Exploring The Movie Database (TMDb): A text mining approach to clustering and topic modeling for movie recommendations

Francesca Del Giudice[1]*, Sara Nava[2]*, Giulia Saresini[3]*

Abstract
This paper explores the application of text mining techniques to analyze The Movie Database (TMDb), a comprehensive repository of movie-related information, with the dual objectives of uncovering similarities among genres and developing a movie recommendation system. Text clustering methods, such as k-means, fuzzy c-means and hierachical clustering, are employed to group movies based on their plot, revealing patterns and relationships across genres and themes. Additionally, topic modeling techniques are utilized to extract latent semantic structures from movie descriptions, enabling the identification of key themes and the construction of a personalized recommendation framework.
This analysis aims to serve as an application of classical text mining techniques for secondary analyses and the development of practical applications.

Keywords
Text Vectorization –Text Clustering – K-Means – Fuzzy c-Means – Hierarchical Clustering – Topic Modeling – Movie Recommendation System – LDA

* *Università degli Studi di Milano-Bicocca*
[1] *f.delgiudice3@campus.unimib.it, 912367*
[2] *s.nava38@campus.unimib.it, 870885*
[3] *g.saresini@campus.unimib.it, 864967*

Contents

Introduction

The movie industry has long been a cornerstone of global entertainment, producing vast amounts of content that spans a diverse range of genres, themes, and styles. Understanding the relationships between films, uncovering hidden patterns, and providing personalized recommendations are essential tasks in this domain, driven by the increasing availability of large-scale movie databases.
One such repository, The Movie Database (TMDb)[1], offers an extensive dataset containing valuable information about movies. Analyzing this data can yield insights into thematic clustering, genre similarities, and audience preferences, which can be leveraged to enhance user experiences.

[1] https://www.kaggle.com/datasets/asaniczka/tmdb-movies-dataset-2023-930k-movies/data

This study investigates the use of text mining techniques to analyze the TMDb dataset, with a focus on two main objectives: adopting clustering algorithms to identify genres similarities based on movie's plot and enhancing recommendation systems through topic modeling. The goal is to demonstrate how clustering can help discover genres similarities and how topic modeling can uncover latent themes, contributing to more effective recommendation systems. By integrating these methods, the study showcases the value of natural language processing (NLP) in understanding movie data and provides practical insights for applications in the entertainment industry.

1. Dataset

The dataset used for the analysis is **The Movie Database (TMDb)**, a comprehensive movie database that provides information about movies, including details like titles, ratings, release dates, revenue, genres, overview, keywords and much more. It consists of around one million records and 24 variables.
The most important features needed in this project are:

- *title:* title of the movie,
- *overview:* brief description or summary of the movie,
- *genres:* list of genres the movie belongs to,
- *keywords:* keywords associated with the movie.

It is important to note that the analyses presented here are based on the version of the dataset from January 4, 2025. Any subsequent updates or changes to the dataset, such as the addition of new movies or modifications in genre classification, will not be reflected in this analysis.

2. Text Clustering

The main goal of this section is to use clustering algorithms on movie data, particularly focusing on genres, to explore and capture genre similarities. By applying various clustering techniques (such as k-means, fuzzy c-means, and hierarchical clustering), the aim is to uncover patterns of relationship structures between different genres, helping to identify clusters that might not be immediately obvious based on conventional genre classification alone.

2.1 Data cleaning for clustering task

The following steps were taken to clean the dataset:

- **Removal of missing values:** the rows with missing values (NaN) or with empty cells in the columns *genres* and *overview* are removed to ensure the analysis is conducted on a complete dataset.
- ***overview* cleaning:** filter the data in such a way that the length of the overview was limited to between **250 and 600 characters**. This decision was made to remove **overly short texts**—either noise or overly concise summaries omitting key details and terms—and to avoid **overly long overviews**. Long overviews could **slow down the performance** of algorithms like **t-SNE** and impact the **Sentence Transformer embeddings** by introducing **redundancy** that might confuse the models.
- ***genres* cleaning:** create binary columns for each unique genre, where each column will indicate whether a particular genre is present in the *genres* column. For instance, if a movie is associated to *['drama', 'action', 'crime']* genres, then the dataset will have the following aspect:

Comedy	...	Action	Drama	Crime
0	...	1	1	1

Table 1. Example of New Genres Visualization

The final *genres* distribution is the following:

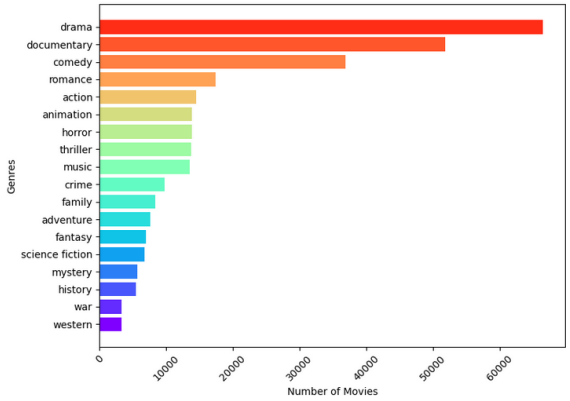

Figure 1. Final *genres* distribution.

2.2 Text Embeddings

To perform clustering, it is necessary to first convert the textual data of *overview* into numerical representations (**embeddings**). For this scope the pre-trained `all-MiniLM-L6-v2` [2] model is used. The `all-MiniLM-L6-v2` model is useful in this context due to its capability to generate high-quality sentence embeddings. These embeddings effectively **capture semantic similarities**, which is crucial for understanding the relationships between movie overviews. Here's how the text processing workflow is structured:

1. **Input Tokenization:** The input sentence is first divided into tokens (either words or subwords).
2. **Transformer Encoding:** The tokens are passed through a transformer model, which generates context-aware embeddings for each token.

[2] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

3. **Pooling:** The individual token embeddings are aggregated to form a single vector representing the entire sentence.

4. **Output:** The final output is a fixed-size vector that serves as a compact, contextually rich representation of the sentence, regardless of its length.

2.3 Text Visualization

To visualize textual data after the embedding process (which generates arrays of 384 dimenisons) it is necessary to reduce the dimensionality of the embeddings in a 2D space. For this aim, the dimensionality reduction technique applied is `t-SNE`. The algorithm performs a nonlinear dimensionality reduction, analyzing the data components and selecting the first two, which are then displayed on the x-y axis for visualization.

Moreover, given the large volume of data, a **random sample of 1000 movies** per genre was extracted to ensure a manageable dataset while maintaining representativeness. The following steps were taken to create the sample:

1. **Filter Data:** Focus on one genre at a time, and select movies where the specific genre column is set to 1.

2. **Treat Each Sample as a Single Genre:** Each sample is considered representative of a single genre, ignoring any overlap with other genres (because a movie can belong to more than one).

By visualizing all the data in a 2D space, the following distribution can be observed (Figure 2:

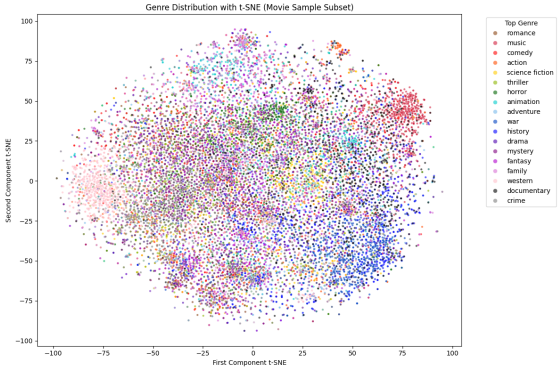

Figure 2. Movie Genres Distribution.

It is clear that there is a significant overlap in the data, and that the genre clusters are not particularly distinguishable. This is not surprising, as films naturally span multiple genres, often creating more complex and specific genre combinations. Consequently, it's difficult for a single genre to have a strong identity within its own cluster. As seen in the graphs below, genres like **music** (Figure 3), **war** (Figure 4), and **western** (Figure 5) are more compact, likely due to very specific themes, while genres like **comedy** (Figure 6), **adventure** (Figure 7), and

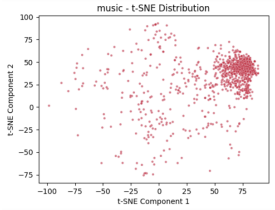

Figure 3. Music Genre.

