Case Study in ULEOR

Unsupervised Learning and Evolutionary Optimisation Using R

Matheus Cruz, Kasi Venkata Sai Kiran Vattem, Katherin Jose, Finn Pieprzyk, Ravi Pandey, and Levi Kletetzka

University of Paderborn, Germany
{mcruz, vkvsk, kj2, pieprzyz. rpandey, kletetzk}@uni-paderborn.de

1 Data Analysis (Matheus Cruz, Kasi Vattem)

In this task, we performed basic data analysis on Netflix data using R and the tidyverse package. The primary focus was on understanding the data and performing specific tasks related to filtering, ranking, and summarizing the data.

1.1 Action Shows with IMDb Rating > 8.9

The dataset `titles` was filtered to include only action shows (`type = SHOW`) with action as one of the genres with an IMDb rating higher than 8.9. The results were sorted in descending order by the IMDb score. This provided a clear view of the top-rated action shows.

1.2 Top 2 Movies from Japan and France

Movies produced solely in Japan or France were extracted by filtering the `production_countries` field excluding co-productions. The data was then grouped by country and sorted by `imdb_score`, `imdb_votes`, and `tmdb_score`. The top two movies from each country were selected.

1.3 Top Sci-Fi Actors

To identify the most prominent actors in sci-fi movies we began by filtering the dataset to include only movies within the sci-fi genre. We then focused on two key metrics to rank the actors: the average IMDb score of the movies they appeared in and the total number of appearances. To ensure fairness in the ranking, both metrics were normalized using the z-score approach, which helps to avoid the influenced by extreme values.After normalization, a combined rank was computed, where the number of appearances was weighted more heavily (60%) compared to the average IMDb score (40%)

1.4 Top 3 best and worst actors

For this section we only needed to calculate the average IMDb score of every actor, sort the list in ascending order for the worst 3, and descending order for the top 3.

2 Outlier Detection and Statistical Testing (Finn Pieprzyk)

To allow a more reliable analysis and reduce bias when treating missing values, the two genres `fantasy` and `crime` with sufficient data and a low number of missing values in the IMDb and TMDB scores in relation to the total number of films and shows in these genres were selected. The statistical data distribution and outliers are analyzed separately below for the IMDb and TMDB ratings, with the results being compared at the end.

2.1 Analysis of IMDb ratings

To understand the distribution of the IMDb ratings for the two selected genres the distribution is visualized first. The figure 1 shows that the distribution deviates from normality. The Shapiro-Wilk test also suggests, with a p-value smaller than the significance level of 0.05, that the IMDb ratings do not follow a normal distribution. Furthermore the data is slightly left skewed with a left tail. This can also be seen in the histogram 2.

