CS155 Group Project 2 Report

Halle Blend, Georgia Malueg, Rachael Kim, Jayden Nyamiaka

February 2023

NOTE: We used 1 late hour for this report.

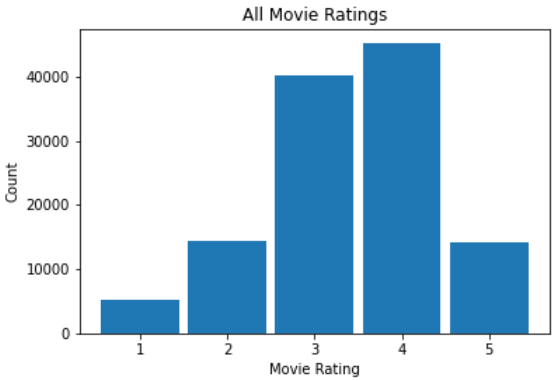
1 Introduction [5 Points]

1. **Team Name:**
   Everybody Likes Jayden
2. **Group Members:**
   Halle Blend, Georgia Malueg, Rachael Kim, Jayden Nyamiaka
3. **Colab Link:**
   Basic Visualizations
   Matrix Factorization Methods
4. **Piazza Link:** Visualization Submission
5. **Division of Labor:**
   Halle: Basic Visualizations. Visualization Helper Methods. Matrix Factorization 1.
   Georgia: Matrix Factorization Visualization 2. Piazza Post.
   Rachael: Matrix Factorization Visualization 3.
   Jayden: Limitations and Extra Basic Visualizations. Matrix Factorization Visualization 2.
6. **Packages Used**
   - numpy
   - matplotlip - pyplot
   - pandas
   - sklearn
   - SVD
   - surprise

2 Basic Visualizations [20 Points]

Part 2 Code

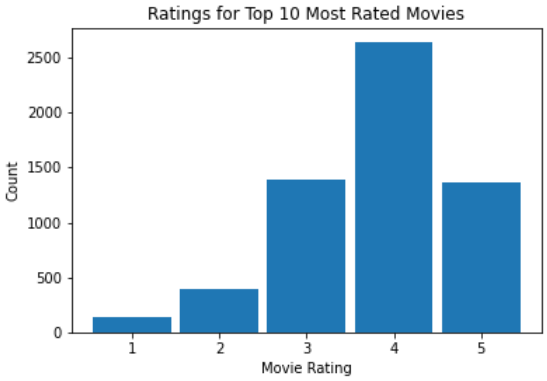
All Ratings in the MovieLens Dataset



General Observations
The data is skewed to the right such that the rating with the highest count is 4 and the rating with the lowest count is 1.

Results vs Expectation
The results are close to what we expected. We would assume that most ratings would be positive with lower counts of the most extreme ratings (1 and 5). Our thinking was that movies usually don't make it through production unless the producers are reasonably confident they'll be liked by viewers, and people are less likely to give extreme ratings since a movie is rarely entirely good or entirely bad.

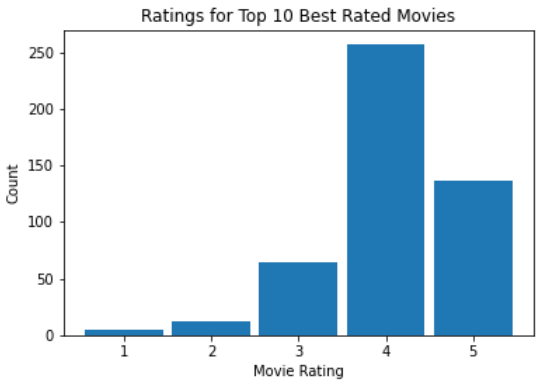
All Ratings of the Ten Most Popular Movies



General Observations
The distribution is somewhat similar to that of the entire MovieLens dataset except that it's skewed more to the right. There are higher counts of 4 and 5 compared to the counts for 1 and 2. Additionally, we see that around 6,000 ($\approx 5\%$) of all ratings account for the 10 most rated movies.

Results vs Expectation
This distribution being somewhat similar to the whole dataset is expected because these 10 movies will have the largest impact on the overall rating distribution. Additionally, it is expected that we would have higher counts of 4 and 5 along with lower counts of 1 and 2 as the most popular movies would have more positive reviews due to mass appeal. If a movie becomes popular, it's usually because many people like it such that many people would rate it highly, which explains why the ratings are so positive.

All Ratings of the Ten Best Movies



General Observations
We see that this distribution is mostly high ratings of 4 and 5 with less lower ratings of 1 and 2.

Results vs Expectation
It is expected that most of the ratings would be 4s and 5s for the best rated movies. However, it is unexpected that some of the best rated movies still have some ratings of 1s and 2s. This shows that not everyone liked the movies that most people rated high.

Compare to Most Popular Movies
It is interesting to compare the total number of ratings for the best rated movies and the most popular movies. We see that the most popular movies have a lot more ratings compared to the best rated movies, and that these movies aren't the same. It shows that popularity and high ratings aren't directly correlated. The popular movies obviously have more ratings while the best movies have a greater percent of more positive ratings (4 and 5); this is as expected.

All Ratings of Movies from Three Genres of your Choice

General Observations
All three distributions are slightly skewed to the right and closely match the overall distribution. We see that the gap between the number of ratings of 3 and 4 relative to each other is largest for the Horror genre (more 3s). Additionally, Comedy has the largest number of reviews by far.

