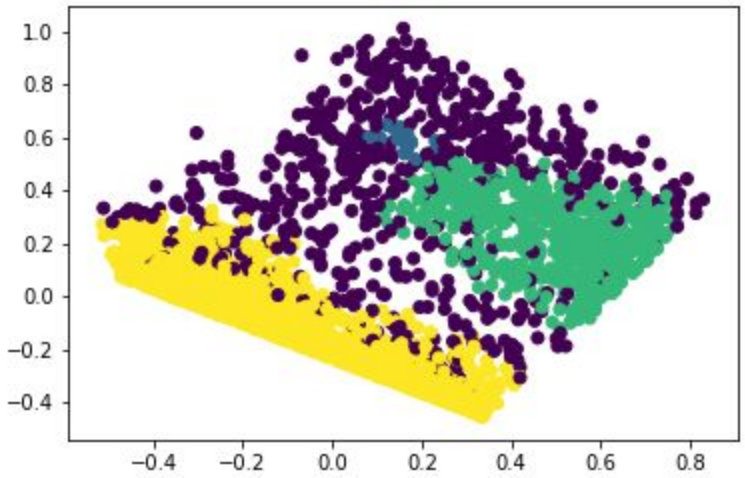
Hierarchical Clustering:



For hierarchical clustering, we also decide to cluster total gross, timespan, all theaters, and combined rating. The graph above is quite similar to the k-clustering plot. By trying k=5, k=4 and k=3, calculate the Calinski-Harabaz score 4093.26, 4164.96 and 4679.16. Therefore we decide to choose k=3 because its score is the highest. By comparing Calinski score, we believe the performance of hierarchical -clustering is poorer than for k-means clustering.

For hierarchical clustering, the ratings for three different clusters are 0.71, 0.39, and 0.20. We could see that the result is almost the same as k-means clustering.

DBSCAN:



The attributes we use for dbscan clustering are total gross, timespan, all theaters, and combined rating. We have tried different parameters for maximum distance between two samples for them to be considered as in the same neighborhood(eps) and the number of samples in a neighborhood for a point to be

considered as a core point(min_samples). We find that when eps=0.07 and min_samples=10, the clusters looks the best and there are 4 clusters. The Calinski score for dbscan is 1104.47 which is not as good as others.

For DBSCAN, we have four clusters and ratings are around 0.72, 0.58, 0.54, and 0.24 for each cluster.

Association Rules

Question: Many movies have more than one genre. Use Apriori Algorithm, find out what genres appear most frequently together, and explain the findings.

Method: Parse the genre column to sparse "one of n" format, use apriori algorithm(sklearn mlxtend class) to generate frequent itemsets and association rules.

Choose 3+ different support levels: Set support value to a very small value (we tried 0.001, 0.0001 and 0.00001) in apriori function, sort the itemsets by their support value to find that the minimum and maximum support levels among all itemsets are 0.000211 and 0.509. In order to control the number of frequent sets less than 30, we choose to set the lowest support level to 0.03. On the other hand, in order to keep enough frequent association rules, we decide to set the largest support level to 0.07.

Result (Interesting Findings):

1. We usually think that Adventure or Action are the most popular movie genres, because they make the money(box office) , which implies they are favored by the majority of the audience. A report from Statista proves this point. [1] In the report, the movie genre Adventure makes most money; the second place is Action and the third place is Drama. However, it turns out that the from 2010 to 2017, the most frequent movie genre is Drama with a support level larger than 0.5, while Action movie and Adventure movies only have the support level of 0.16 (the 4th place) and 0.1(the 9th place), respectively.
2. Some top frequent genre combinations (association rules) with high confidence and conviction value are:
    - {Romance} -> {Drama, Comedy}
    - {Adventure} -> {Action}
    - {Biography} -> {Documentary}

    The high confidence value and conviction value of these rules imply these genres are bounded closely, and the consequent (genre on the right side) is highly dependent on the antecedent (genre on the left side).When it is a romance movie, the movie genre is very likely to be drama and comedy. (Romance movie directors like to give their stories a happy ending.) When it is an adventure movie, it is also very likely to be an action movie. When it is a biography movie, it is also very likely to be a documentary movie. These rules intuitively make sense.