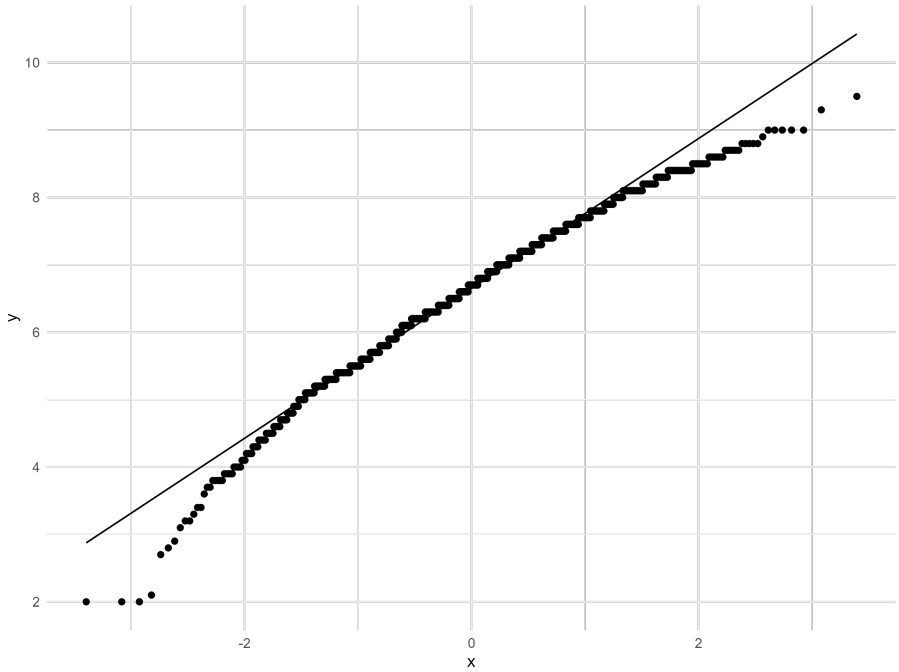


Figure 1: QQ-plot of IMDb ratings for the selected genres

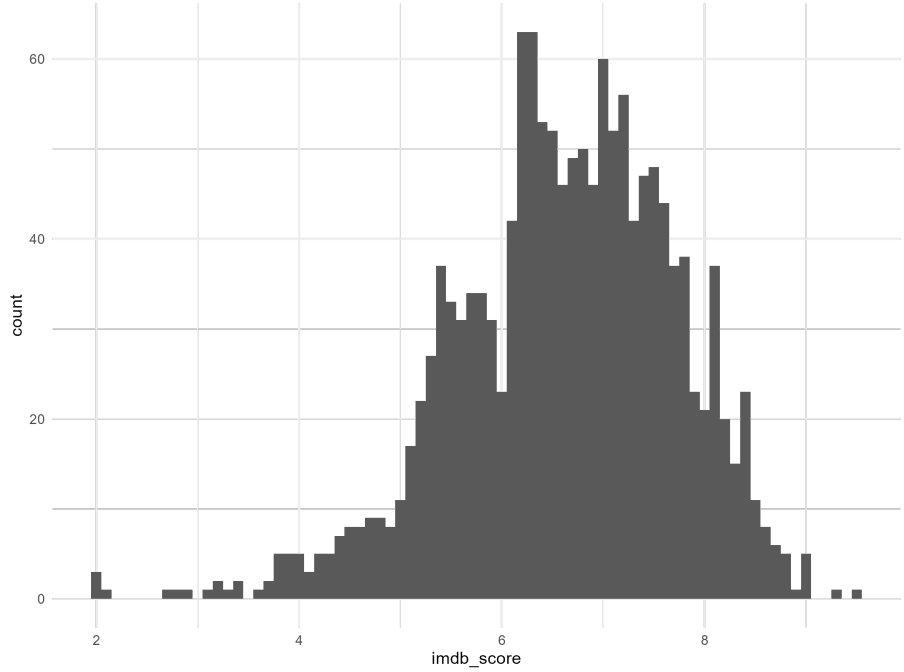


Figure 2: Histogram of IMDb ratings for the selected genres

As shown in the figure 3 these outliers on the left represent movies and shows with a very low rating smaller than 3.7. In general, movies and shows tend to be rated higher on a scale of 1 to 10, with a median of 6.7 and 50% of ratings between 5.9 and 7.4.

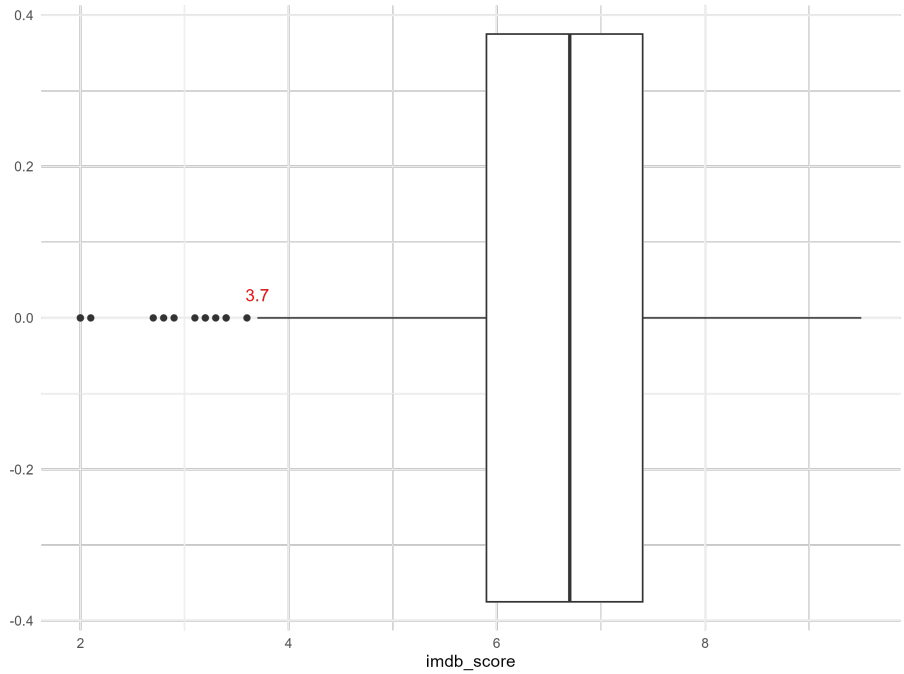


Figure 3: Boxplot of IMDb ratings for the selected genres

To find out the reasons for these outliers, the correlation of the IMDb scores with various variables can be examined. When examining the outliers, it can be seen that they tend to have a lower number of total votes. The figure 4 and a Pearson correlation coefficient of 0.26 show a moderately positive correlation between the number of votes and the IMDb score thus suggesting that movies and shows with more votes are rated higher. The reason for this can be that films and shows with a higher rating are likely to be watched more often and therefore receive more ratings. Moreover films and series with a higher number of votes from a wider target audience are more likely to have a balanced distribution of scores, while films and series with fewer votes can be more easily influenced by extreme scores. IMDb tries to correct that by using a weighted average instead of the arithmetic mean. In addition, a different weighting calculation is used when unusual voting activity is detected [IMD24]. All in all, this could be the reason why there are not many movies and shows with extremely low or high scores.