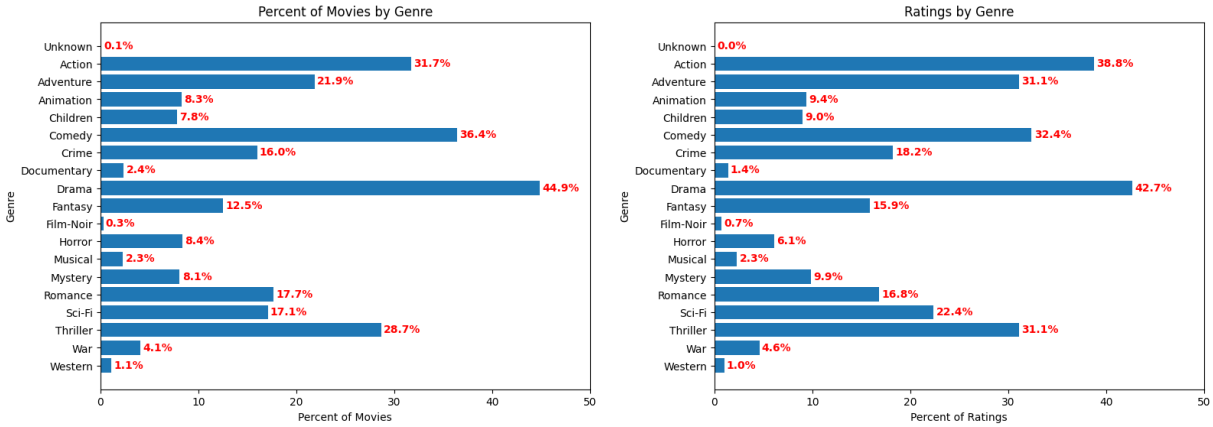
Results vs Expectation
These distributions are expected. The review counts agree with the fact that the Comedy genre is very popular such that comedy movies are produced and rated more. Additionally, the rating distributions for Sci-Fi and Horror reflect the frequent production of some low-budget Sci-Fi and Horror movies, which are low quality and therefore review poorly. We also expected for the average film to have a rating of 3, which is about what happens here (around 3.5). Lastly, the ratings for these genres are slightly less positive compared to the overall distribution (higher percent of 3s). We believe this is because these are popular genres, so many of the movies produced may be unoriginal and/or low quality, leading to lower ratings on average.

Limitations

These basic visualizations of the dataset provide us with insight into the distribution and structure of our data. We need this to better interpret our models and decide if the models we are making are accurately modeling the data. Additionally, although it wasn't required of us for this project, we usually need to decide which type of model to build given a certain problem; these basic visualizations are crucial in these considerations.

Nevertheless, there are limitations of the basic visualizations we did above. Specifically, even though we are splitting up the movies by different criteria, we are only looking at the ratings. Although the ratings are ultimately what we are trying to predict with our model, we lose a lot of information by not analyzing other characteristics of our data. We are not comparing movies and ratings by genres or checking for any data imbalance, which could potentially lead to bias in our model. In an attempt to mitigate this, we made extra visualizations below.

Extra Basic Visualizations



These graphs show us the break up of movies and ratings by genre side by side. With this, we can see if there's any imbalances in our data. For example, if 20% of movies are of one genre, yet 40% of ratings are by that genre, then we would have a clear imbalance in our data such that movies of that genre are rated much more frequently than other movies. This imbalance could lead to bias in our model such that our model treats that genre in some special way. And, even if it doesn't change the way we train our model, we at least need to know that these imbalances exist so we can better interpret our data and explain any existing trends in our predictions.

From the graphs, we can see that the ratings are about proportionally distributed throughout genres with less than a 10% discrepancy for all genres, so we can be pretty confident that our model won't suffer from bias due to an imbalance in our data. with this, we are able to account for one limitation of the other basic visualizations.

3 Matrix Factorization Methods [60 Points]

Part 3 Code

Method 1

For method 1, we used collaborative filtering to make two matrices: $U \in \mathbb{M}_{k\times m}$ and $V \in \mathbb{M}_{k\times n}$. We used these matrices to represent a matrix $Y \in \mathbb{M}_{m\times n} \cong U^T V$. This matrix Y represents each user's rating for every movie in the dataset where $m = 992$ is the total number of users, $n = 1500$ is the total number of movies, and $k = 20$ is the number of latent factors. The element Y_{ij} represents the user i's rating for a movie j. In our dataset, not all users rate every movie. This leads to unknown rating values in the dataset. Therefore, we use collaborative filtering to learn a latent representation of movies U and users V to explain observed ratings, predicting all users ratings of every movie. We learn a latent representation such that

$$\arg\min_{U,V} \frac{\lambda}{2}\left(\|U\|_F^2 + \|V\|_F^2\right) + \frac{1}{2}\sum_{i,j}\left(y_{ij} - u_i^T v_j\right)^2$$

where u_i^T and v_j^T are the i^{th} and j^{th} rows of U and V respectively. First, we initialize U and V matrices by randomly drawing samples from a uniform distribution between $(-0.5, 0.5)$. Next, for each epoch, we randomly iterate through each data point (i, j, y_{ij}) in the training set, updating u_i and u_j by stochastic gradient descent (SGD). We calculate the gradients as follows:

$$\delta u_i = \lambda u_i - v_j(y_{ij} - u_i^T v_j)$$

$$\delta v_j = \lambda v_j - u_i(y_{ij} - u_i^T v_j)$$

We then update $u_i = u_i - \eta * \delta u_i$ and $v_j = v_j - \eta * \delta v_j$ with learning rate η. We implemented an early stopping condition and stop at epoch t if $\Delta_{t-1,t}/\Delta_{0,1} \leq \epsilon = 0.0001$ where $\Delta_{t-1,t}$ is the loss reduction between epoch t and $t-1$ and $\Delta_{0,1}$ is the loss reduction between before training and after training the first epoch. We decided to choose this stopping condition because we can reasonably presume that if the loss is reducing by less than $\epsilon = 0.0001$, then the model has converged close enough and training any more is simply unnecessary computation. We have practical proof from the implementation in HW 5 that this is true.