---

1 https://www.statista.com/statistics/188658/movie-genres-in-north-america-by-box-office-revenue-since-1995/

Predictive Analysis

Hypothesis Testing I:

Question: Is there a difference between the two mainstream rating systems (imdb Rating vs. Rotten Tomatoes)?

Null Hypothesis: Rotten Tomatoes - imdbRating= 0

Alternative Hypothesis: Rotten Tomatoes - imdbRating $\neq$ 0

Method: We use **independent t-test** to test this hypothesis. We choose this method because we are only comparing two rating systems and independent t-test is suitable for this kind of test.

Result: We find that for 77% of records with both ratings, there is actually a significant difference (p_value= 3.5e-05) between the two ratings. This proves that different platforms do review movies differently, and thus we reject the hypothesis.

Hypothesis Testing II:

Question: Is there a significant difference in terms of box office for different years/ studios/ ratings(PG-13, R, etc)?

Null Hypothesis: There is no difference between different years/studios/ratings.

Alternative Hypothesis: There is significant differences between different year/studios/ratings.

Method: We use **Anova Test** to test these hypotheses. We choose this method because anova is suitable for testing differences between multiple categories.

Result: We find the following:

- Box office by year: We conduct an anova analysis on box office by year, and we find that there's no significant difference between each year (p_value = 0.94), setting the threshold at 0.05. Thus the movie industry does not have drastic change in terms of box office over the years, and we accept the null hypothesis.
- Box office by studios: We further group the studios into 4 categories: super, large, mid_sized and indie for ease of analysis. We find that the box office for different sized studios do have a significant difference (p_value= 3.5e-89) and we reject the null hypothesis, i.e. studios of different size can impact the box office.
- Box office by rated(PG, PG-13, etc): We further group the studios into 5 categories: Other', 'PG', 'PG-13', 'R', 'NOT RATED' for ease of analysis. We find that the box office for different rating do have a significant difference (p_value= 2.2e-145) and we reject the null hypothesis. Because rating can restrict the certain people from watching, it can impact the box office.

Hypothesis Testing III:

Question: Is there a linear relationship between the independent variables (rated, studios, ratings, etc.) and the dependent variable Y (box office)?

Null Hypothesis: There is no linear relationship between independent variables (rated, studios, ratings, etc.) and the dependent variable Y (box office).

Alternative Hypothesis: There exists a linear relationship between independent variables (rated, studios, ratings, etc.) and the dependent variable Y (box office).

Method: We use **linear regression** for this question. We choose linear regression because box office is a continuous variable. Method details are explained below:

a. Feature Selection
    i. **Numeric columns:**
        1. Looking at correlation above, we exclude variables with too low a correlation with box office (min. Threshold at 0.2) .
        2. Then, we use RFE feature selection function to rank the numeric variables, and got the following ranking (Timespan > All theaters> imdbVotes> Opening Theaters). Thus, we would have an idea of which variables to discard if needed (noted that it's possible that we can include all)
        3. Finally, 'All theaters' and 'Opening theaters' are highly correlated (corr= 0.96), we include only 'All theaters'(since it has a higher ranking) to avoid overfitting.
    ii. **Categorical columns**: From anova, we know that studios & rated columns potentially have impact on box office. Thus, we create dummy variables for those two columns.
    iii. **Features Selection**: From above, we create 8 combinations of features, and will run regression for each combination to see which one performs the best.

b. Model Evaluation:

We use a cross validation method to evaluate the goodness of a model. We take two measures: **rmse & cross validated R^2.** A low rmse and a high cross validated R^2 indicate a good model.

c. Regression Result:

The result shows that the slope is not 0 and we reject the null hypothesis. Testing the 8 features above, we find out that Model 5 and Model 7 give very similar test statistics (Cross Validated R^2 both equal to 0.6). However, Model 5 include one more variable than Model 7. Using Occam Razor, we think Model 7 is the best predicting model. It includes the following variables:

	Variable	Var Name	Coef.
1	# of theaters	All Theaters	20232
2	Length that the movie runs in cinema	Timespan	53815
3	# of votes on imdb	imdbVotes	186
4	How the movie is rated (categories include: Other, PG, PG-13, R, Not Rated)	Rated_Other	3.60e+06
		Rated_PG	3.96e+06
		Rated_PG-13	-5.81e+06
		Rated_R	-1.22e+07

