

A Comparative Study of K-Means, DBSCAN, and Hierarchical Clustering for Content Segmentation in Netflix Movies and TV Shows

Girish Karhana^{1,*}, Rohan², Indar Kumar³

Abstract

Over-the-top (OTT) platforms like Netflix offer a vast and diverse range of movies as well as television programs, resulting in a difficult task of content organization. The unsupervised machine learning technique of clustering is an efficient approach to organizing these OTT platforms, as this technique helps in the grouping of content on the basis of their similar semantics, i.e., without using any labeled data. The study explores the use of clustering algorithms to enhance content segmentation. This paper proposes a comparative study of three machine learning techniques of clustering, namely K-means, hierarchical clustering, as well as DBSCAN, on Netflix OTT content. Different text parameters, such as name, director, actors, categories, as well as description, of the content are transformed into numerical values using the Word2Vec model, followed by principal component analysis (PCA) dimensionality reduction. The task of evaluation of the results are evaluated using the Silhouette Score, Davies–Bouldin Index, Calinski–Harabasz Index, as well as within-cluster sum of squared errors (SSE) measures. Experimental outcomes signify that the proposed K-means technique of clustering is superior to other clustering techniques in measuring the efficiency of the formed clusters, along with the fact that the DBSCAN technique is incompetent in forming efficient clusters because of the sparse nature of high-dimensional text embeddings. Such results help to identify and highlight the strengths and limitations of each clustering algorithm for OTT content analysis.

Keywords: OTT platforms, Netflix content analysis, unsupervised learning, clustering algorithms, Word2Vec embeddings, principal component analysis, content segmentation

*Author for Correspondence

Girish Karhana
E-mail: girishkarhana@gmail.com

¹Student, School of Technology, Doon University, Dehradun, India

²Student, Department of CSE, GNIOT, Greater Noida, Uttar Pradesh, India

³Student, Department of CSE, Independent Researcher, USA

Received Date: June 24, 2026

Accepted Date: July 28, 2026

Published Date: August 15, 2026

Citation: Girish Karhana, Rohan, Indar Kumar. A Comparative Study of K-Means, DBSCAN, and Hierarchical Clustering for Content Segmentation in Netflix Movies and TV Shows. International Journal of Algorithms Design and Analysis Review. 2026; 4(2): 9–16p.

INTRODUCTION

The rapid growth of digital streaming services has transformed audience consumption of movies and television content. A growing OTT platform, such as Netflix, hosts thousands of movies and television shows across multiple genres, languages, and cultures. While this abundance increases user choice, it also makes effective organization and segmentation increasingly challenging. Efficient content segmentation is essential for improving content discovery, enhancing the user experience, and supporting personalized recommendation systems. Therefore, managing and organizing such massive volumes of content at scale has become a considerable challenge for modern streaming

platforms. Conventional content categorization techniques often depend on manually labeled data or predefined rules, which are costly to develop and difficult to maintain as content continuously evolves. Clustering is an unsupervised learning method for analyzing massive amounts of data without the need for labeled data. This makes it possible to group related data items according to the underlying patterns in the data. This study investigates the applicability of K-means, hierarchical clustering, and DBSCAN algorithms for content segmentation by applying Word2Vec embeddings to textual data, followed by dimensionality reduction using principal component analysis (PCA). Subsequently, multiple internal validation indices were employed for comparative evaluation to determine the strengths and limitations of each clustering algorithm.

LITERATURE REVIEW

Clustering techniques have been widely used in the fields of ML and content segmentation to analyze large-scale datasets. Clustering is essential for finding patterns and similarities between things and users in the context of content segmentation. The outcomes of various associated studies that have examined this are presented in Aggarwal et al. (2016) [1]. Clustering is a fundamental method for grouping similar items and users, enabling personalized content delivery. The author discusses how clustering-based segmentation improves recommendation quality when explicit labels are unavailable. Among partition-based clustering algorithms, K-means has been one of the most extensively studied methods, according to MacQueen et al. (1967) [2]. The author showed that K-means works well when the clusters are compact and well-separated. However, the algorithm was also shown to be sensitive to the choice of initial centroids and the predefined number of clusters, limiting its robustness in complex real-world datasets. The standard implementation of K-means relies on Lloyd's algorithm [3], an iterative procedure that alternates between assigning data points to the nearest centroid and updating the centroid, which is commonly taught in statistics and data science. To address the limitations of partition-based methods, density-based clustering algorithms have been introduced, with DBSCAN being one of the most prominent algorithms. According to Ester et al. (1996) [4], the DBSCAN algorithm can discover clusters of arbitrary shapes while identifying noise and outliers. The study showed that DBSCAN outperformed partition-based methods with irregular cluster-shaped noise. However, it is also noted that the algorithm's performance is highly sensitive to parameter selection and may degrade in datasets with varying density, which poses a challenge in heterogeneous content datasets. Early work by Johnson et al. (1967) [5] introduced agglomerative and divisive hierarchical clustering approaches. Introduced agglomerative and divisive hierarchical clustering approaches, highlighting their ability to reveal relationships between data points through dendrogram representation. This interpretability makes hierarchical clustering particularly useful for exploratory data analysis and understanding the cluster formation at different levels of granularity. Limited attention has been paid to conducting a unified comparative analysis of multiple clustering algorithms on real-world streaming platform datasets using consistent preprocessing and evaluation criteria. In particular, existing studies often lack a systematic examination of how algorithmic assumptions—such as cluster shape, density sensitivity, and noise handling—impact clustering outcomes in heterogeneous content metadata. This gap motivates the need for a comprehensive comparative study that evaluates K-means, DBSCAN, and hierarchical clustering on the Netflix Movies and TV Shows dataset, providing practical insights into algorithm selection for content segmentation and recommendation system applications [6].

