# PCA-SVD-Clusters

Shri Atluri

November 25, 2024

## 1 Introduction

This project involves the analysis and transformation of user-movie ratings data from the MovieLens dataset. The objective is to explore and extract insights using data science techniques such as clustering, Principal Component Analysis (PCA), and Singular Value Decomposition (SVD). By following the structured methodology outlined below, the project seeks to provide a comprehensive understanding of the dataset while addressing the specific requirements in the rubric.

### 1.1 Objective

The goal of this project is to transform, analyze, and visualize a user-movie rating dataset to uncover patterns and relationships. The key steps involve:

1. **Data Preparation:** Transform the dataset into a user-movie ratings matrix, filling missing values with zero. Identify the top-rated movies and most active users to understand initial trends.

2. **Clustering:** Apply k-means clustering to group users based on their movie preferences. Analyze the resulting clusters to determine patterns and identify the most relevant $k$ value using the elbow method.

3. **Principal Component Analysis (PCA):** Reduce the dimensionality of the dataset to identify the intrinsic structure of the data. Visualize the reduced data and assess how much variance is retained using varying numbers of components.

4. **Singular Value Decomposition (SVD):** Decompose the data to further understand the latent features of the user-movie matrix. Compare explained variance ratios with the clustering inertia to validate the analysis.

### 1.2 Overview of the Steps

- **Data Transformation:** The dataset is converted into a matrix with 610 users and 9,724 movies. Missing ratings are replaced with zeros, ensuring a complete representation.
- **Clustering:** K-means clustering is performed with various values of $k$ ($k = 2, 4, 8, \ldots, 128$) to identify meaningful groupings of users based on their movie preferences.
- **PCA:** The dimensionality of the dataset is reduced to two components for visualization. Variance analysis is conducted to determine the intrinsic dimensionality.
- **SVD:** The user-movie matrix is decomposed, and the top singular values are analyzed to understand the relationship between clustering and dimensionality reduction.
- **Conclusion:** A summary of findings from all analyses is presented, highlighting key insights and patterns discovered during the project.

## 2 Data Transformation

The transformation process involved preparing the MovieLens dataset by creating a user-movie ratings matrix and analyzing the data to identify key trends. This section is divided into three parts: matrix creation, top-rated movies, and most active users.

### 2.1 Matrix Creation

The dataset used for this project comprises two files: `ratings.csv` and `movies.csv`. The `ratings.csv` file contains user ratings for movies, with columns for `userId`, `movieId`, `rating`, and `timestamp`. The `movies.csv` file contains movie metadata, including `movieId`, `title`, and `genres`.

A user-movie ratings matrix was created by pivoting the `ratings.csv` dataset, with rows representing users (`userId`), columns representing movies (`movieId`), and the cell values containing the corresponding movie ratings. Missing values in the matrix were filled with zeros to ensure completeness, resulting in 610 rows and 9,724 columns.

### 2.2 Top-Rated Movies

The analysis identified the top 3 movies based on the number of users who rated them. Using the `movieId` from the `ratings.csv` file, the number of ratings for each movie was computed and merged with the `movies.csv` file to retrieve the movie titles. The top 3 movies are:

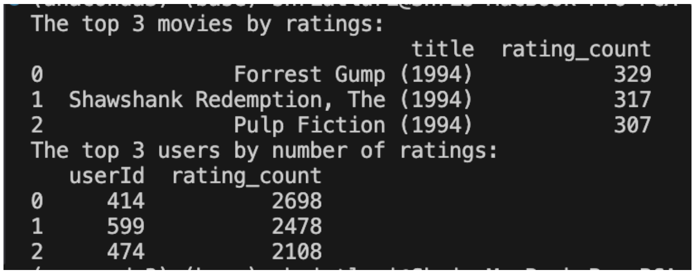
- **Forrest Gump (1994)**: 329 ratings

- **The Shawshank Redemption (1994)**: 317 ratings
- **Pulp Fiction (1994)**: 307 ratings

## 2.3 Most Active Users

The analysis also identified the top 3 users who rated the greatest number of movies. This was done by counting the number of ratings provided by each user in the `ratings.csv` file. The top 3 users and their respective counts are:

- **User ID 414**: 2,698 ratings
- **User ID 599**: 2,478 ratings
- **User ID 474**: 2,108 ratings



```
The top 3 movies by ratings:
                             title  rating_count
0               Forrest Gump (1994)          329
1  Shawshank Redemption, The (1994)          317
2               Pulp Fiction (1994)          307
The top 3 users by number of ratings:
   userId  rating_count
0     414          2698
1     599          2478
2     474          2108
```