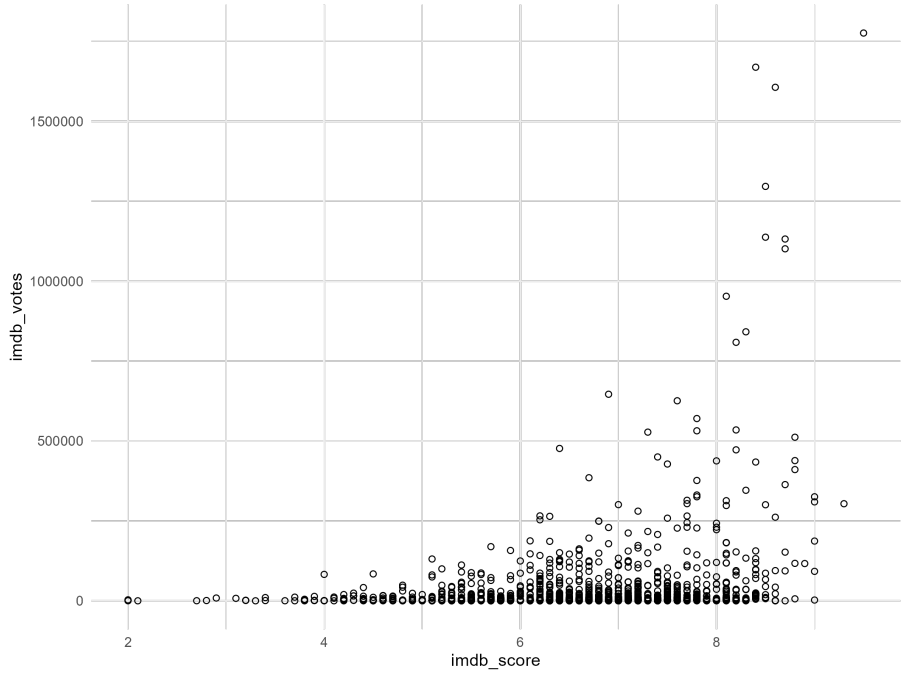


Figure 4: Scatterplot of IMDb ratings and corresponding number of votes

In addition to the number of votes, there is also a positive correlation between the IMDb rating and the year of release, suggesting that older films and series are rated higher, which may be due to their cultural influence and the fact that they are rated in a different context than modern films and series. The age rating correlates positively with the IMDb rating as well. Furthermore, shows are on average rated better than movies which can be attributed to the deeper connection with the viewer caused by the longer runtime. It is important to note that only the correlation was analyzed, which only indicates the relationship between these variables, but does not prove that they are causally linked.

2.2 Analysis of TMDB ratings

As illustrated in the figures 5 and 6 the distribution of TMDB ratings seems to be close to a normal distribution but is not due to heavy outliers on the left and right which is supported by the Shapiro Wilk test.