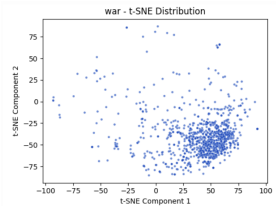

Figure 4. War Genre.

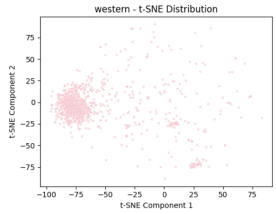

Figure 5. Western Genre.

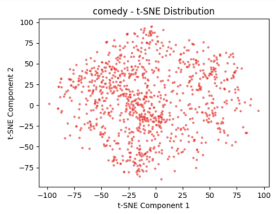

Figure 6. Comedy Genre.

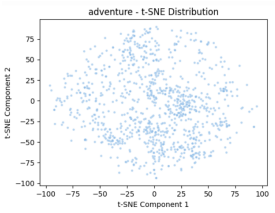

Figure 7. Adventure Genre.

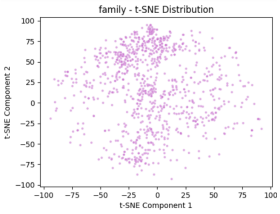

Figure 8. Family Genre.

family (Figure 8), which are more generic and frequently accompany other genres, are notably more dispersed.
Given this, an interesting next step could be to explore whether there are groups of genres that tend to cluster together more closely in the clusters that will be identified.

2.4 K-means algorithm

K-Means algorithm is a popular unsupervised clustering technique used for partitioning datasets into k distinct clusters.

2.4.1 Algorithm

The K-Means algorithm follows these steps:

1. **Initialization:** Randomly select k points, centroids, that represent initial document vectors.

2. **Assignment Step:** Assign each data point (document vector) x_i to the cluster with the closest centroid c_j, based on a distance measure such as cosine similarity or Euclidean distance:

$$\text{Cluster}(x_i) = \arg \min_{j \in \{1,2,\ldots,k\}} \text{distance}(x_i, c_j)$$

3. **Update Step:** Recalculate the centroids as the mean of all document vectors assigned to the cluster:

$$c_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$

Here, C_j is the set of document vectors assigned to cluster j, and $|C_j|$ is the number of vectors in the cluster.

4. **Repeat:** Continue the assignment and update steps until centroids stabilize or the maximum number of iterations is reached.

The goal of K-Means is to minimize the within-cluster sum of squares (WCSS), defined as:

$$\text{WCSS} = \sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - c_j\|^2$$

For text clustering, the distance metric can be adapted to better capture semantic relationships, such as cosine distance:

$$\text{Cosine Similarity} = \frac{\vec{x_i} \cdot \vec{c_j}}{\|\vec{x_i}\| \|\vec{c_j}\|}$$

One of the main limitations of K-means is that it requires the number of clusters (k) to be predefined. To address it, the **Elbow Method** is used to determine an optimal value for k. The plot (figure 9) shows the optimal k with respect to the intertia (sum of squared distances to the nearest cluster center).

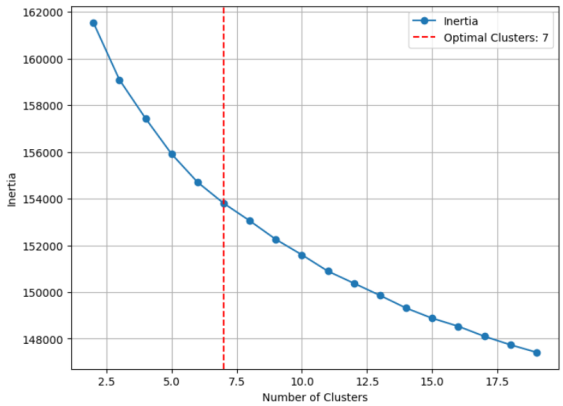

Figure 9. Elbow method for optimal cluster number selection.

Around 7 clusters, the rate of decrease in inertia starts to slow down, forming the elbow. This indicates that adding more clusters beyond this point does not significantly improve the clustering quality.

2.4.2 Evaluation

To evaluate the quality of the clustering results two common metrics are used:

- **Silhouette Score:** measures how similar data points are within the same cluster compared to those in other clusters. A higher score (close to 1) indicates well-separated and cohesive clusters, while a lower or negative score suggests overlapping or poorly defined clusters.
- **Davies-Bouldin Index:** evaluates the average similarity ratio of each cluster to its most similar cluster. A lower score indicates better clustering performance, with more distinct and compact clusters.

These metrics provide insights into the effectiveness of the clustering algorithm and help assess whether the clusters represent the data structure effectively.

Metric	Value
Silhouette Score	0.0177
Davies-Bouldin Index	5.541

Table 2. K-means evaluation metrics

The results from the clustering evaluation are not satisfactory. The Silhouette score is very close to zero, indicating that the clusters are poorly separated, while the Davies-Bouldin Index is relatively high, suggesting that the clusters are very dispersed and overlapping.
Try also to assess the alignment between the clustering results and the ground truth genres with **Adjusted Rand Index** (ARI). The ARI measures the similarity between the clusters and the true labels. ARI assumes values ranging from -1 to 1:

- 1: Perfect agreement (clusterings are identical).
- 0: Random agreement (no better than chance).
- $<$0: Worse than random (systematic disagreement).

In this particular case, one can compare the binary columns representing individual genres with the clustering column obtained from the application of **k-means** on the data (which was subsequently added to the dataset). However, the results are relatively poor and disappointing, as illustrated in the table 3.

Genre	ARI
Documentary	0.056
Music	0.023
Drama	0.020
Action	0.009
...	...
Mystery	0.000
Horror	-0.001
Fantasy	-0.001
Romance	-0.003

Table 3. Sorted ARI for movie genres

The analysis reveals that the clustering outcomes are disappointing. None of the calculated indices indicate meaningful or robust results. This aligns with expectations set during the preliminary analysis, where several critical limitations of the data were identified:

- **Genre overlap:** the multilabel classification of films across multiple genres has led to significant overlap.

This overlap diminishes the ability to distinctly separate data points into coherent clusters.

- **Lack of separation:** the overlapping nature of genres results in data points that are closely packed and broadly distributed throughout the embedding space. This lack of separation directly undermines the effectiveness of clustering.

Given these constraints, the current clustering approach fails to provide insightful or actionable results. It is evident that alternative methodologies must be explored.

2.4.3 Cluster visualization

To identify patterns, it can be beneficial to shift the perspective by visualizing the genre distribution, as shown in Section 2.3, with movies color-coded based on their assigned clusters. Keep in mind that, for the reasons previously explained, the visualization only includes the random sample extracted and reduced in dimensionality.
The results of the clustering closely align with the initial expectations and previous index calculations. As anticipated, no well-defined clusters emerged, with data points often scattered randomly across the space. While certain dense regions of data points can be observed, they are typically too close to other clusters, preventing clear separation.
Analyzing each cluster, it emerges as follows:

- **Cluster 0 (Red)**: a highly concentrated zone for documentary and especially music (figure 10), that dominates this cluster.
- **Cluster 1 (Orange)**: a dense region for drama and romance (figure 11), followed by thriller, comedy, and crime in lesser concentrations.
- **Cluster 2 (Yellow)**: The cluster under consideration primarily includes films belonging to the comedy, family, and animation (figure 12) genres. However, for each genre, the points assigned to this cluster are rather dispersed.
- **Cluster 3 (Green)**: weakly represented across most genres, with few films present in the lower part of the graph and in case like mystery (figure 13) are very close to each others.
- **Cluster 4 (Cyan)**: genres such as crime (figure 14), mystery, thriller, and action are more concentrated within this cluster, forming a somewhat cohesive group.
- **Cluster 5 (Dark blue)**: encompasses a variety of genres, including documentary (figure 15), science fiction, fantasy, animation, history, and war. Documentary films are particularly dense in this region, while the other genres are more dispersed.
- **Cluster 6 (Purple)**: similar to Cluster 3, war, documentary (figure 15), and history are the main genres represented, though not tightly clustered.