METHODOLOGY

The systematic approach used to perform the content segmentation process on Netflix movies and television shows using unsupervised learning is described in this section. Data collection, preprocessing, text embedding, dimensionality reduction, clustering, and evaluation comprise the entire workflow of the proposed methodology (Figure 1).

Data Description

We used the Netflix Movies and TV Shows dataset, as shown in Table 1. The dataset was collected from Kaggle. There are 12 attributes in the dataset of 8807 content items, including both movies and

TV shows. These attributes offer rich textual and contextual information, making the dataset well-suited for semantic analysis and clustering [7].

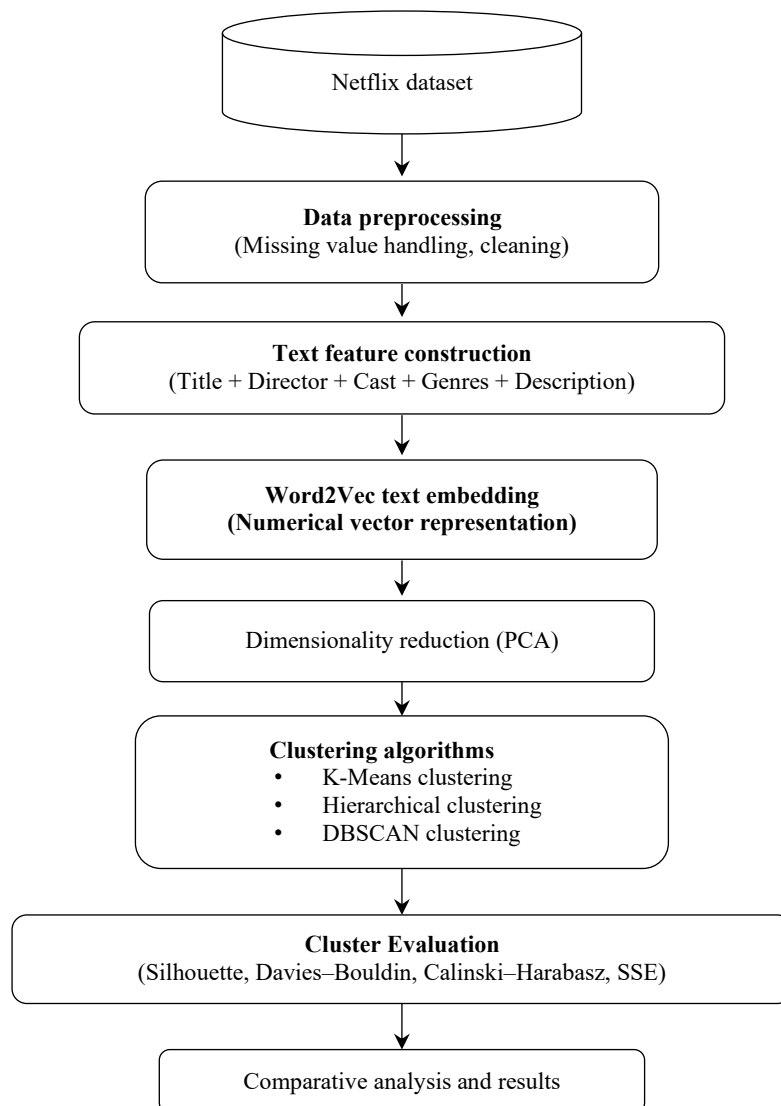


Figure 1. Methodology pipeline for Netflix content segmentation.

Table 1. Presents the detailed description of the dataset attributes used in this study.

S.N.	Attribute name
1	show_id
2	type
3	title
4	director
5	cast
6	country
7	date_added
8	release_year
9	rating
10	duration
11	listed_in
12	description

Data Preprocessing

Data preprocessing is the most important step in data analysis and programming, which involves cleaning, transforming, and organizing data into a usable format. In this study, the missing values in the director and cast attributes were replaced with suitable placeholders to maintain consistency. The categorical attributes of the related items were converted into appropriate formats, and irrelevant identifiers were removed from the dataset. Date-related attributes were standardized to ensure uniformity across the data and to improve the overall quality of the data.