Figure 1: Top three rated movies and most active users

## 2.4 Code for Transformation

The following Python code was used to transform the data, identify the top-rated movies, and determine the most active users in the dataset:

```
import pandas as pd

file_path = ’ratings.csv’
ratings_data = pd.read_csv(file_path)

movies_file_path = ’movies.csv’
movies_data = pd.read_csv(movies_file_path)

# creating the matrix
user_movie_matrix = ratings_data.pivot(index = ’userId’, columns = ’movieId’,
values = ’rating’)
user_movie_matrix = user_movie_matrix.fillna(0)

# finding the top movies as well as their names, count the values
```

```
top_movies = ratings_data['movieId'].value_counts().head(3).reset_index()
top_movies.columns = ['movieId', 'rating_count']

# merge to get titles
top_movies_titles = top_movies.merge(movies_data, on = 'movieId', how = 'left')

print('The top 3 movies by ratings:')
print(top_movies_titles[['title', 'rating_count']])

# finding the top 3 users using same methodology
top_users = ratings_data['userId'].value_counts().head(3).reset_index()
top_users.columns = ['userId', 'rating_count']

print("The top 3 users by number of ratings:")
print(top_users)
```

This code utilizes the `pandas` library to load and manipulate the dataset. It creates a user-movie matrix, counts the number of ratings for each movie and user using the valuecounts() method, and merges the results with the `movies.csv` file to retrieve movie titles.

# 3 Clustering

The clustering step involved grouping users based on their movie rating patterns using k-means clustering. This section describes the analysis for determining the optimal number of clusters, the clustering results, and insights into the most highly rated movies in each cluster.

## 3.1 Inertia Analysis

To determine the optimal number of clusters ($k$), k-means clustering was applied to the user-movie rating matrix for values of $k = [2, 4, 8, 16, 32, 64, 128]$. The inertia, which measures the sum of squared distances between points and their cluster centroids, was calculated for each $k$.

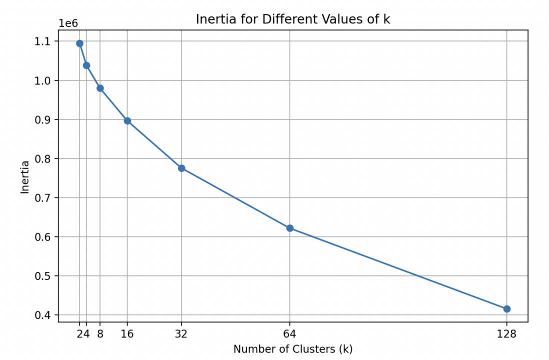


Figure 2: Number of clusters and Inertia

The results were plotted to visualize the ”elbow point,” where the rate of inertia decreases. The inertia plot suggests that $k = 32$ is an appropriate choice, balancing the compactness of clusters and the model’s simplicity.

## 3.2 Optimal Number of Clusters

Based on the elbow method, $k = 32$ was selected as the optimal number of clusters. This value provides a meaningful balance between capturing user preferences and avoiding excessive fragmentation of the data.

## 3.3 Cluster Analysis

With $k = 32$, the dataset was clustered, and the top three movies with the highest average ratings within each cluster were identified. These movies represent the preferences of users in each group. Below are some examples of clusters and their corresponding top-rated movies:

- **Cluster 0**:
  - *Shawshank Redemption, The (1994)*
  - *Matrix, The (1999)*
  - *Fight Club (1999)*
- **Cluster 1**:
  - *Sense and Sensibility (1995)*
  - *Dead Man Walking (1995)*
  - *Shawshank Redemption, The (1994)*
- **Cluster 2**:
  - *Toy Story (1995)*

  - *Casino (1995)*
  - *Star Wars: Episode IV - A New Hope (1977)*

- **Cluster 31**:
  - *Pulp Fiction (1994)*
  - *Dr. Strangelove or: How I Learned to Stop Worr...*
  - *Vertigo (1958)*

These results indicate that the clustering effectively captures user preferences, as the top-rated movies within each cluster often belong to distinct genres or themes. For example, Cluster 0 is dominated by highly rated drama and action movies, while Cluster 2 includes a mix of animated and science fiction classics.

## 3.4 Conclusion

The k-means clustering process identified meaningful groupings of users based on their movie preferences. The selection of $k = 32$ as the optimal number of clusters provided well-defined user groups, and the top-rated movies within each cluster reflect distinct audience preferences.