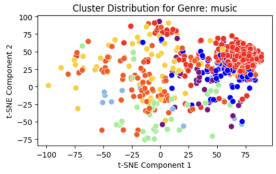

Figure 10. Music.

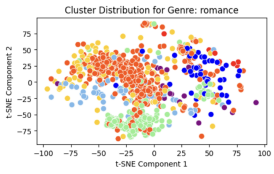

Figure 11. Romance.

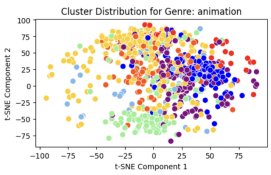

Figure 12. Animation.

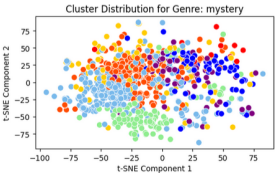

Figure 13. Mystery.

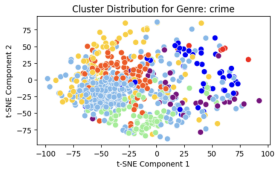

Figure 14. Crime.

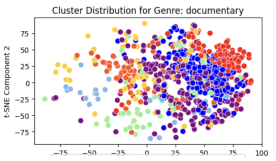

Figure 15. Documentary.

It is clear that, although some clusters appear more compact, the clusters are generally not well-separated. This outcome is not surprising, as the preliminary analysis already highlighted the complexity of the problem.

2.4.4 Genre distribution

At this point, the focus shifts from analyzing the clusters themselves to examining the distribution of genres across clusters. The relative frequency distribution of each genre is visualized within the clusters to assess how genres are spread and identify patterns of similarity between them. This analysis helps to uncover closely related genres by observing genre combinations in clusters.
By using relative frequencies instead of absolute frequencies, the analysis accounts for genre imbalance, ensuring a more accurate representation of genre distribution within clusters. The result is a beginning approach for a deeper understanding of how genres relate to each other across the dataset.
In each cluster, specific behaviors can be observed, including: *genre predominance*, where one genre dominates the cluster; *higher presence of multiple similar genres*, indicating thematic affinity; and *heterogeneous proportions*, where movie distributions across genres are relatively balanced. To illustrate, a single example for each of these behaviors is displayed for visualization purposes.

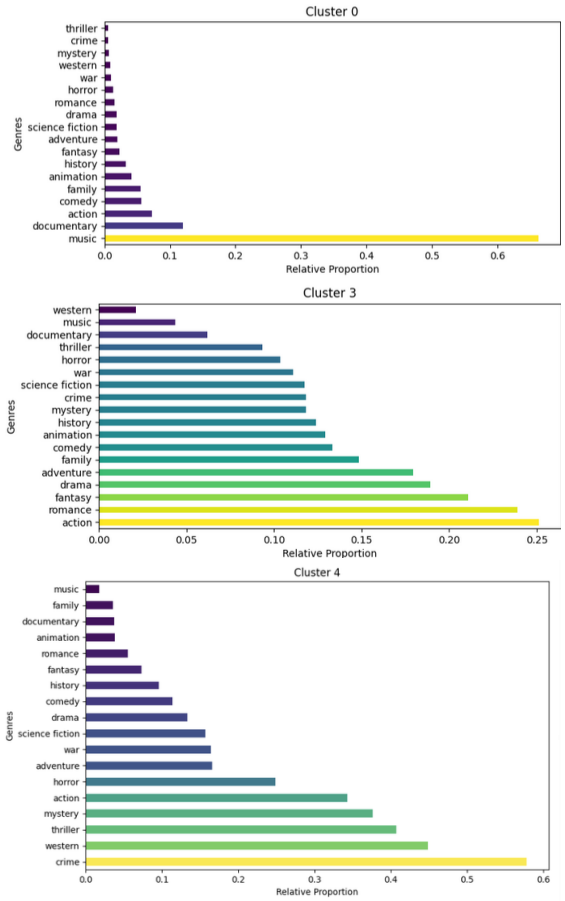

Figure 16. Genre distribution within most representative clusters.

- **Cluster 0**: The individuality of the *music* genre stands out, with over 60 % of films from this genre concentrated in this cluster. Other genres are almost entirely absent, indicating that music films are likely highly isolated and show little affinity with other genres.
- **Cluster 3**: This cluster exhibits a more diverse composition with smaller portions of various genres (lower than 25 % of movies for each one). The most present genres are *romance*, *action*, and *fantasy*, but their proportions are not significantly greater than those of other genres. The fact that the distributions show little variation results in a heterogeneous cluster.
- **Cluster 4**: This cluster shows a higher predominance of a few genres, specifically *crime*, *western*, and *thriller*, with more than 40 % of their movies classified in this cluster. These proportions are significantly greater than those of other genres, suggesting possible thematic similarities among these three genres.

The results demonstrate the varying degrees of thematic coherence within clusters. While some genres align strongly with specific clusters, others are more dispersed. A deeper investigation into genre relationships and overlaps could further enhance understanding of the clusters' composition.

2.4.5 Genre correlation analysis

Given these considerations and the interesting insights from analyzing the distribution of movies within each genre across the identified clusters, it's necessary to find a way to understand which genres the clustering algorithm has identified as more similar.

To achieve this, the approach was modified to calculate not the **relative frequencies** of each genre within each cluster (which involves counting the movies for each genre in each cluster and dividing by the genre's size), but instead to calculate the **relative frequency** of each cluster within the genres (where, for a fixed cluster, the count of movies in each genre is divided by the cluster's size).

The formula for the relative frequency of a **cluster** within each genre is given by:

$$f_{\text{cluster}}(g) = \frac{\text{Number of movies of genre } g \text{ in cluster } c}{\text{Total number of movies in cluster } c}$$

Calculating the relative frequency based on the total size of the cluster allows to observe how genres are distributed across clusters and how each cluster can represent a **"combination"** of genres, which is the key information in the context of clustering. If the relative frequency were calculated based on the total size of the genre, it would provide a less insightful view of the relationships between genres in the clusters and could distort the interpretation of the clusters. This is because such an approach would only focus on the overall distribution of genres, which was already analyzed earlier to understand the assignment of movies to genres within each cluster.

After this step, the correlation between genres was calculated based on the proportions derived, considering only the top three most correlated genres for each genre. The correlation between genres g_1 and g_2 across clusters can be calculated using the Pearson correlation coefficient:

$$r_{g_1,g_2} = \frac{\sum_{c=1}^{C} \left(f_{\text{cl}}(g_1,c) - \bar{f}_{g_1}\right)\left(f_{\text{cl}}(g_2,c) - \bar{f}_{g_2}\right)}{\sqrt{\sum_{c=1}^{C} \left(f_{\text{cl}}(g_1,c) - \bar{f}_{g_1}\right)^2}\sqrt{\sum_{c=1}^{C} \left(f_{\text{cl}}(g_2,c) - \bar{f}_{g_2}\right)^2}}$$

Where:

- r_{g_1,g_2} is the correlation between genres g_1 and g_2,
- C is the total number of clusters,
- $f_{\text{cl}}(g,c)$ is the relative frequency of genre g in cluster c,
- $\bar{f}_g$ is the mean relative frequency of genre g across all clusters.

Once this was done, a graph (figure 17) was created to visualize the genre correlations within the clusters, with nodes and edges added based on the correlation values from the

DataFrame. Each edge represents the correlation between two genres, and its weight corresponds to the correlation value. Moreover, the size of the nodes is proportional to their **degree**, and the thickness of the edges reflects the strength of the correlation. In this way, based on clustering results, it is possible to find out different possible **genre similarities**.