Classification：K-Nearest Neighbors

a. **Feature Selection**: We take a look at the correlation table of each column. We choose columns whose correlation with Total Gross are larger than 0.2. (See correlation table in appendix.) We choose 'Rated', 'Studio_new', 'All Theaters Bin', 'Opening Bin', 'Opening Theaters Bin', 'Timespan Bin', 'imdbVotes Bin'. Then remove those columns, which are highly correlated with each other. 'All Theaters Bin','Opening Bin', 'Opening Theaters Bin' are highly correlated; correlations among them all exceed 0.8. We remove 'Opening Bin' and 'Opening Theaters Bin'. So the features that we select are 'All Theaters Bin','Timespan Bin','imdbVotes Bin','Rated','Studio_new'. According to the correlation value with Total Gross, we created 4 feature groups.
    - Feature group 1=['All Theaters Bin','Timespan Bin']
    - Feature group 2=['All Theaters Bin','Timespan Bin','imdbVotes Bin']
    - Feature group 3=['All Theaters Bin','Timespan Bin','imdbVotes Bin','Rated']
    - Feature group 4=['All Theaters Bin','Timespan Bin','imdbVotes Bin','Rated','Studio_new']

b. **Result**: We found that the best model (with the highest accuracy at 0.691) is using feature group 4.: # of theaters ('All theaters Bin'), length that a movie runs in cinema('Timespan Bin'), # of imdbVotes ('imdbVotes Bin'), how the movie is rated ('Rated'), studio of movie(Studio_new).

Classification：Decision Tree

a. **Feature Selection:** Use the same feature as KNN.
    - Feature group 1=['All Theaters Bin','Timespan Bin']
    - Feature group 2=['All Theaters Bin','Timespan Bin','imdbVotes Bin']
    - Feature group 3=['All Theaters Bin','Timespan Bin','imdbVotes Bin','Rated']
    - Feature group 4=['All Theaters Bin','Timespan Bin','imdbVotes Bin','Rated','Studio_new']

b. Result: We found that the best model has the highest accuracy at 0.713, which is feature group 3.: # of theaters ('All theaters Bin'), length that a movie runs in cinema('Timespan Bin'), # of imdbVotes ('imdbVotes Bin') and how the movie is rated ('Rated').

Classification: Naive Bayes (Gaussian, Multinomial and Bernoulli)

a. **Feature Selection:** Use Chi square test (implemented by sklearn select k best function) to score the attributes and select the k attributes with highest scores.
    - Feature group 1=['All Theaters Bin', 'imdbVotes Bin']
    - Feature group 2=['Studio_new', 'All Theaters Bin', 'imdbVotes Bin]
    - Feature group 3=['Studio_new', 'All Theaters Bin', 'Timespan Bin', 'imdbVotes Bin']
    - Feature group 4=['Rated', 'Studio_new', 'All Theaters Bin', 'Timespan Bin', 'imdbVotes Bin']

b. Result: Overall Naive Bayes models have poorer performance than other models.

Gaussian: best model has the highest accuracy at 0.552, which is feature group 3.: # of theaters ('All theaters Bin'), length that a movie runs in cinema('Timespan Bin'), # of imdbVotes ('imdbVotes Bin') and studio of movie(Studio_new).

Multinomial: best model has the highest accuracy at 0.440, which is feature group 1: # of theaters ('All theaters Bin') and # of imdbVotes ('imdbVotes Bin').

Bernoulli: best model has the highest accuracy at 0.262, which is feature group 4.: # of theaters ('All theaters Bin'), length that a movie runs in cinema('Timespan Bin'), # of imdbVotes ('imdbVotes Bin'), studio of movie(Studio_new) and how the movies is rated ('Rated').

Classification: SVM (Linear, Linear Kernel, RBF(Gaussian) Kernel)

a. **Feature Selection:** Same strategy with Naive Bayes
b. Result: Overall SVM models have pretty good performance comparing to other models.

SVM with Linear Kernel: best model has the highest accuracy at 0.719, which is feature group 4.: # of theaters ('All theaters Bin'), length that a movie runs in cinema('Timespan Bin'), # of imdbVotes ('imdbVotes Bin'), studio of movie(Studio_new) and how the movies is rated ('Rated').
Linear SVM: best model has the highest accuracy at 0.560, which is feature group 1: # of theaters ('All theaters Bin') and # of imdbVotes ('imdbVotes Bin').
RBF (Gaussian) SVM: best model has the highest accuracy at 0.723, which is feature group 3.: # of theaters ('All theaters Bin'), length that a movie runs in cinema('Timespan Bin'), # of imdbVotes ('imdbVotes Bin') and studio of movie(Studio_new).