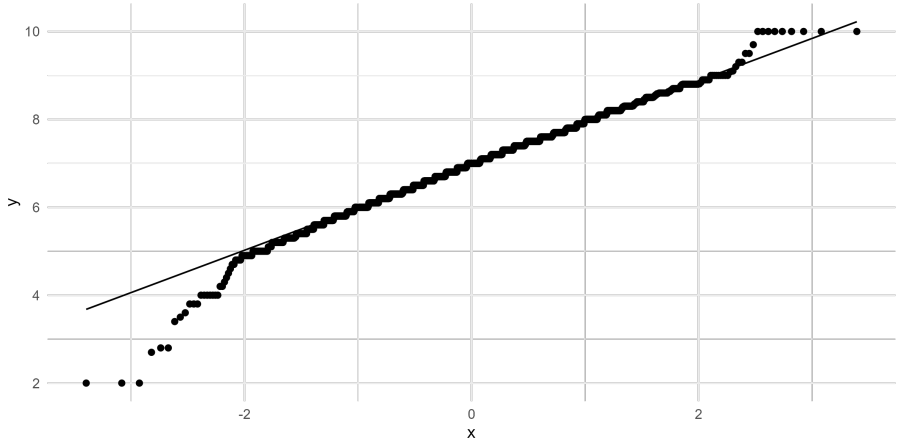


Figure 5: QQ-plot of TMDB ratings for the selected genres

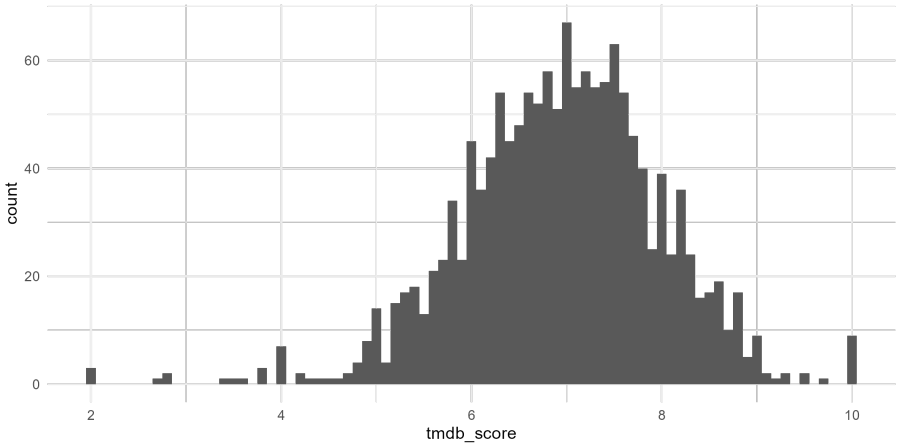


Figure 6: Histogram of TMDB ratings for the selected genres

The Boxplot 7 shows that these outliers are movies and shows with a rating lower than 4.4 or higher than 9.5. It can be seen that the median of the ratings is 7 while 50% is rated between 6.3 and 7.6.

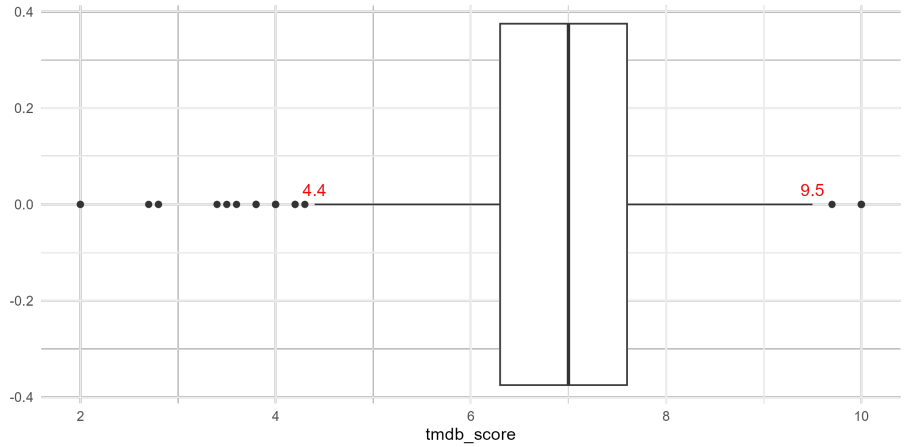


Figure 7: Boxplot of IMDb ratings for the selected genres

The TMDB rating correlates with the various attributes in a similar way to the IMDb rating.

2.3 Comparison of IMDb and TMDB ratings

Since one cannot assume that both ratings follow a normal distribution the non parametric Wilcoxon-Mann-Whitney test is used to test whether the ratings follow a similiar distribution. The test is based on the rank of the data and therefore ignores the different format of the ratings, potential outliers and also tries to correct for ties in the data. The comparison of the distributions of the IMDb and TMDB ratings shows that there is no statistical evidence that they follow the same distribution. The difference in the distribution can also be observed in the different histograms 2 and 6.

There may be several reasons for the different distribution of ratings. One reason could be the different calculations used for the rating. While IMDb uses a weighted average, TMDb uses the arithmetic mean, which could lead to more extreme ratings. In addition, the platforms are used by two different user groups, with TMDB users possibly being more international, as the variance in ratings by country of production is lower. These user groups also differ in size, reflecting the varying popularity of the sites. IMDb, with over 250 million users, is the more popular site compared to TMDB with 35 million users, resulting in different distribution of ratings [Nee25; TMD24].

2.4 Outlier Analysis of Actors Based on Appearances in Netflix Titles(Katherin Jose)

Analysis Summary

- **Approach Used**: Z-score method was applied to identify actors whose total appearances in Netflix titles significantly deviated from the mean. Actors with a Z-score greater than 3 (or less than -3) were classified as outliers.
- **Dataset**: The dataset contains information about actors and the number of titles they appear in.