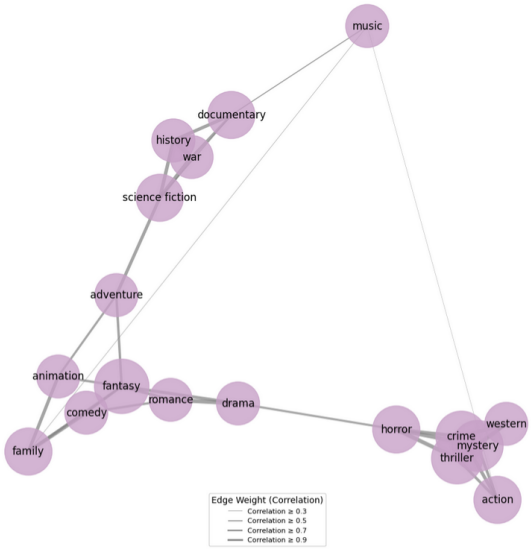

Figure 17. Genre distribution within each cluster.

Upon analyzing the cluster distributions and the frequency-based correlations, several **groups of similar genres** can be observed:

- **Music**: As expected from previous analyses, **music** remains highly isolated in the dataset, with very low correlations to other genres. This reinforces its **distinctiveness** in comparison to other genres.
- **War, Documentary, Science Fiction, and History**: The first group consists of **war**, **documentary**, **science fiction**, and **history**. The relationship between these genres is intuitive:
  - **War** films often depict historical events, and **history** can be represented both dramatically and in a **documentary** format.
  - However, **documentary** and **science fiction** show a relatively **weaker correlation** with each other.
- **Adventure, Animation, Family, Fantasy, Comedy, Romance, and Drama**: A larger and more dispersed group includes **adventure**, **animation**, **family**, **fantasy**, **comedy**, **romance**, and **drama**. These genres tend to be more **generic** and frequently co-occur. For instance, **animated films** are often associated with **family** films, while **romance** and **comedy** often overlap in terms of audience and thematic elements.
- **Horror, Thriller, Crime, Mystery, Western, and Action**: The final group includes **horror**, **thriller**, **crime**, **mystery**, **western**, and **action**. These genres are often **interrelated** due to thematic overlap, such as the **suspense** or **dangerous situations** central to both **thriller** and **horror**. Similarly, **crime** and **mystery** are highly correlated due to their focus on **investigative** and **criminal** elements. **Western** and **action** films also share a focus on **adventure** and physical confrontation, though they are less closely tied than other genres in this group.

These observations align with the **calculated genre correlations**, revealing natural **clusters based on thematic similarities**.

At this point, applying different clustering techniques will help provide new insights and potentially more meaningful results.

2.5 Hierarchical Clustering

Hierarchical clustering [3], also known as hierarchical cluster analysis (HCA), is a method used to build a hierarchy of clusters. It is widely used in data mining and statistics to organize data into a tree-like structure called a dendrogram.

The key decision in hierarchical clustering is how to measure the dissimilarity between clusters. Several linkage criteria are available, which influence the shape and structure of the resulting clusters. These include:

- **Complete-Linkage Clustering (Maximum linkage)**: The dissimilarity between two clusters is the maximum distance between any two points, one from each cluster.
- **Single-Linkage Clustering (Minimum linkage)**: The dissimilarity is the minimum distance between any two points, one from each cluster.
- **Average Linkage (UPGMA)**: The dissimilarity is the average distance between all pairs of points, one from each cluster.
- **Ward's Method**: This method minimizes the total within-cluster variance, i.e., it merges the clusters in a way that minimizes the increase in the sum of squared errors (SSE) after each merge.

The choice of linkage criterion can have a profound impact on the clustering result. For example, complete-linkage tends to produce more compact, spherical clusters, whereas single-linkage can create elongated or chain-like clusters.

Ward's method is a popular agglomerative hierarchical clustering approach that focuses on minimizing the variance within each cluster. At each step, it **merges** the **two clusters** whose combination results in the **least increase** in the **total within-cluster variance**. The mathematical formulation for the distance between two clusters A and B in Ward's method is given by:

[3] https://en.wikipedia.org/wiki/Hierarchical_clustering

$$d(A,B) = \frac{|A||B|}{|A \cup B|} \|\mu_A - \mu_B\|^2$$

where μ_A and μ_B are the centroids of clusters A and B, respectively, and $|A|$ and $|B|$ are the sizes of the clusters. This method is particularly effective when the goal is to create clusters that are as homogeneous as possible in terms of their internal variance.

2.5.1 Visualization and Interpretation

The dendrogram (18) is the primary visualization tool for hierarchical clustering. It provides an intuitive representation of how clusters are formed at various levels:

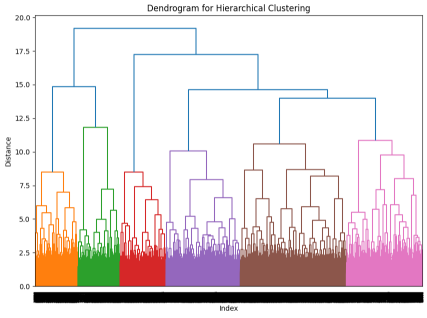

Figure 18. Dendrogram.

- **Leaf nodes**: represent individual movies clusters.
- **Branches**: indicate groupings of similar movies based on text features.
- **Height of branches**: reflects the dissimilarity between clusters, with shorter branches indicating higher similarity.

In this analysis, the focus is on two scenarios: clustering with $k = 4$ and $k = 8$ clusters.

2.5.2 Number of clusters: 4

Since, as previously mentioned, evaluating model performance using indices such as **Silhouette**, **Davies-Bouldin**, and **ARI** doesn't make sense — both because these metrics wouldn't be meaningful for the given data and because they are no longer relevant for the analyses that have been decided to conduct — proceed, as with **KMeans**, by analyzing how genres are distributed within the clusters. This is done by examining the **correlation** between genres based on the assignments generated by the clustering.

The genre distribution of the 4 clusters are plotted in figure 19.

In the first cluster (`cluster 0`), not many films seem to have been assigned, with the two most prevalent genres being **history** and **war**, each accounting for around **40%** and **60%** of the observations. Once again, there is a cluster (`cluster 1`) with a predominance of **music** films, with about **50%** of

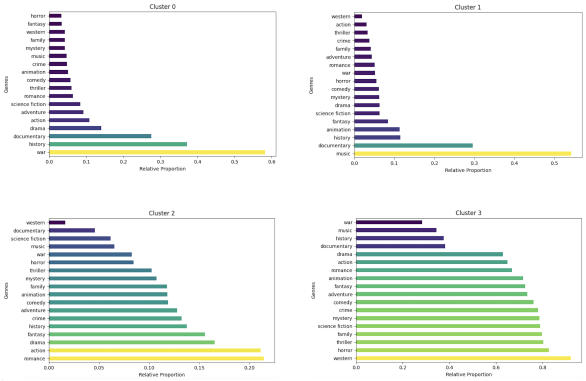

Figure 19. Genres Distribution in each Cluster.

films in this category, with other genres present at significantly lower levels, except for **documentary** with an honest **30%** of observations. The remaining `clusters`, `2` and `3`, appear to have relatively **balanced and low percentages across all genres**, with minimal differences and no clear predominance of any particular genre. This last observation suggests that the clustering isn't working well, randomly assigning movies into clusters.

Based on the results, the clusters do not seem to capture any distinct similarities between genres, likely due to an insufficient number of clusters in which the films were grouped.

Now, apply the same procedure used for K-Means to assess genre similarity, focusing on the correlations of relative frequencies within clusters.

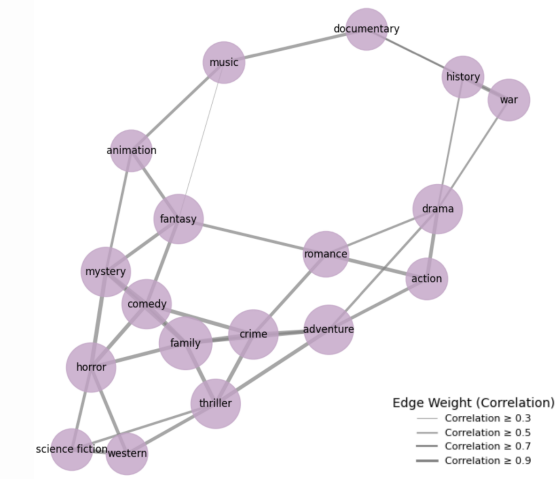

Figure 20. Genre similarity (HC - 4 Clusters).