# 4 Principal Component Analysis

Principal Component Analysis (PCA) was applied to reduce the dimensionality of the user-movie ratings dataset. This section discusses the data transformation, visualization, and intrinsic dimensionality of the dataset.

## 4.1 Data Transformation

The user-movie matrix was transposed such that rows represent movies and columns represent users. The data was then mean-centered by subtracting the mean rating for each user. This preprocessed data served as input for PCA.

## 4.2 Dimensionality Reduction (PCA)

PCA was applied to the mean-centered matrix with $k = 2$ components to reduce the dataset to a two-dimensional space. The first two principal components explain 17.62% and 4.19% of the total variance, respectively.

## 4.3 Visualization

The results of PCA were plotted in a 2D space, with each point representing a movie. Points were colored based on the movie's primary genre, extracted from the dataset. The visualization is shown in the figure below.

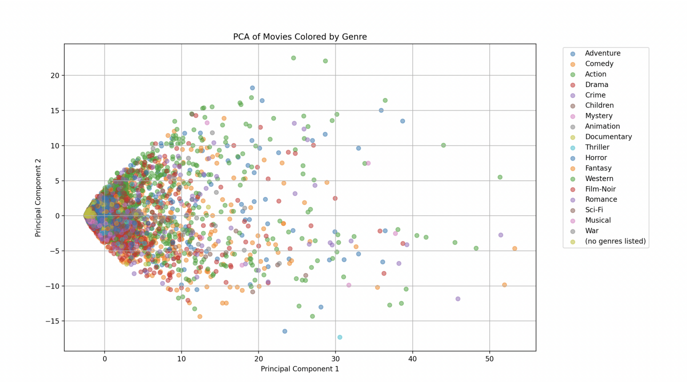


Figure 3: PCA of the movies colored by genre

### 4.4 Intrinsic Dimensionality

The cumulative explained variance was analyzed to determine the intrinsic dimensionality of the dataset:

- Only **1 component** is required to explain 80% of the variance.
- Similarly, **1 component** is sufficient to explain 40% of the variance.

This suggests that the dataset is highly compressible, and a single principal component captures most of the variance. However, reducing the dataset to two components ($k = 2$) provides a more interpretable visualization while retaining approximately 21.81% of the total variance.

### 4.5 Discussion

The PCA visualization reveals clusters of movies based on their genres. For example, movies of similar genres (e.g., action, animation) tend to cluster together, indicating that genre significantly influences user ratings. However, some overlap between genres may reflect users' diverse tastes or the multi-genre nature of certain movies. While $k = 2$ provides a helpful visualization, the intrinsic dimensionality analysis indicates that most of the information in the dataset is captured by the first principal component.

## 5 Singular Value Decomposition (SVD)

### 5.1 Singular Values and Variance Distribution

This step applied Singular Value Decomposition (SVD) to the user-movie rating matrix with $k = 128$. SVD decomposes the original matrix into three matrices: $U$, $\Sigma$, and $V^T$, where $\Sigma$ contains the singular values in descending order. These singular values indicate the variance explained by each component.

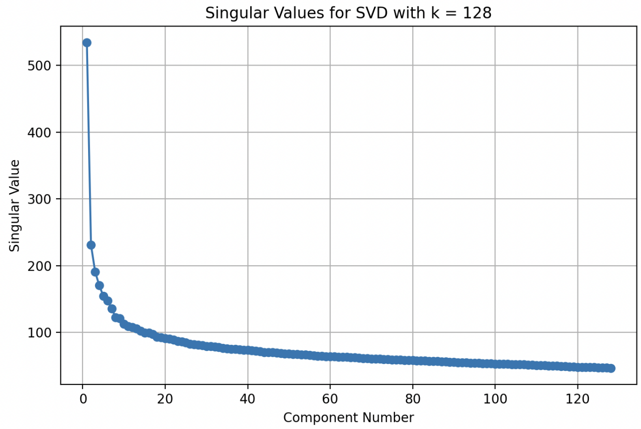


Figure 4: Singular Values for SVD with $k = 128$.

The plot above shows the singular values for the first 128 components. The steep decline in singular values suggests that most of the variance is captured by the first few components, while subsequent components contribute progressively less variance. This aligns with the expectation that the user-movie ratings dataset is inherently low-dimensional, meaning that a smaller number of components can effectively summarize the data.

### 5.2 Interpretation of Results

The most considerable singular value corresponds to the first variance pattern, which captures the most significant trend in the data (e.g., broad user preferences or widely popular movies). Subsequent singular values represent orthogonal variance patterns, capturing smaller and more specific trends. The rapid decay of singular values highlights the diminishing returns of retaining additional components beyond the initial ones.