Findings

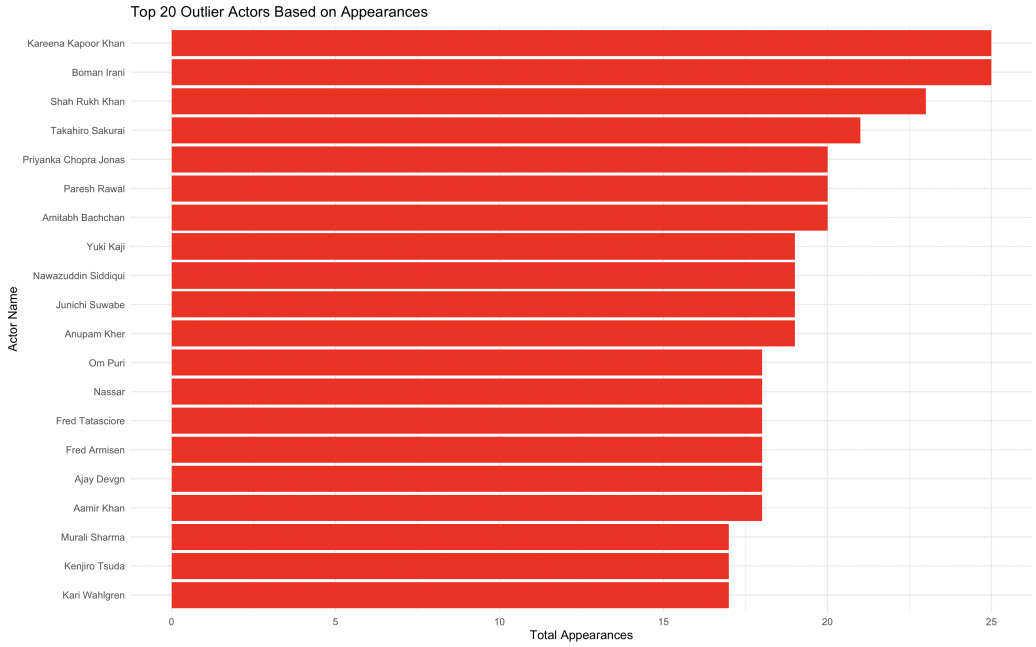
1. **Number of Outliers**:
   A total of 1,284 actors were identified as outliers. These actors appeared in a significantly higher number of titles compared to the average actor in the dataset.
2. **Prominent Outliers**:
   Some of the top actors with the highest number of appearances include:
   - **Kareena Kapoor Khan**: 25 appearances
   - **Boman Irani**: 25 appearances
   - **Shah Rukh Khan**: 23 appearances

- **Takahiro Sakurai**: 21 appearances
- **Priyanka Chopra Jonas**: 19 appearances

3. **Patterns Observed**:
   Many of the identified outliers are prominent figures in the Indian film industry (e.g., Bollywood) and anime voice acting. This suggests regional content or popular genres (e.g., anime) drive higher appearances for these actors.

Visualization

The bar graph below highlights the top 20 outlier actors based on their total appearances in Netflix titles. The analysis reveals that certain actors play a prominent role in Netflix's content offerings, particularly in specific genres or regional productions. These actors appear in a disproportionately high number of titles, which may indicate trends in content popularity or production focus.



3 Clustering (Levi Kletetzka and Ravi Pandey)

The clustering will be performed with the three different algorithms which are known from the lecture: **Hierarchical Clustering Algorithm (HCA)**, **K-Means Clustering** and the **DBScan**. As a basic exploration we choose the proposed features of the *release_year* of the movie from 'titles.csv' and use the second file 'credits.csv' to create a aggregated feature called *'total_count'* which corresponds to the cardinality of the movies cast.

3.1 Hierarchical Clustering Algorithm

Setup of Experiment For the Hierarchical Cluster Analysis (HCA), , the primary decision involves determining the number of clusters based on the dendrogram generated. Unlike k-means clustering HCA does not require an initialisation procedure. Instead, clusters are built step by step, either from individual data points upwards known as agglomerative approach or from a single cluster downwards known as divisive method.

We used hierarchical clustering to organize shows based on scaled IMDb ratings and runtime. Data were first normalized to ensure that IMDb scores (ranging from 1 to 10) and runtime (measured in minutes) contributed equally to distance calculations. To decrease variance within each cluster, Ward's method was utilized.

The branches of the dendrogram represents the distance (or dissimilarity) at which clusters are merged. A higher merge point on this axis indicates that merging clusters had a larger dissimilarity. For instance, major branches merging higher up the scale suggest that those clusters are quite distinct from each other, which is crucial when considering the natural groupings within the data.