Method 2

Method 2 is very similar to method 1 except we incorporate bias vectors $\vec{a}$ of length M and $\vec{b}$ of length N. These bias vectors model global tendencies for each user and movie such that the ith element of $\vec{a}$ is the bias of user i and the jth element of $\vec{b}$ is the bias for movie j. Apart from these two bias vectors, all other representations U, V, Y are the same. Given these, we learned a latent representation such that

$$\arg\min_{U,V,a,b} \frac{\lambda}{2}(\|U\|_F^2 + \|V\|_F^2 + \|a\|^2 + \|b\|^2) + \frac{1}{2}\sum_{i,j}((y_{ij} - \mu) - (u_i^T v_j + a_i + b_j))^2$$

where u_i^T and v_j^T are the i^{th} and j^{th} rows of U and V respectively, a_i and b_j are the i^{th} and j^{th} elements of $\vec{a}$ and $\vec{b}$ respectively, and μ is the mean of the labels (ratings) of the training dataset. Same as method 1, we initialize the elements of U, V, $\vec{a}$, and $\vec{b}$ to randomly drawn samples from a uniform distribution between (-0.5, 0.5). We then use SGD on each data point (i, j, y_{ij}) and update our model using the following gradients taken from the optimization of the latent representation:

$$\delta u_i = \lambda u_i - v_j((y_{ij} - \mu) - (u_i^T v_j + a_i + b_j))$$

$$\delta v_j = \lambda v_j - u_i((y_{ij} - \mu) - (u_i^T v_j + a_i + b_j))$$

$$\delta a_i = \lambda a_i - ((y_{ij} - \mu) - (u_i^T v_j + a_i + b_j))$$

$$\delta b_j = \lambda b_j - ((y_{ij} - \mu) - (u_i^T v_j + a_i + b_j))$$

Our model is thus updated accordingly via $u_i = u_i - \eta * \delta u_i$, $v_j = v_j - \eta * \delta v_j$, $a_i = a_i - \eta * \delta a_i$, $b_j = b_j - \eta * \delta b_j$ with learning rate η. The stopping condition to stop at epoch t if $\Delta_{t-1,t}/\Delta_{0,1} \leq \epsilon$ is the same implementation as the method 1 with $\epsilon = 0.0001$. Since we know this condition and ϵ work for method 1 as a good indication of convergence, we use it again for method 2. However, it is worth nothing that the loss calculation used to compute loss reduction Δ is different than method 1. Instead the error function used is the equation that defines our latent representation:

$$\frac{\lambda}{2}(\|U\|_F^2 + \|V\|_F^2 + \|a\|^2 + \|b\|^2) + \frac{1}{2}\sum_{i,j}((y_{ij} - \mu) - (u_i^T v_j + a_i + b_j))^2$$

Once trained, user i's rating of the movie j is predicted by $\mu + u_i^T v_j + a_i + b_j$. Therefore, using method 2, we include the global tendency bias vectors $\vec{a}$ and $\vec{b}$ in our model predictions and include them in our regularization. This should allow us to model the tendencies of each user and movie, and thus, we predicted that this would result in a better performing model.

Method 3

For method 3, we used surprise library of Python to train our model (as we were advised to in the guide). We used surprise SVD to train our model. The prediction r_{ui} of user u's rating of movie i is calculated with the following formula, as described in the documentation:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

If user u is unknown, then $b_u = p_u = 0$. Similarly, if movie i is unknown, then $b_i = q_i = 0$. To estimate all the unknown, the algorithm minimize the regularized squared error by performing stochastic gradient descent.

$$\sum_{r_{ui} \in R_{train}} (r_{ui} - \hat{r}_{ui})^2 + \lambda(b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2)$$

$$b_u \leftarrow b_u + \gamma(e_{ui} - \lambda b_u)$$
$$b_i \leftarrow b_i + \gamma(e_{ui} - \lambda b_i)$$
$$p_u \leftarrow p_u + \gamma(e_{ui} \cdot q_i - \lambda p_u)$$
$$q_i \leftarrow q_i + \gamma(e_{ui} \cdot p_u - \lambda q_i)$$

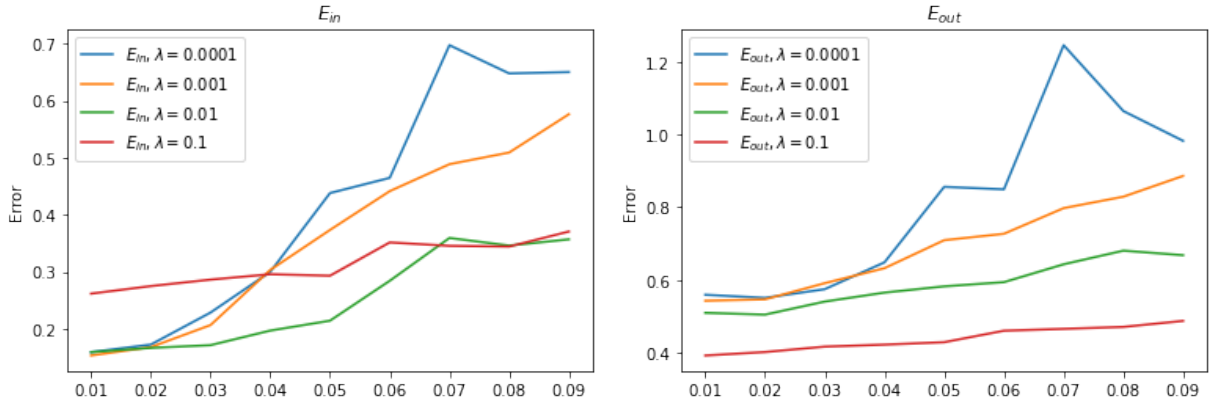
where $e_{ui} = r_{ui} - \hat{r}_{ui}$, γ is the learning rate, and λ is the regularization term. p_u (user factors) and q_i (movie factors) are the matrices we want (U and V). We used numpy linalg library's SVD method to break it down into $V = A\Sigma B$, for visualization part of the project.
When we compared our surprise SVD model with and without bias term, the model with bias term seemed to perform better on average with the test data (MSE error for model with bias was around 1.4, but MSE error for model without bias was around 1.5). So, we decided to include bias term in our model.