Classification: Random Forest

a. **Feature Selection**: We use feature_selection function from sklearn, and it automatically calculates the importance of each of the variables, as shown below. Note that the 'opening bin' variable ($of box office on the opening day) is the best predictor- but when we apply the model to another movie, we won't know the opening box office many times. Thus, we decided to create two models: one with 'Opening' and one without 'Opening'. We then tested the k most important features for k in 1:4 with 'opening', and for k in 1:3 without 'opening' （see feature importance table in appendix)
b. **Model Evaluation:** We use the cross validation method and use cross-validated accuracy score to evaluate the model. The higher the accuracy the better.
c. **Result**:
    i. Without opening, we find that the best model (with the highest accuracy at 0.717) contains three variables：
        1. # of theaters: All Theaters Bin
        2. Length that the movie runs in cinema: Timespan Bin
        3. The studio that produced the movie: Studio_new
    ii. With 'opening', we find that the best model (with the highest accuracy at 0.793) contains the following:
        1. $Box office at the opening day: Opening Bin
        2. # of theaters: All Theaters Bin
        3. Length that the movie runs in cinema: Timespan Bin

Appendix

Correlation Table:

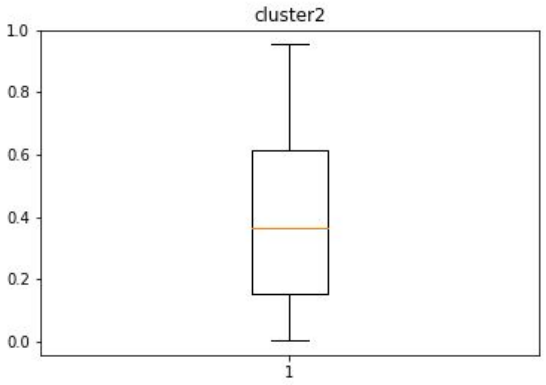
	Total Gross Bin	Rated	Studio_new	Year_x	All Theater Bin	Opening Bin	Opening Theaters Bin	Timespan Bin	Metascore Bin	Rotten Tomatoes Bin	combined_rating Bin	imdbVotes Bin
Total Gross Bin	1	0.309282	0.200042	0.026007	0.919025	0.912929	0.763484	0.465673	-0.05231	-0.13419	-0.04833	0.748709
Rated	0.309282	1	0.13435	0.036128	0.379801	0.302343	0.269822	-0.06036	-0.16978	-0.23566	-0.21753	0.448081
Studio_new	0.200042	0.13435	1	-0.02485	0.194693	0.198827	0.183943	0.051779	-0.07323	-0.10 09	-0.08934	0.196449
Year_x	0.026007	0.036128	-0.02485	1	0.080987	0.021621	0.027364	-0.08004	0.070299	0.09555	0.074777	-0.05878
All Theater Bin	0.919025	0.379801	0.194693	0.080987	1	0.90809	0.836241	0.289623	-0.18841	-0.27049	-0.19497	0.740469
Opening Bin	0.912929	0.302343	0.198827	0.021621	0.90809	1	0.901751	0.300729	-0.20491	-0.27626	-0.21 02	0.710583
Opening Theaters Bin	0.763484	0.269822	0.183943	0.027364	0.836241	0.901751	1	0.097145	-0.36451	-0.41562	-0.37018	0.598407
Timespan Bin	0.465673	-0.06036	0.051779	-0.08004	0.289623	0.300729	0.097145	1	0.336398	0.30591	0.368121	0.295037
Metascore Bin	-0.05231	-0.16978	-0.07323	0.070299	-0.18841	-0.20491	-0.36451	0.336398	1	0.790842	0.8171	-0.03876