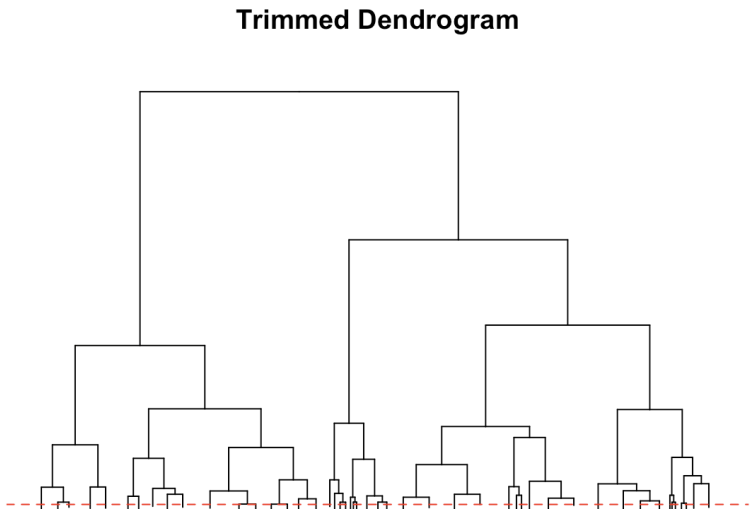


Figure 8: Dendrogram for HCA

Based on the Dendrogram as represented in figure 8, we came to the conclusion to use 3 groups to create clear distinctions.

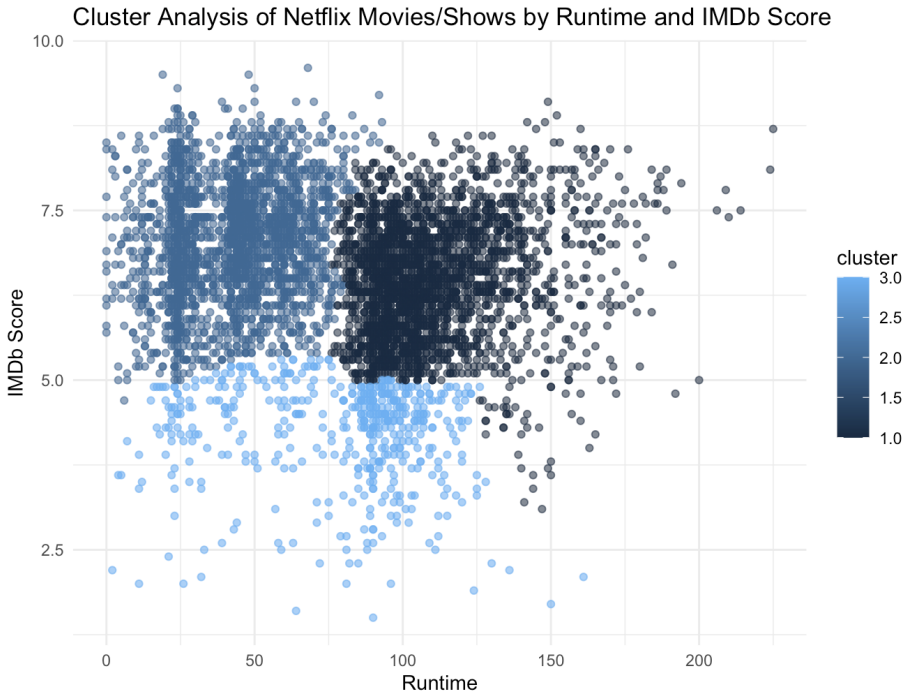


Figure 9: Cluster Results for HCA

Results In the above plot 9 we can see three distinct clusters, each corresponding to varying degrees of runtime and IMDb scores. Movie titles with the highest ratings do not necessarily have longer runtime. To a Netflix used one could suggest the movies in medium blue and dark blue clusters, as they have highest rating with varying Runtime.