Choosing Parameters (Method 1 & 2)

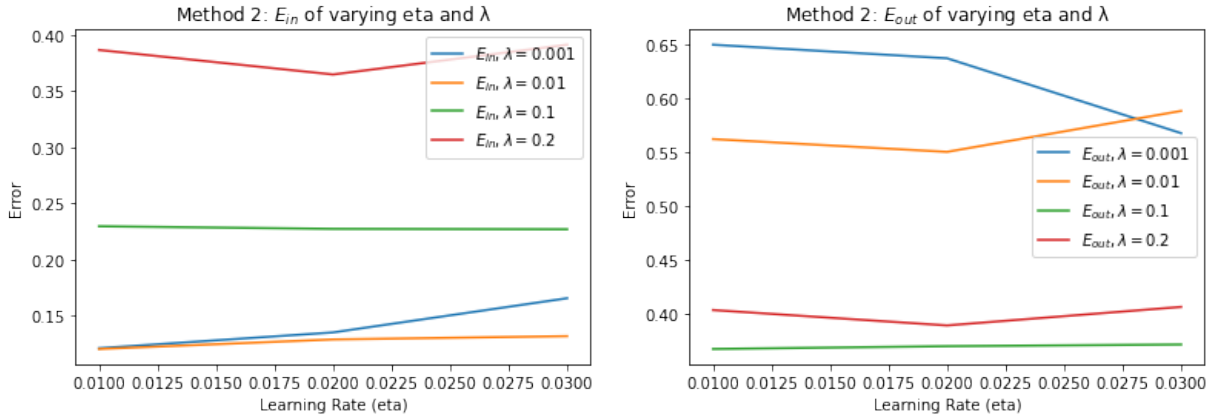
For method 1, we trained for different values of λ and η to find the parameters that minimized unregularized mean squared error on the test set. Since method 1 and method 2 are very similar, we thought that the optimal parameters for method 2 would be close to the optimal parameters for method 1. Thus, we tested method 2 with learning step sizes and regularizations parameters that were very close to the optimal parameters from method 1. By honing in on these predicted optima, we were able to limit our testing and avoid uselessly training models we knew would perform poorly. Ultimately, our intuition was confirmed as method 1 and method 2 proved to share the same optimal parameters for eta and reg. Ultimately, both method 1 and method 2 shared the same optimal parameters given our implementation of Grid Search: an eta of 0.01 and reg of 0.1. We found these optima by finding which parameters yielded the lowest testing error E_{out}. The performance of our models given different parameters for eta and reg can be visualized in the graphs below.

Loss curves

Loss curves for Method 1 Matrix Factorization where the x-axis represents values of η are below:



Loss curves for Method 2 Matrix Factorization are below:

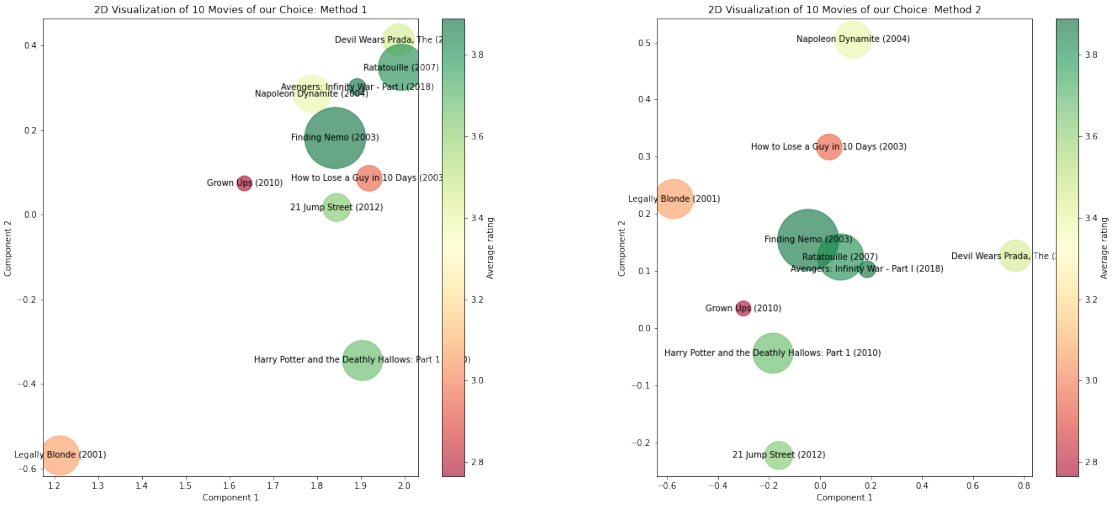


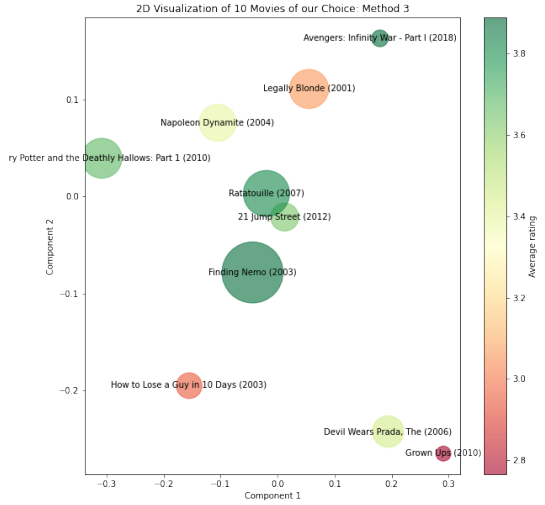
Model Comparison

We compared methods 1, 2, and 3, by finding the best models for all 3 methods and computing their mean squared error (MSE) given the same training set and test set. After training each model, we computed the prediction on each test point for each model, resulting in the MSE. Throughout many runs, Method 2's best model consistently outputted the lowest MSE. Method 1's best model outputted an MSE of around 0.76. Model 2's best model outputted an MSE of around 0.72. Model 3's best model outputted an MSE of around 1.47. We expected Method 2 to perfrom slightly better than Method 1 because they are very similar except for the bias vectors. The bias vectors let us model slightly more information, and thus yielded higher performance. The off-the-shelf implementation for Method 3 proved to perform worst on our given dataset by a significant margin compared to Method 1 and 2. Considering that mathematically, Method 3 is very similar to Method 2 with its own implementation of bias vectors, representation matrices, and regularization, then its poor performance may potentially reveal that the parameters it set weren't optimal or that its implemented training method wasn't well suited for this dataset.