As shown in Figure 20, it is clear that the result is **not well-defined**: all **genres** appear to be **highly correlated with one another**, and there are no clear, well-defined groups of similar genres. The genres that seem somewhat more separated are **history**, **war**, and **documentary**, which are quite close to each other and reasonably sensible in their association. The **music** genre also appears more separated, though still relatively close,

suggesting a degree of isolation in its clustering.
The result obtained does not seem as satisfactory as it appeared for the k-means clustering, because it is rather uninformative and unable to capture genre groups. Therefore, further techniques will be tested to improve the analysis and gain more meaningful insights.

2.5.3 Number of clusters: 8

Following the same approach as before, analyze the distribution of movies from each genre within the 8 identified clusters.

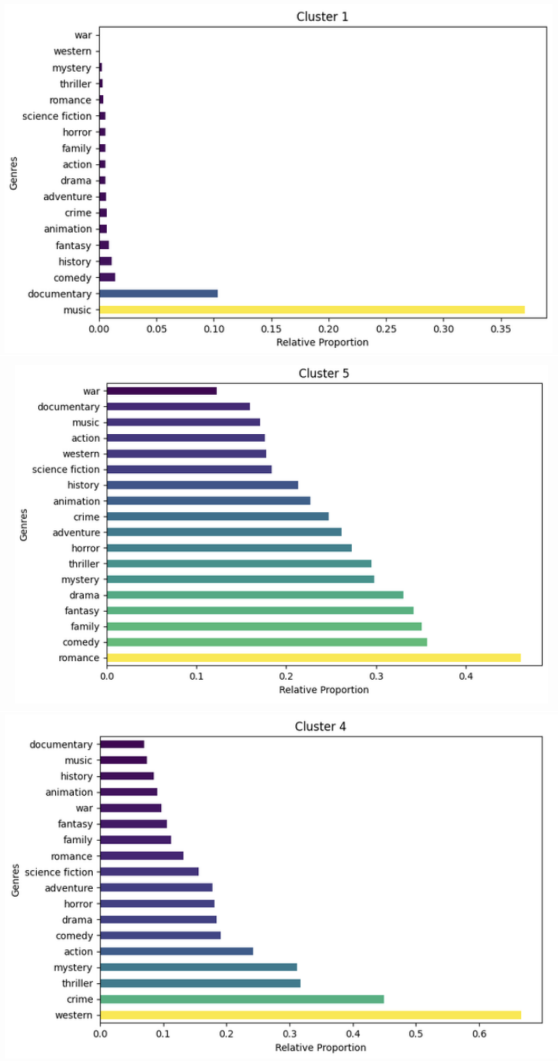

Figure 21. Genre distribution within most representative clusters.

The distribution of genres within the clusters reveals distinct trends and highlights the limitations of the clustering approach. Cluster 0 is predominantly composed of **war** films, with **history** making up roughly half of the representation. Cluster 1 is strongly characterized by **music**, which emerges as the most distinct and well-defined genre in the dataset (first graph in Figure 21).

Cluster 4 (shown in the last bar plot of Figure 21), although having smaller overall proportions, exhibits higher percentages of **western**, **mystery**, and **crime** films, suggesting a thematic concentration. **Western** makes up more than 60% of the observations in this cluster.
In contrast, clusters 2, 3, and 5 (the last one represented in the second bar plot of Figure 21) show weaker patterns due to balanced distributions. In the first two clusters, all genres are represented at very low levels, around 20% for the most frequent genres, while in cluster 5, the percentages are slightly higher. This suggests that these clusters may consist of observations that were classified more arbitrarily, lacking a clear thematic focus.
Similarly, cluster 6 has very low proportions of movies in each genre, with values lower than 10%, except for the predominant genre, **horror**, which comprises only 25% of the movies. In cluster 7, the most frequent genres are **science fiction**, **animation**, and **family**, each accounting for around 30% of the movies, a proportion that is not significantly high.

The observed results suggest that while some clusters capture dominant genres effectively, others fail to provide clear or significant insights due to their heterogeneous nature. This is reflected in clusters with similar percentages across genres, and those that align better with related genres.

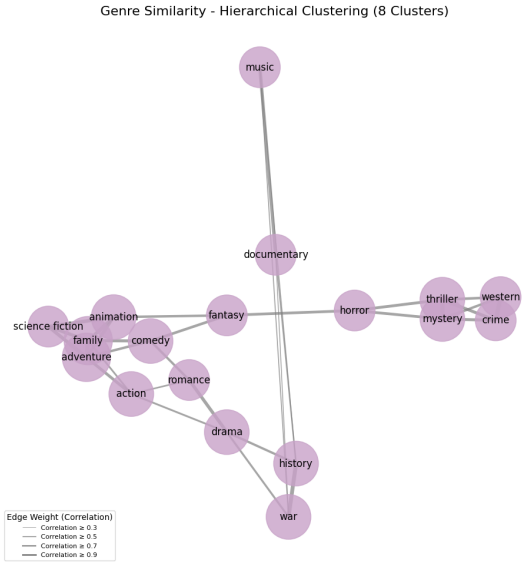

Figure 22. Genre similarity (HC - 8 Clusters).

Consider the graph at Figure 22, which shows genre similarity based on correlations. The current clustering results show an improvement compared to the **hierarchical clustering** approach with **four** clusters. Smaller, more distinct genre groups have emerged, adding some clarity to the structure of the data. A notable grouping includes **crime**, **western**, **mystery**, **thriller**, and **horror**, closely resembling a similar cluster observed using **k-means**. Meanwhile, a larger, more dispersed group comprises the remaining genres, with **documentary** and **music** continuing to exhibit isolation from other

genres.

Despite these improvements, a significant issue persists: the **high correlations between genres**. This likely stems from earlier observations on the relative frequency distributions of genres across clusters, which suggested arbitrary clustering rather than meaningful separations.

Given the **multi-label nature of the problem**, a promising next step would be to explore **soft clustering techniques**. These methods could account for the overlapping nature of the data, potentially yielding better-aligned clusters and more informative insights into the genre relationships within the dataset.

2.6 Fuzzy C-Means

Fuzzy clustering [4] (also referred to as soft clustering or soft k-means) is a form of clustering in which each data point can belong to more than one cluster.
Clustering or cluster analysis involves assigning data points to clusters such that items in the same cluster are as similar as possible, while items belonging to different clusters are as dissimilar as possible. Clusters are identified via similarity measures, such as distance, connectivity, and intensity. Different similarity measures may be chosen based on the data or the application.

Membership grades are assigned to each data point (tag). These membership grades indicate the degree to which data points belong to each cluster. Points on the edge of a cluster, with lower membership grades, may be in the cluster to a lesser degree than points in the center of the cluster.

One of the most widely used fuzzy clustering algorithms is the **Fuzzy C-means clustering** (FCM) algorithm, applied in this project.
The fuzzy c-means algorithm is very similar to the k-means algorithm:

- Choose a number of clusters.
- Assign coefficients randomly to each data point for being in the clusters.
- Repeat until the algorithm has converged (i.e., the coefficients' change between two iterations is no more than ε, the given sensitivity threshold):
  - Compute the centroid for each cluster.
  - For each data point, compute its coefficients of being in the clusters.

Any point x has a set of coefficients giving the degree of being in the k-th cluster, $w_k(x)$. With fuzzy c-means, the centroid of a cluster is the mean of all points, weighted by their degree of belonging to the cluster. Mathematically, the centroid is defined as:

$$c_k = \frac{\sum_x w_k(x)^m x}{\sum_x w_k(x)^m},$$

where m is a hyperparameter that controls how fuzzy the cluster will be. The higher the value of m, the fuzzier the cluster will be in the end.