3.2 K-Means Clustering

Setup of Experiment For the K-means clustering there are mainly two choices to make before running the algorithm on the data: Choosing an initialization method for the algorithm and determining the number of clusters. In order to choose the number of cluster there are two possibilities: Plotting the total within groups sum of squares against the number of clusters and identifying the visual 'elbow-point' or using the gap statistic. Again both methodologies will be checked and compared. In terms of initialization method we will use all options presented in the lecture: Lloyd, Forgy and kmeans++. The LLoyd Initialization is self-coded while Forgy and kmeans++ were provided by the kmeans-algorithm of the clustlearn package of R

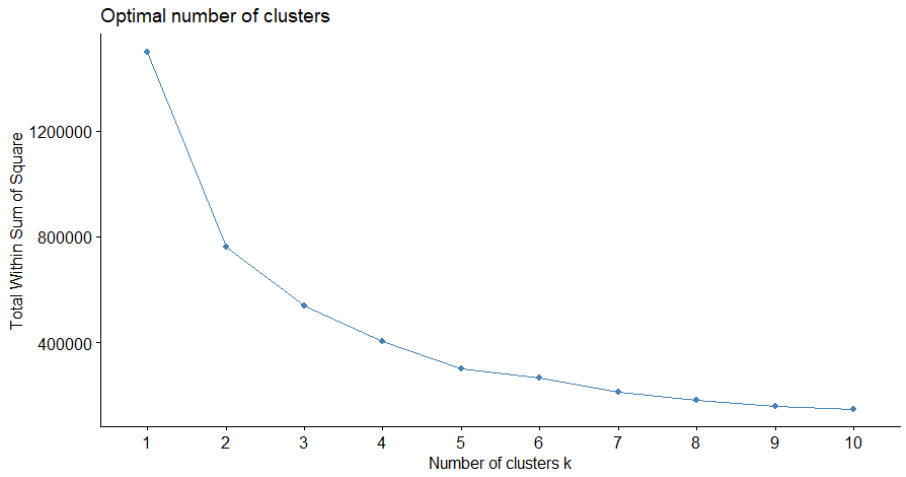


Figure 10: Elbow-Graph for K-Means

One can identify a big drop in the total within sum of squares after k = 2. Based on this graph the optimal number of clusters should be two. Now we will compute the gap statistics as a second metric and compare the two results:

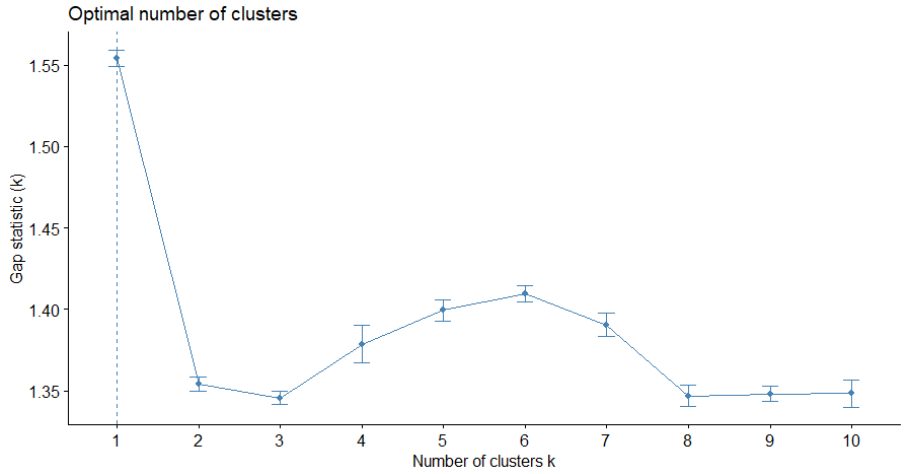


Figure 11: Gap statistic graph

This gap statistic graph does not seem to make sense. It could be the case that there was a implementation error in creating it. In the following the number of clusters remains 2 as indicated by the graph of the Total Within Sum of Squares.

Results The clustering for k=2 is the same independent of the choosen initialization method. As an example the figure for the kmeans++ initialization is displayed (see Figure 12). As one can see the Kmeans-Algorithm has basically divided the instances into two classes with the first class containing all movies with a cardinality of the cast over 25 and the second one containing all movies with with a cardinality of the cast under 25.