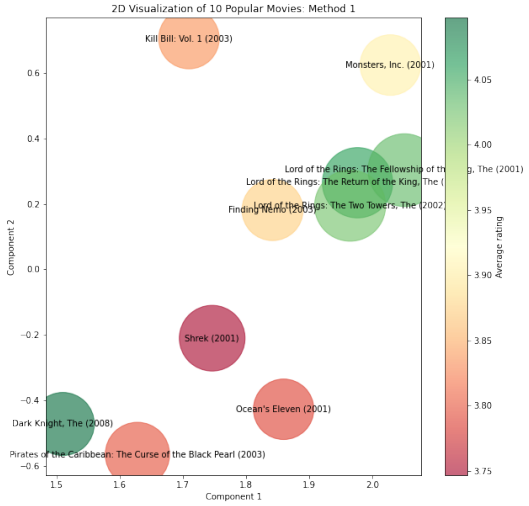
4 Matrix Factorization Visualizations

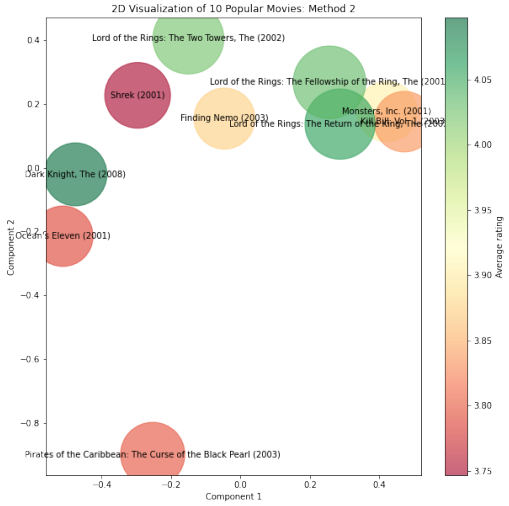
10 Movies of our Choice

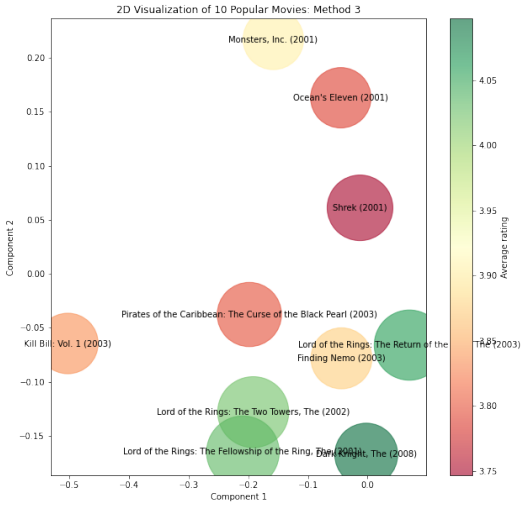




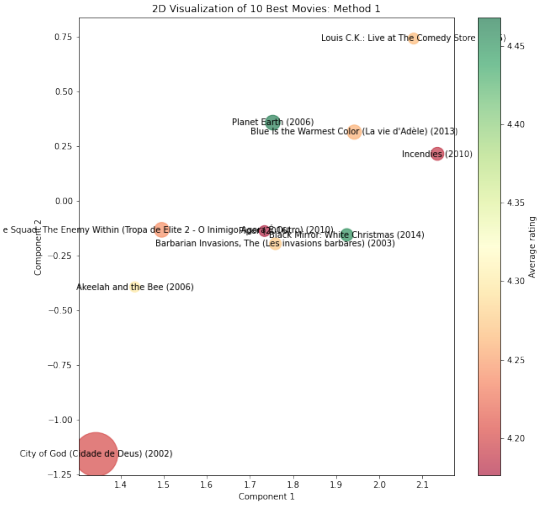
10 Most Popular Movies

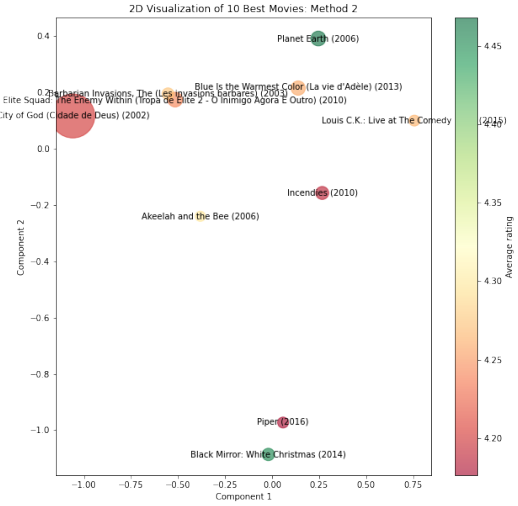


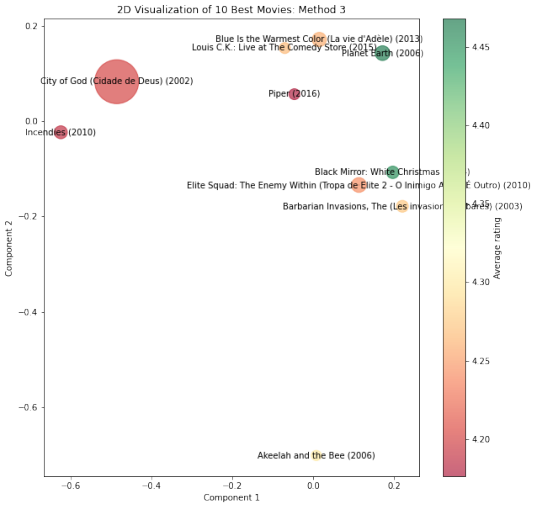




10 Best Movies

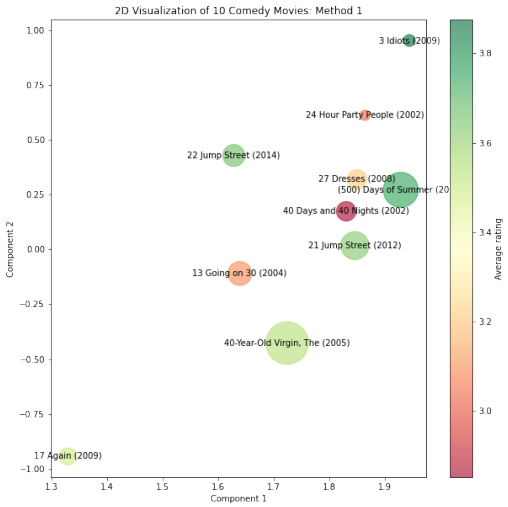


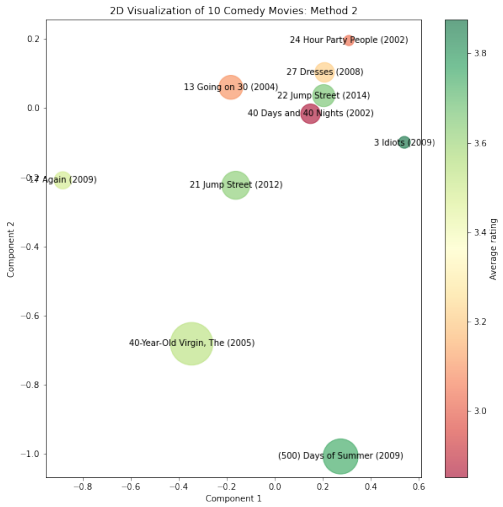


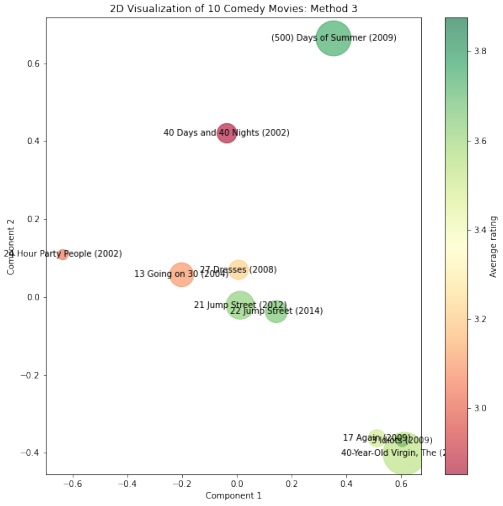


10 Movies from Three Genres Selected