The FCM algorithm attempts to partition a finite collection of n elements $X = \{x_1, \ldots, x_n\}$ into a collection of c fuzzy clusters with respect to a given criterion.
The algorithm returns a list of c cluster centers $C = \{c_1, \ldots, c_c\}$ and a partition matrix $W = w_{i,j} \in [0,1]$, where each element w_{ij} tells the degree to which element x_i belongs to cluster c_j.
The FCM aims to minimize an objective function:

$$J(W,C) = \sum_{i=1}^{n} \sum_{j=1}^{c} w_{ij}^m \|x_i - c_j\|^2,$$

where

$$w_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}.$$

2.6.1 Fuzzy C-Means Application

To identify the optimal number of clusters, the elbow method is employed as made for k-means.

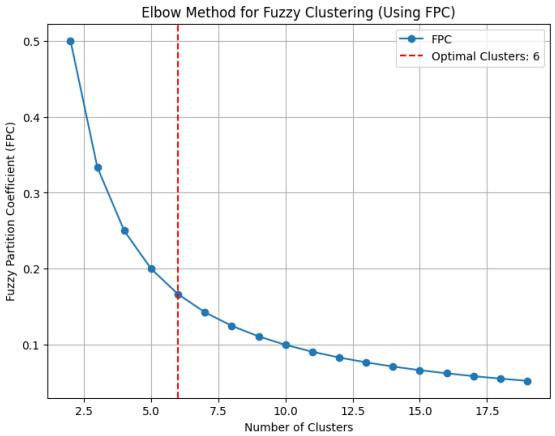

Figure 23. Elbow Method (Fuzzy c-means).

In this case, $C = 6$ represents the most appropriate number of clusters in which group movies.
The examination of genre proportions within each cluster reveals distinct patterns but also highlights potential issues of overgeneralization and randomness in some cluster assignments.

Cluster 0 and Cluster 4: Clusters 0 (second graph in Figure 24) and 4 exhibit very low proportions for each genre, with less than 1.7% of films per genre. This suggests that these

[4] https://en.wikipedia.org/wiki/Fuzzy_clustering

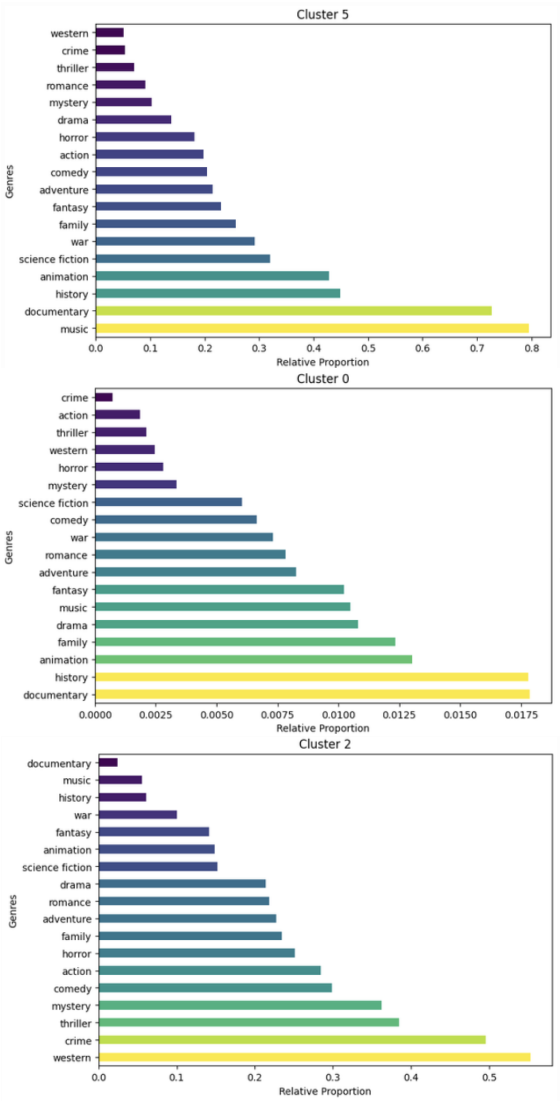

Figure 24. Genre distribution within most representative clusters.

clusters may lack relevance and could predominantly include data points assigned arbitrarily, possibly representing outliers or uncertain cases.

Cluster 1: This cluster demonstrates strong genre associations, with over 40% of war films, around 30% of history, action, science fiction, adventure, and western films. These genres share thematic similarities, reflecting a cohesive grouping.

Cluster 2: Genres like western (50%) and crime (50%) dominate this cluster, followed closely by mystery (40%) and thriller (40%), showcasing realistic affinities between these genres. This distribution can be observed in the last graph in Figure 24.

Cluster 3: Cluster 3 is heavily composed of romance (60%), drama (50%), and fantasy (40%) films. However, moderate proportions of other genres suggest some level of overgeneralization, which may dilute the apparent dominance of specific genres. Thus, it can be considered as an heterogeneus group.

Cluster 5: Cluster 5 (first graph in Figure 24) stands out as the most distinct grouping, with nearly 80% of music films and over 70% of documentaries, alongside moderate contributions from other genres. This clear concentration of certain genres reinforces its uniqueness.

In summary, Clusters 1, 2, and 5 show strong, thematically cohesive groupings, but also, referring to cluster 5, genre distinction, while Clusters 0, 4, and 3 might require further analysis due to their lower genre concentrations or heterogeneity.

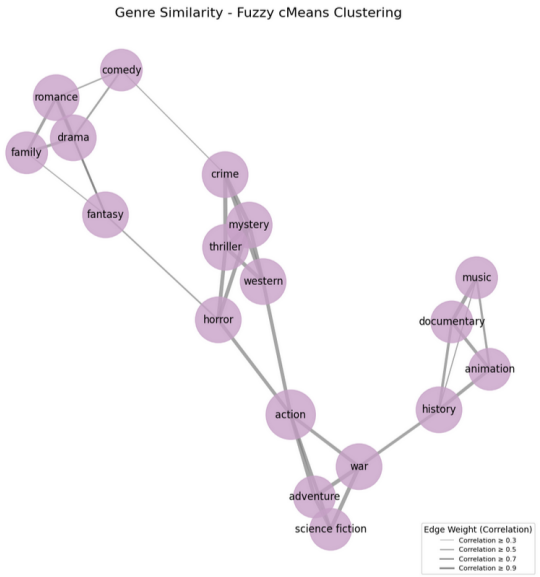

Figure 25. Genre Similarity (Fuzzy 6-means).

The results obtained through *soft clustering* (Figure 25) are similar to those of *k-means*, but genre associations seem to be organized in smaller and more cohesive groups. Several **distinct genre clusters** emerge, reflecting strong thematic connections, while some genres exhibit **unexpected relationships**, providing new perspectives on their affinities.

One cluster groups **documentary, history, animation, and music**, which is particularly noteworthy. The connection between **documentary and history** is intuitive, as they often share factual or narrative overlap. However, the inclusion of **music**, which is typically isolated, is surprising. Its strong correlation with **documentary**, as indicated by their cluster proximity and shared dense regions, highlights a less obvious but meaningful relationship. **Animation**'s place in this cluster is less straightforward but might stem from overlapping distributions. The connection between **music and animation**, though not immediately intuitive, suggests shared patterns worth exploring further. The connection between **music and documentary** is explained by the Figure 24, in the first barplot, where a large proportion of movies of both genres are attributed to the same cluster.

Another well-defined cluster includes **war, action, adventure, and science fiction**, reflecting clear thematic ties. These genres are often united by narratives involving **conflict** or **exploration**, which makes their grouping logical.

The genres **western, horror, thriller, mystery, and crime** form a separate, cohesive cluster with high internal correlations. This grouping is similar to the one observed both in *k-means* and *hierarchical clustering (8 clusters)* and reflects their shared focus on suspense, tension, and narrative intrigue. These genres naturally complement one another, making their clustering consistent across different algorithms.