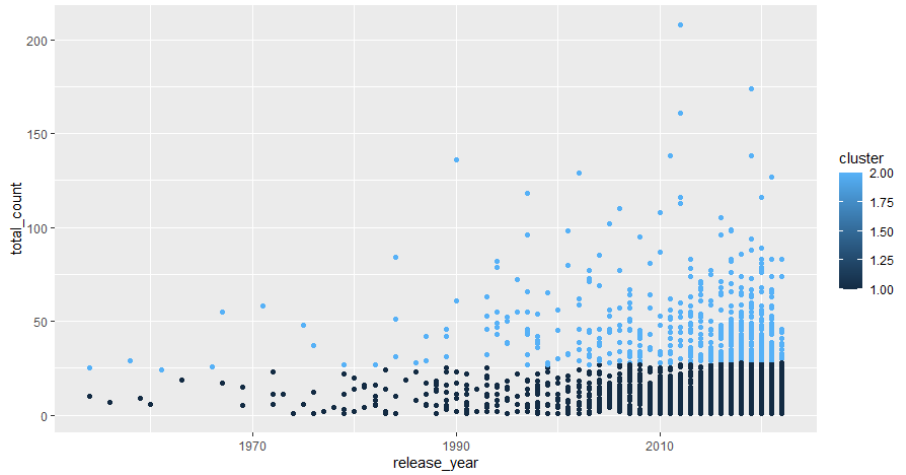


Figure 12: Cluster Result for the kmeans++ initialization

3.3 DBScan

Setup of Experiment Before running the DBScan algorithm on the data two algorithm-specific hyperparameters have do be determined: **Epsilon** and **minPoints**. For this we use the known heuristic from the lecture to plot a k-dist graph with k=4 the k and identify the visual 'elbow-point'. Using this gained information we set $\epsilon = k - dist(x)$ and minPoints = k. The k-distance graph for the chosen features is displayed below.

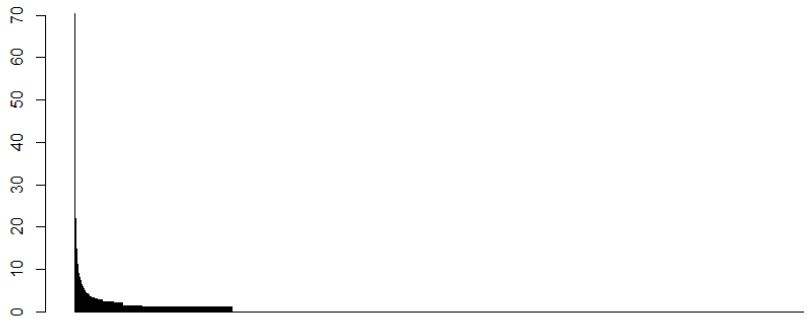


Figure 13: K-Distance Graph for minPoint = 4

One can observe that the first big dip of the graph is finished for a approximated distance of 10. This is the value choosen for ϵ. With this we have determined both necessary hyperparameters needed to start running the DBScan algorithm.

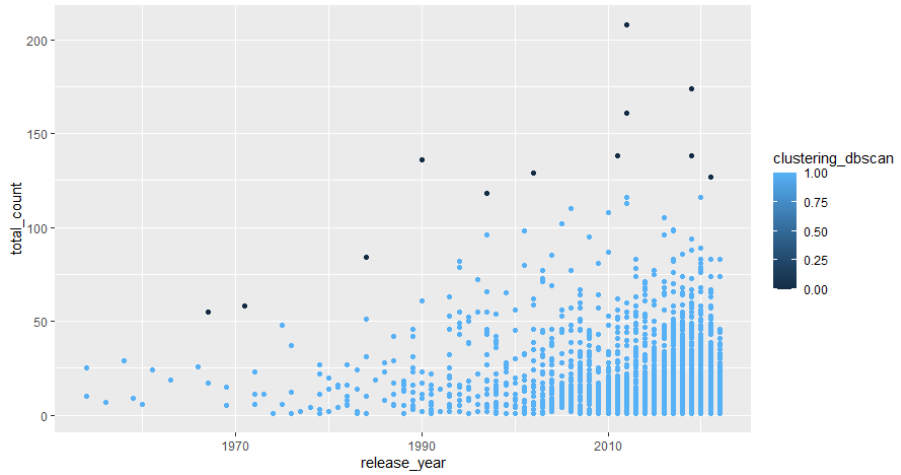


Figure 14: Result of the DBScan algorithm

Results The DBscan Algorithm seems to have a trend in the data showcasing that the cardinality of the cast of the movies increases in recent years. Based on this it clustered the data into two classes with the first class containing movies which exceed the approximate linearity of this trend and movies which fall below this trend line.

3.4 Result Analysis

The Kmeans Algorithm and the Dbscan seem to have identified two different kind of clusters in the data. This could be explained by the fact that the cluster identified by the Dbscan are non-konvex which the Kmeans algorithm is know to not be able to identify.

References

[IMD24] IMDb. *Ratings FAQ*. Opened: 2025-01-12. 2024. URL: https://help.imdb.com/article/imdb/track-movies-tv/ratings-faq/G67Y87TFYYP6TWAV?ref_=helpms_helpart_inline#.

[Nee25] Col Needham. *IMDb Top Data Contributors for 2024*. Opened: 2025-01-11. 2025. URL: https://community-imdb.sprinklr.com/conversations/data-issues-policy-discussions/imdb-top-data-contributors-for-2024-and-happy-new-year-2025-from-imdb/6774f63f02e6d86e38658477.

[TMD24] TMDB. *Consensus*. Opened: 2025-01-12. 2024. URL: https://www.themoviedb.org/consensus.