### 5.3 Explained Variance for Different $k$

The sum of the explained variance ratio was calculated for various values of $k = [2, 4, 8, 16, 32, 64, 128]$ to evaluate how much variance is retained by the corresponding number of components. The results are summarized in Table 1.

### 5.4 Discussion and Comparison with Clustering Inertia

The results in the table below indicates that as $k$ increases, the proportion of variance explained by the SVD components grows. For $k = 128$, approximately 74.61% of the variance is captured. However, smaller values of $k$ retain less variance:

| $k$ | **Sum of Explained Variance Ratio** |
|---|---|
| 2 | 0.1728 |
| 4 | 0.2219 |
| 8 | 0.2870 |
| 16 | 0.3635 |
| 32 | 0.4608 |
| 64 | 0.5876 |
| 128 | 0.7461 |

Table 1: Sum of Explained Variance Ratios for Different Values of $k$.

- At $k = 32$, 46.08% of the variance is explained, which aligns with the choice of $k = 32$ for clustering in Question 2. This suggests that $k = 32$ balances information retention with computational efficiency.
- At $k = 2$, only 17.28% of the variance is retained, meaning most of the information in the dataset is lost.

Comparing these results to the inertia values from clustering, we observe that the explained variance provides a finer measure of information retention. While inertia directly measures compactness within clusters, the explained variance evaluates the data's structure and highlights diminishing returns when increasing $k$.

Overall, the results validate the choice of $k = 32$ as a meaningful balance between simplicity and accuracy. In contrast, higher values (e.g., $k = 128$) retain significantly more variance. Still, they may lead to overfitting, especially with the vast increase of K to the return of variance you get in exchange.

## 5.5 SVD with $k = 2$: Reduced Dimensionality

SVD was applied with $k = 2$ components to explore the reduced representation of the user-movie rating matrix. The resulting two components represent the data's most significant patterns of variance. Each user is plotted in this 2D space, with their points colored according to their cluster memberships derived from Question 2. The results are shown in the plot below.

## 5.6 Discussion and Comparison with Clustering and PCA

The visualization in the figure above reveals how users are distributed in the 2D space defined by the first two singular vectors. Each point corresponds to a user, and the color indicates their cluster assignment from Question 2. Several observations can be made:

- **Cluster Separation:** Users within the same cluster tend to group together in the SVD space, suggesting that the SVD components align well with the clustering structure from Question 2. However, some clusters

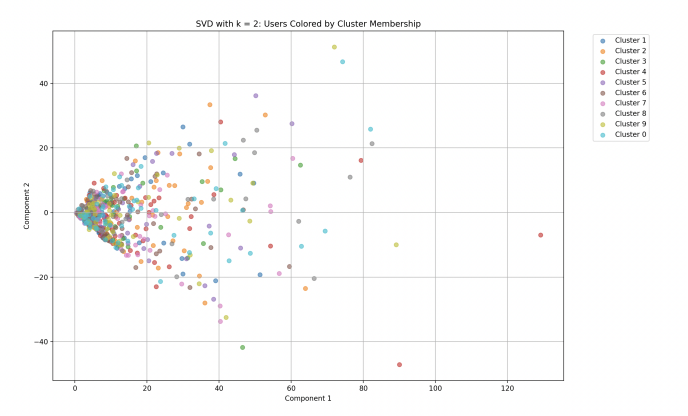


Figure 5: SVD with $k = 2$: Users Colored by Cluster Membership.

show significant overlap, indicating that users in different clusters may still share similarities in their movie ratings.

- **Patterns Compared to PCA:** Unlike the PCA visualization, which focused on variance among movies, this plot emphasizes variance among users. The SVD components capture the dominant user behavior patterns, leading to distinct groupings that correspond to shared rating tendencies.
- **Outliers:** A few users are located far from the central cluster. These outliers may represent users with unique or highly specific rating behaviors that deviate from the general patterns.
- **Alignment with Clustering:** The alignment between the clusters from Question 2 and the SVD visualization suggests that the reduced dimensions effectively capture the key differences between user groups. This supports the validity of the clustering results and demonstrates the compatibility of SVD for dimensionality reduction and clustering.

### 5.7 Impact of $k = 2$

Reducing the dimensionality to $k = 2$ facilitates visualization and highlights user behavior patterns. However, this reduction only captures the two most dominant variance patterns, which may not fully explain the details of user preferences. For a more detailed analysis, higher values of $k$ (e.g., $k = 32$ or $k = 64$) could be used to retain additional variance and have more refined clustering results.