Finally, a broader cluster of **drama, romance, family, comedy, and fantasy** emerges. This grouping is distinct from others due to its versatility: these genres frequently appear across various narratives, blending with other genres, which makes their clustering less rigid. **Comedy and fantasy** exhibit subtle connections to **crime and horror**, respectively, but the cluster as a whole captures genres that are **generic and highly adaptable**. **Drama and comedy** remain prominent but less discriminative, reflecting their tendency to span multiple genres.

2.7 Results comparison

To compare the clustering results comprehensively, the graphs generated for each algorithm are analyzed and evaluated through key metrics such as **modularity**, **density**, and **clustering coefficient**. These measures help in assessing the quality of the relationships and groupings within each clustering approach, appeared with genres correlation analysis.

- **Modularity (Community detection):** it measures the density of links inside communities compared to links between communities. If the modularity score is high, the graph has clear communities (clusters of genres). If the score is low, then the graph is more evenly distributed, and there are no clear clusters.
- **Graph density:** it says how well-connected the genres are. If the graph has high density (most genres are connected to each other with high correlation), it suggests there's no clear cluster structure. The formula for graph density is:

$$\text{Density} = \frac{2 \cdot E}{N \cdot (N-1)}$$

  Where:

  - E is the number of edges (correlations between genres)
  - N is the number of nodes (genres)

- **Clustering coefficient:** it quantifies how close the genres are to forming clusters by measuring the tendency of genres to form triangles (where three genres are all connected). A higher clustering coefficient means genres tend to cluster together.

Table 4. Comparison of Metrics Across k-Means and Fuzzy c-Means.

Metric	K-Means	Fuzzy C-Means
Modularity Score	0.5792	0.5717
Graph Density	0.2222	0.2157
Clustering Coefficient	0.5796	0.6648

Table 5. Comparison of Metrics Across Hierarchical Clustering Algorithms.

Metric	HC(4)	HC(8)
Modularity Score	0.3392	0.5123
Graph Density	0.2418	0.2157
Clustering Coefficient	0.4722	0.5926

The results underscore the effectiveness of **k-means**, **fuzzy c-means**, and also **hierarchical clustering (8 clusters)** for this application. These methods achieved **higher modularity** than hierarchical clustering with only 4 clusters, signifying more meaningful and well-defined groupings. However, **fuzzy c-means** emerged as particularly effective due to its **higher clustering coefficient**, reflecting better local connectivity and an enhanced ability to capture nuanced overlaps between genres.

In contrast, **hierarchical clustering (4 cluster)** underperformed significantly. The method yielded **lower modularity** and **clustering coefficients**, indicating its struggles to form clear groupings or maintain connectivity within clusters. This result aligns with earlier observations of its inability to clearly separate genres or produce informative clusters. Maybe, the selected number of clusters isn't correct at all: in fact, as it can be seen, a good selection of the number of clusters shows improvement.

K-Means:

- Forms a **closed network structure**, where all genres are interconnected.
- Groups are robust but show more interconnectivity, which might dilute distinctions between clusters.
- Simpler implementation makes it practical while still delivering strong performance.

Fuzzy C-Means:

- Produces a **linear graph structure**, where extremes are not directly linked, reflecting distinct clusters.
- Similarity groups are more **compact** and **better separated**, showcasing its advantage in managing overlapping data.
- Excels in representing the multi-label nature of the data, identifying nuanced relationships.

Hierarchical Clustering (8 clusters):

- Produces a **linear graph structure**, where extremes are not directly linked, reflecting distinct clusters.
- Clusters tend to be **larger and less refined**, often grouping loosely related genres together.
- Its deterministic nature and ability to preserve the hierarchy of relationships make it suitable for exploring nested genre structures.

These results underscore the **strengths** of **soft clustering** for **complex datasets**.

2.8 Future developments

Focusing specifically on genre analysis to capture genre similarity, here are some potential future developments:

- **Contextual Genre Analysis:**Implementing context-aware clustering, where genres are grouped not only by their inherent properties but also by their context within the movie's narrative or production (e.g., genre co-occurrence in the plot, director style, or audience reception).
- **Incorporating Sentiment Analysis:**Incorporate sentiment analysis into genre similarity detection. By analyzing the emotional tones or themes within a movie (using reviews or metadata), it could reveal hidden genre connections based on how movies make audiences feel, rather than just their surface-level categorization.

3. Topic Modeling

The second task of this study focuses on the application of **topic modeling** [9], which was done based on the keywords associated with movies. The objective is to evaluate the potential benefits of incorporating topic-related information into a **movie recommendation system**. Specifically, a basic recommendation system, utilizing **cosine similarity** between movie overviews, serves as the baseline for comparison. To assess the effectiveness of topic modeling, the study employs an **extrinsic evaluation metric** by investigating whether augmenting the movie recommendation system with information regarding the topics (or the top two topics, as will be further discussed) of the movies results in improved recommendations.

3.1 Data cleaning for topic modeling task

The following steps are followed to clean The Movie Database (TMDb):

- **Removal of missing values**: the rows with missing values (NaN) in the columns *keywords* and *overview* are removed to ensure the analysis is conducted on a complete dataset.
- **Removal of duplicates**.
- **Title cleaning**: filter movies with titles consisting only of alphabetic characters.
- **Removal of adult films**: Adult films were removed from the dataset due to the following reason: they represented over 40,000 observations, with very similar and highly repetitive keywords. Unlike other films, the keywords for adult films showed low variability. Applying Latent Dirichlet Allocation (LDA) on these data resulted in only two topics: one specific to adult films and another generic topic that included all other films.

3.2 Topic Modeling Implementation

3.2.1 Keywords Pre-processing

The preprocessing of keywords involves four different steps to prepare the data for further analysis and ensure consistency across the dataset.

1. **Step 1: Handling Missing Values**
   First, any missing values in the **keywords** column were handled by replacing them with an empty string. This step was necessary to avoid errors or inconsistencies when processing the data.

2. **Step 2: Removing Unnecessary Contextual Information**
   Next, the **keywords** were carefully processed to remove unnecessary contextual information, specifically any text enclosed in parentheses. This was achieved by identifying and eliminating such content, which could potentially introduce noise or irrelevant details into the analysis. This ensures that only the core **keyword** remains for each entry, providing a cleaner and more focused dataset.

3. **Step 3: Handling Compound Keywords**
   In the preprocessing phase of the **keywords**, a decision had to be made regarding the handling of compound **keywords** (such as '*world_war_ii*'). The final choice was to split these compound **keywords**, as we wanted to avoid situations where topics related to wars in general, or other types of wars (e.g., '*nuclear_war*'), might not be considered part of the same topic as '*world_war_ii*'. This could happen because the context provided by other **keywords** might group them with unrelated terms. Of course, another valid (and perhaps even preferable for some) approach would have been to treat composite **keywords** as unique entities, creating more specific topics. However, this was the approach selected for the current work.

4. **Step 4: Creating a Term Matrix (TF)**
   Finally, a term matrix was created from the processed **keywords**. In this step, the cleaned and split **keywords**

were used to construct a document-term matrix, which represents the frequency of occurrence of each **keyword** across the dataset. The matrix was generated using a method that excluded common stop words (such as 'the', 'and', etc.), ensuring that the focus remained on meaningful **keywords**. This matrix serves as the foundation for the topic modeling task.

CountVectorizer is preferred over TF-IDF for vectorizing keywords in topic modeling because it aligns with the probabilistic nature and assumptions of models like LDA. It emphasizes raw word frequencies, which are essential for discovering latent structures in text. TF-IDF, while useful in other contexts, introduces weighted and normalized values that can distort the input and hinder the performance of topic modeling algorithms.

3.2.2 Latent Dirichlet Allocation

The next step is identifying the optimal number of topics for the **Latent Dirichlet Allocation (LDA)** model. LDA is a generative probabilistic model used for topic modeling, assuming that words in a document come from a mixture of topics, each with a specific word distribution. It uses a two-level Dirichlet process: assigning a multinomial distribution to each document for topic proportions and to each topic for word distributions. LDA uncovers hidden thematic structures by estimating the topic distribution for each document and the word distribution for each topic. The primary output consists of topics and their word distributions, and each document is assigned to a distribution of topics.