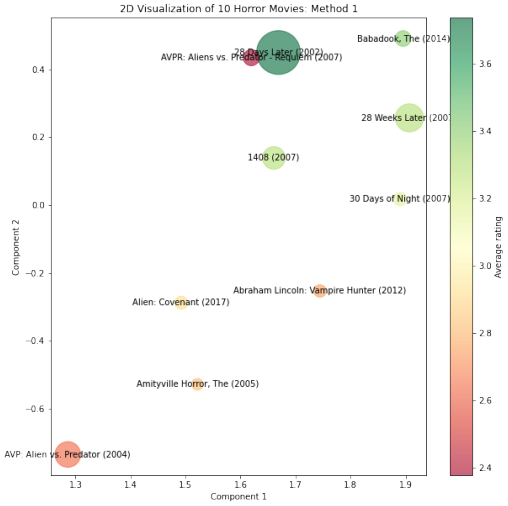
Comedy

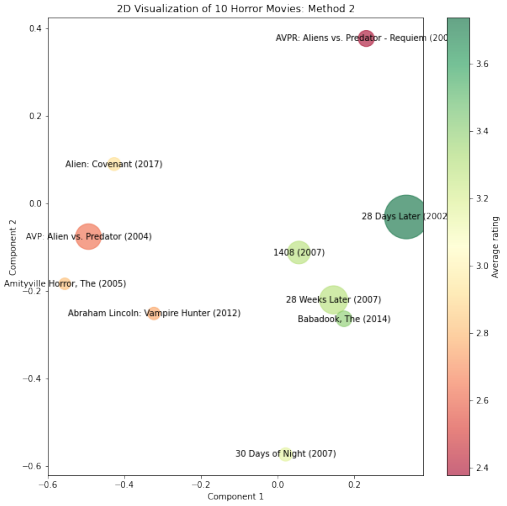


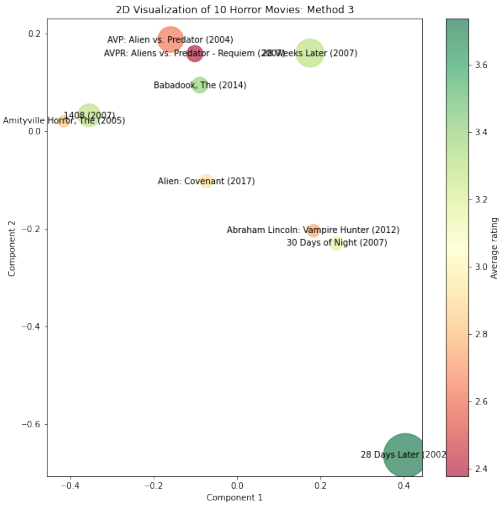




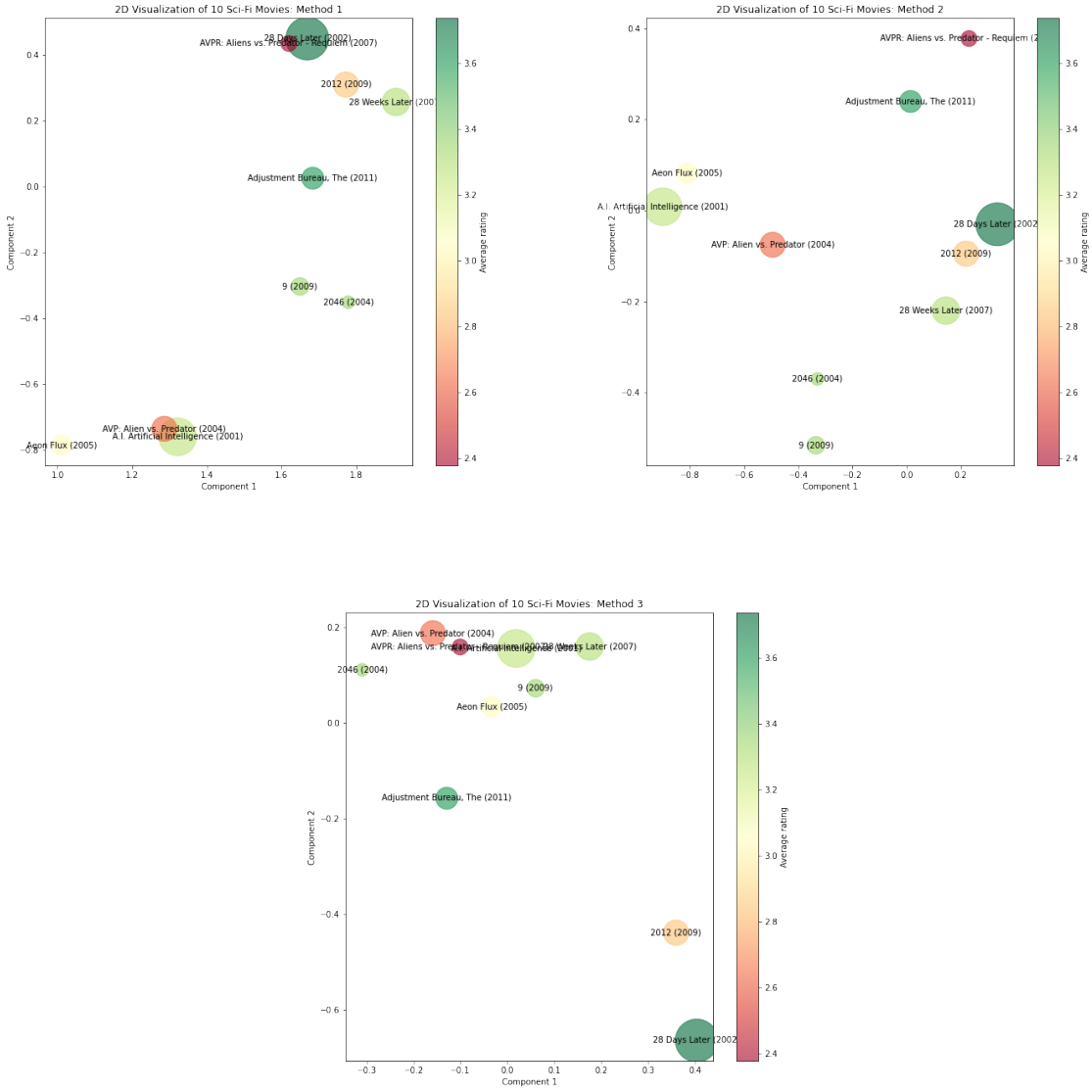
Horror


2D Visualization of 10 Horror Movies: Method 1
Babadook, The (2014)
28 Days Later (2002)
AVPR: Aliens vs. Predator - Requiem (2007)
28 Weeks Later (2007)
1408 (2007)
30 Days of Night (2007)
Abraham Lincoln: Vampire Hunter (2012)
Alien: Covenant (2017)
Amityville Horror, The (2005)
AVP: Alien vs. Predator (2004)
Component 1
Component 2
Average rating