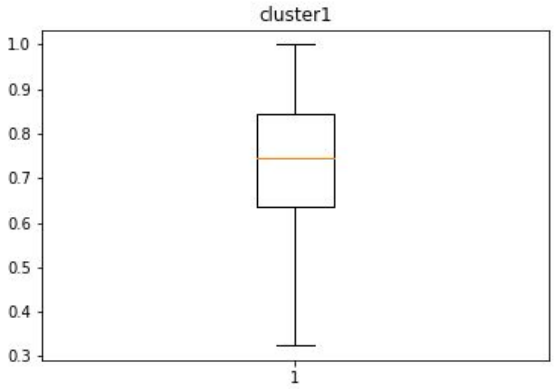
Rotten Tomatoes Bin	-0.13419	-0.23566	-0.1009	0.09555	-0.27049	-0.27626	-0.41562	0.30591	0.790842	1	0.879501	-0.13638
combined_rating Bin	-0.04833	-0.21753	-0.08934	0.074777	-0.19497	-0.2102	-0.37018	0.368121	0.8171	0.879501	1	-0.05705
imdb Votes Bin	0.748709	0.448081	0.196449	-0.05878	0.740469	0.710583	0.598407	0.295037	-0.03876	-0.13638	-0.05705	1

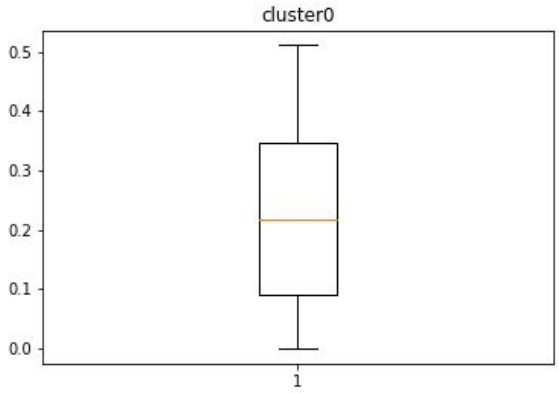
Random Forest Feature Importance

Variable	Importance
Rated	0.05
Studio_new	0.09
Year_X	0.07
All Theaters Bin	0.27
Opening Bin	0.20
Opening Theaters Bin	0.05
Timespan Bin	0.11
Metascore Bin	0.03
Rotten Tomatoes Bin	0.03
Comibined_rating Bin	0.04
Imdb Votes Bin	0.06

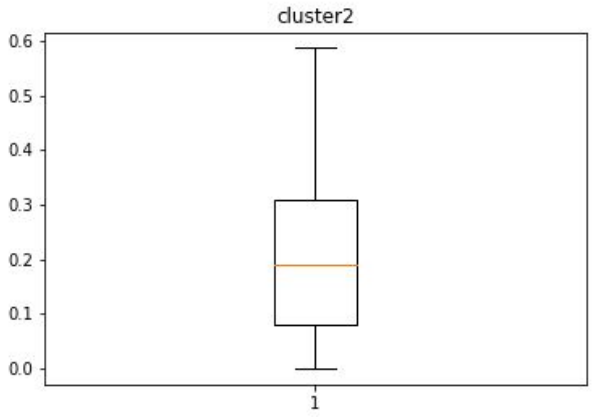
Boxplot for ratings in different clusters for K-Means:

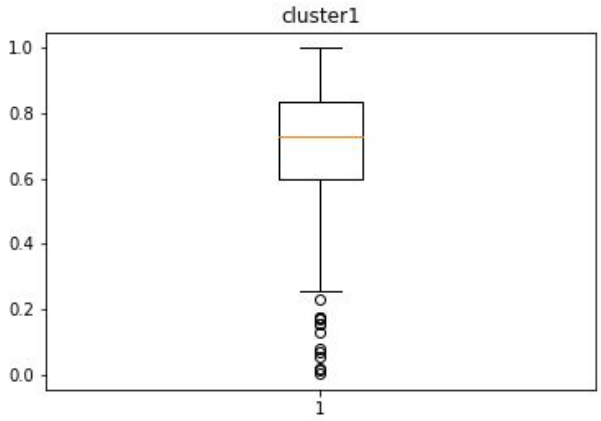


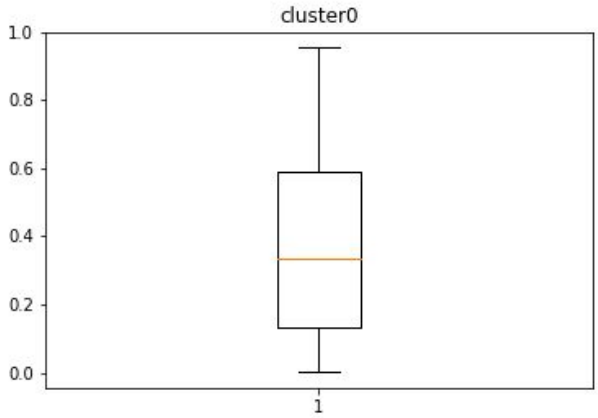




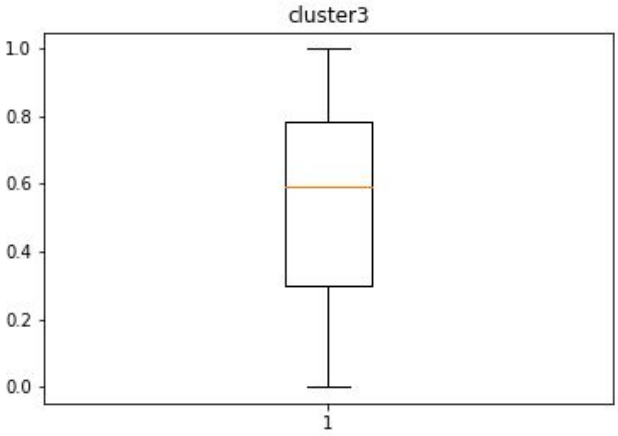
Boxplot for ratings in different clusters for Hierarchical:

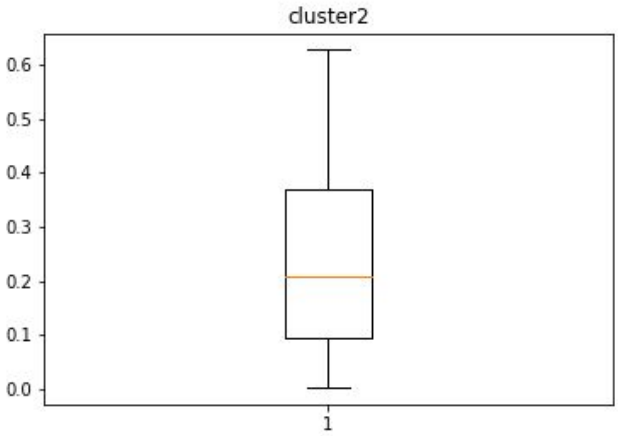


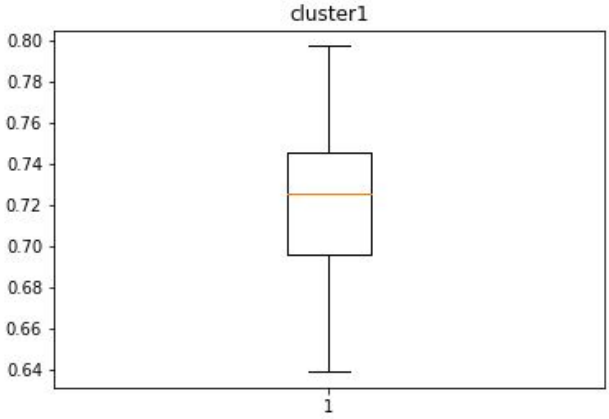


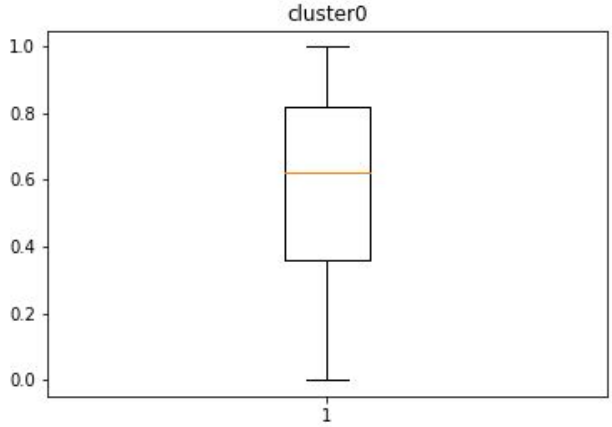


Boxplot for ratings in different clusters for DBSCAN:


cluster3
1.0
0.8
0.6
0.4
0.2
0.0
1


cluster2
0.6
0.5
0.4
0.3
0.2
0.1
0.0
1


cluster1
0.80
0.78
0.76
0.74
0.72
0.70
0.68
0.66
0.64
1


cluster0
1.0
0.8
0.6
0.4
0.2
0.0
1