The LDA model is trained iteratively with varying numbers of topics and evaluated using the **Coherence metric**[10], which measures semantic consistency. Topics are ranked by coherence scores, and the number of topics with the highest score is chosen as optimal. In this case, the optimal number of topics is 7, with a coherence score of 0.389. The final LDA model is trained on the term matrix, identifying latent thematic structures in the movie dataset and representing each topic as a distribution of keywords. These keywords are ranked to interpret the thematic content of each topic.
The next step involves **assigning the first and second most probable topics to each movie** in the dataset, based on the topic distribution produced by the LDA model, along with the posterior probability of a topic given a word, which reflects the certainty of the model's assignment. Depending on the case, the recommendation system may consider only the first topic, both the first and second topics, or none at all, based on the confidence score and their relevance to the recommendation process.

3.3 Recommendation system

The movie recommendation system presented here integrates both content-based and topic modeling techniques to generate personalized movie suggestions.
To enhance the recommendation process, the system incorporates topic modeling by assigning each movie a primary and secondary topic, along with their associated certainties. These certainties are used to categorize the movies into different levels of relevance to the identified topics.
The recommendation process relies on cosine similarity, which measures the similarity between the input movie's overview and all other movies in the dataset. To improve the accuracy of the recommendations, the system applies a topic-based weighting mechanism, adjusting the similarity scores according to the relevance of the movie's assigned topics. This allows the system to prioritize movies that not only share similar content but are also thematically aligned with the input movie.

Topic	Top Keywords
Topic 1	sports, documentary, arts, martial, age, biography, coming, social, music, cinema
Topic 2	war, world, school, new, city, police, ii, york, high, prison
Topic 3	comedy, murder, musical, stand, movie, horror, killer, drug, revenge, serial
Topic 4	relationship, family, love, child, mother, father, marriage, friendship, daughter, death
Topic 5	woman, director, gay, lgbt, christmas, theme, wrestling, opera, alien, dance
Topic 6	film, short, concert, silent, music, rock, video, animation, experimental, cartoon
Topic 7	based, book, novel, story, softcore, true, pink, anime, lost, philippines

Table 6. Identified topics and their top keywords

Title	Topic	Certainty	Topic 2	Certainty 2
Inception	1	0.482	2	0.354
Interstellar	3	0.494	5	0.259
The Dark Knight	2	0.463	1	0.304
Avatar	1	0.254	4	0.223
The Avengers	5	0.428	1	0.330

Table 7. First 5 movies: topics and certainty scores

Topic Selection Rule: The system uses the following rules to determine which topics to prioritize for each movie recommendation, based on the certainties associated with the primary and secondary topics:

Quartiles for Certainty:

- **QF:** Calculated from the certainty values (25th, 50th, and 75th percentiles).
- **QS:** Calculated from the second_certainty values (25th, 50th, and 75th percentiles).

Decision Logic for Certainty Relevance:

- **High ($QF[0.75] \leq$ certainty):** If the certainty of the first topic is high, the system selects only the first topic as dominant ("first_only").

- **Moderate** ($QF[0.25] \leq$ **certainty** $< QF[0.75]$)**:**
  - If the second_certainty is high (QS[0.75]): Both topics are considered ("both").
  - If the second_certainty is low: Only the first topic is considered moderately ("first_only_moderate").
- **Low (certainty** $< QF[0.25]$)**:** No topic is selected ("none").

Given an input movie *A*, and considering a random movie *B* (different from A), if the topic of movie A is deemed significant and the topic of movie B has a certainty above the first quartile and is the same of movie A, then the **cosine similarity** of B is **increased by a multiplier** that accounts for the level of certainty of B. To generalize this rule, the recommendation system employs a **binary variable**, which equals 1 only if the topic of a given movie matches the topic of the input movie (A). The higher the **certainty**, the greater the boost to the cosine similarity.
If both the first and second topics are relevant for a movie, the cosine similarity is further enhanced by combining the relevance (**similarity**) and confidence (**certainty**) of both topics.

Finally, the system returns the top 3 recommended movies, ensuring that the suggestions are both content-relevant and contextually appropriate based on the user's input movie.

3.4 Comparing Recommendations for *Insurgent*

In this section, the results of movie recommendations are compared based on two different approaches: one that considers only the movie overviews and another that incorporates both the overviews and the keywords. This comparison will help determine whether the inclusion of the topics improves the recommendation process.

The case of the film *Insurgent*, assigned to topic 6, will be used as an example for this comparison.

3.4.1 Recommendations Without Topic Consideration

Using the dataset where only overview similarity was considered, the recommendations are:

Title	Topic
Antagonist	3
Geißel der Menschheit	3
Laura's Wedding	3

Table 8. Recommended Movies for *Insurgent* without topic information.

All recommended movies belong to Topic 3, which is unrelated to the input movie's topic (Topic 6). This indicates a lack of thematic alignment.

3.4.2 Recommendations With Topic Consideration

Using the dataset where topics were incorporated into the recommendation system, the results are:

Title	Topic
Allegiant	6
Babangluksa	6
The Seven Deadly Sins Part 2	6

Table 9. Recommended Movies for *Insurgent* with topic information.

All recommended movies belong to Topic 6, ensuring thematic alignment with the input movie.

3.4.3 Analysis

- **Recommendations Without Topics:**The suggestions focus on textual similarity in the overview, leading to some overlap in tone, but failing to align with the input movie's topic. Topic mismatch results in movies that might not appeal to users seeking thematic relevance.
- **Recommendations with Topics:**Incorporating topic alignment ensures that the suggested movies share a thematic connection with the input movie. For instance, *Allegiant* directly ties into the same franchise, while *The Seven Deadly Sins Part 2* and *Babangluksa* explore themes of conflict and resolution that align with the input movie's narrative style.

4. Conclusions

This work has explored techniques that complement traditional text clustering and topic modeling tasks on textual data. The results achieved in clustering demonstrate the potential for further research and applications (such as those described in Paragraph 2.8), as the use of clustering techniques has proven effective in identifying similarities among different films.

Regarding the topic modeling developed on the movies' keywords, it can be stated that, while the approach has indeed worked—evident in the recommendations generated with the aid of topic modeling—the quality of the topics obtained suggests that an alternative pre-processing strategy for the keywords might have yielded more coherent and meaningful topics.

References

[1] J. B. MacQueen, *Some Methods for Classification and Analysis of Multivariate Observations*, Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967.

[2] David M. Blei, Andrew Y. Ng, Michael I. Jordan, *Latent Dirichlet Allocation*, Journal of Machine Learning Research, 2003.

[3] Wei Cheng, Zhiqiang Wei, and others, *Hierarchical clustering algorithms for large data sets: A review*, Knowledge-Based Systems, 2020.

[4] S. K. Sharma, R. P. Verma, and others, *A Comprehensive Survey on K-means Clustering Algorithms and Their Applications*, 2019.

[5] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.

[6] L. Van der Maaten and G. Hinton, *Visualizing Data using t-SNE*, Journal of Machine Learning Research, 2008.

[7] Silveira, M. G. and Medeiros, S., *An Evaluation of Text Clustering Methods*, Journal of Data Science, 2017.

[8] Blei, D. M. and Lafferty, J. D., *A correlated topic model of Science*, 2007.

[9] H. Jelodar, Y. Wang, C. Yuan, X. Feng, X. Jiang, Y. Li, and L. Zhao, "Latent Dirichlet Allocation (LDA) and Topic modeling: models, applications, a survey," *arXiv preprint arXiv:1711.04305*, 2018. Available at: `https://doi.org/10.48550/arXiv.1711.04305`.

[10] Hamed Rahimi, Jacob Louis Hoover, David Mimno, Hubert Naacke, Camelia Constantin, and Bernd Amann. Contextualized Topic Coherence Metrics. *arXiv*, 2023. `https://arxiv.org/pdf/2305.14587`.