2D Visualization of 10 Horror Movies: Method 2
AVPR: Aliens vs. Predator - Requiem (2007)
Alien: Covenant (2017)
28 Days Later (2002)
AVP: Alien vs. Predator (2004)
1408 (2007)
Amityville Horror, The (2005)
28 Weeks Later (2007)
Abraham Lincoln: Vampire Hunter (2012)
Babadook, The (2014)
30 Days of Night (2007)
Component 1
Component 2
Average rating


2D Visualization of 10 Horror Movies: Method 3
AVP: Alien vs. Predator (2004)
28 Weeks Later (2007)
Babadook, The (2014)
1408 (2007)
Amityville Horror, The (2005)
Alien: Covenant (2017)
Abraham Lincoln: Vampire Hunter (2012)
30 Days of Night (2007)
28 Days Later (2002)
Component 1
Component 2
Average rating

Sci-Fi



Observations and Comparisons

In general, we noticed that the movies that are similar (plot, genre, type) tend to be clustered together on the plot. For instance, the two Alien vs Predator movies (AVP and AVPR) are placed very closely to each other on both Horror and Sci-Fi plots (Method 3). And 21 Jump Street and 22 Jump Street are clustered together on the Comedy plot (Method 3). We see a similar trend in 10 Most Popular Movies (Method 1) with the three Lord of the Rings Movie. And even with the movies that are not in the same series, we see this kind of trend. For instance, the 10 popular movies visualization (method 1) accurately clumps together all 3 Lord of the Rings movies. The visualization also places both shrek and Monsters, Inc., two similar animated movies, in similar areas. Furthermore, shrek is placed closer to lord of the rings since both are fantasy movies. The two movies in this visualization classified as crime movies are the two highest movies on the y-axis. Pirates of the Caribbean, Dark Knight, Finding Nemo, and the Lord of the rings movies are all adventure movies which are placed at similar y-axis values. As the y-axis increases, the amount of action and crime in the movies increase as well. The three animated movies Finding Nemo, Shrek, and Monsters Inc. are all grouped at similar x-values. With the exception of Dark Knight, the 6 leftmost movies are more light-hearted and fun to watch in comparison to the 4 rightmost movies, creating a trend

of more serious themes as the x-values increase. The visualization of 10 comedy movies accurately clumps together the comedy and romance movies 17 Again, 3 Idiots, 27 dresses, The 40-Year-Old Virgin, and 500 Days of Summer. Furthermore, 21 and 22 Jump Street are clumped closely together right below the comedy and romance section. On the far upper left, 24 Hour Party People is classified as a musical drama, causing it to have less similarities to the other movies and to be grouped alone. The movie 13 going on 30 was also grouped alone since it is in the fantasy genre, causing it to have a distinct attribute. Furthermore, 40 days and 40 Nights is the only satirical film in this visualization and is by far the most erotic as well, causing it to be grouped separately. Therefore, we can see that as the y-axis increases, the amount of satire increases (21 and 22 Jump Street are directly above 40 Days and 40 Nights). Furthermore, the movies on the right-hand side of the graph do not include musical numbers besides the Bollywood movie the 3 Idiots which has the eighth most positive x-value.
We also noticed that movies of high ratings tend to form a cluster together in some of our plots. For instance, there is a cluster of movies with high ratings in the 10 Movies of our choice plot and in 10 Most Popular Movies plot. Since we are only plotting the two components of our factors matrix, this could possibly imply that there are few deciding factors that really determine the popularity of the movies (we disregard 10 movies of our choice plot in this discussion because that choice was random).

Result vs Expectations
The result was as expected for most of our plots. We see that similar movies are placed close to each other on the plot.

Most Popular Movies Compared to Best Movies
We noticed that on average, 10 Best Movies have higher ratings than 10 Most Popular Movies (which is expected). This also shows that popularity doesn't always guarantee higher ratings - when we compare the list of movies in the two plots, there is no overlap. Also, most of the points on the Best Movies plots are small circles, meaning that it was less popular (our circle sizes represent the number of ratings submitted), while 10 Popular Movies plot has big circles, representing large number of ratings submitted.
Both plots also had a very narrow range for the ratings compared to the other 4 plots. Their average rating scale ranged from 3.75 to 4.05 (most popular) and 4.20 to 4.45 (best movies). Most Popular Movies has more definitive clusters formed on the plot (a cluster of high rating movies, a cluster of Lord of the Rings movies), while Best Movies plot seems to have less of that pattern (with Blue Is the Warmest Color and Planet Earth placed next to each other in method 3 plot).

The Three Genres We Chose
We chose Comedy, Horror, and Sci-Fi for our visualization. It is surprising that Horror and Sci-Fi produced very similar plots. One of the reasons could be that there are a lot of overlapping movies between these two. For instance, they both have AVP, AVPR, 28 Weeks Later, and 28 Days Later. So, a lot of these movies seem to be placed in a very similar location on the plot relative to each other in both plots. Comedy has totally different set of movies so we don't observe the same pattern in that.