Feature Engineering and Text Representation

The dataset contained unstructured textual data. Relevant textual attributes, such as the title, director, cast, categories, and description, were integrated to create a single text representation for each content item. The semantic context of every movie or television program is included in this integrated text.

The Word2Vec embedding technique was used to convert textual data into numerical form. Word2Vec learns distributed vector representations based on the context of distributed vector representations of words. Fixed-length numerical embeddings were performed by averaging the word vectors corresponding to each content item to create document-level embeddings.

Dimensionality Reduction

The high dimensionality of the Word2Vec embeddings can negatively affect the clustering performance. Therefore, PCA was employed in this study to reduce the dimensionality of the embedding space while preserving the most important information. Additionally, PCA enhances computational performance, eliminates redundancy, and facilitates the visualization of data.

Clustering

K-Means. K-means is one of the most popular clustering algorithms because of its ease of use and scalability. Jain (2010) provides an extensive review of K-means and its applications, highlighting its efficiency when applied to large datasets [8]. In this study, the number of clusters was selected using the Elbow Method, where the chosen value corresponds to the point beyond which increasing the number of clusters results in only marginal improvements in the cluster compactness.

Hierarchical Clustering

Hierarchical clustering constructs a tree-like structure of clusters, offering greater flexibility with respect to the cluster shape. This method was preferred because it produces a hierarchical representation that reveals the relationships among data points and allows analysis at multiple levels of clustering granularity [9].

DBSCAN Clustering

DBSCAN is a density-based clustering technique that recognizes outliers as noise based on their local density in the feature space and groups closely spaced data points. Unlike K-means and hierarchical clustering, it does not assume compact or spherical cluster structures; instead, it produces clusters as dense regions divided by areas of lesser density.

Evaluation Metrics

Clustering is an unsupervised task. Internal validation metrics were used to evaluate the cluster quality. The following metrics were employed:

Silhouette Score

One metric for assessing the quality of clustering is the Silhouette Score. It is computed by calculating the distance between various clusters and the cohesiveness within a cluster [6].

Davies–Bouldin Index

The Davies–Bouldin Index (DBI) is calculated as the average similarity of each cluster with the most similar cluster. A lower average similarity indicates better separation between the clusters and, consequently, better clustering performance [7].

Calinski–Harabasz Index

The Calinski–Harabasz Index is calculated by measuring the ratio of between-cluster dispersion (separation) to within-cluster dispersion (cohesion). A higher value of this index indicates better-defined and well-separated clusters.

Within-Cluster Sum of Squared Errors

The within-cluster sum of squared errors (SSE) measures the compactness of clusters and is applicable only to centroid-based clustering algorithms, such as K-means.

Comparative Analysis

The comparative analysis shows that K-means outperforms hierarchical clustering and DBSCAN across most internal validation metrics, producing compact and well-separated clusters. Hierarchical Clustering provides reasonable results but with reduced cluster cohesion. DBSCAN exhibits poor performance owing to the high-dimensional and sparse nature of text embeddings, leading to excessive noise detection. Consequently, K-means was identified as the most suitable algorithm for content segmentation in this study [10].

EXPERIMENTAL RESULT AND ANALYSIS

This section presents the experimental results obtained by applying K-means, hierarchical, and DBSCAN clustering to the Netflix Movies and TV Shows dataset. The clustering results were evaluated using several metrics, including the Silhouette Score, Davies–Bouldin Index, Calinski–Harabasz Index, and Within-Cluster SSE. Clustering visualizations are also provided to better understand the distribution of content groups.

Clustering Visualization

To visualize the cluster formation, the high-dimensional Word2Vec embeddings were converted to two dimensions using PCA. Cluster plots provide insights into how different algorithms group Netflix content.

K-Means Clustering

The K-means algorithm groups the content based on semantic similarities using textual attributes such as title, cast, director, genres, and description (Figure 2).

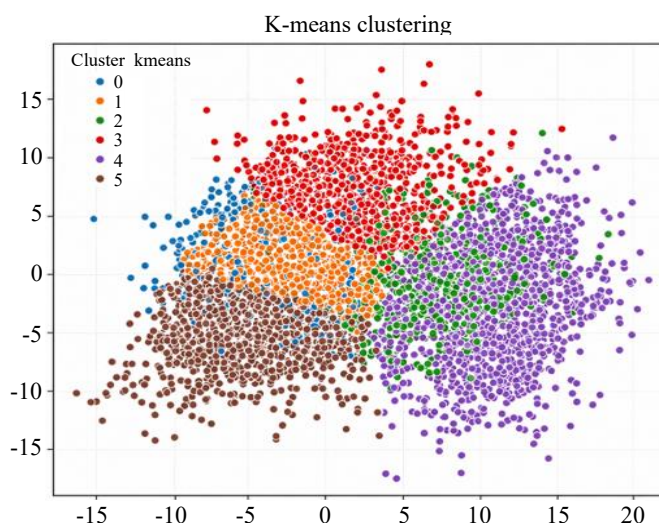


Figure 2. K-means clustering visualization of Netflix content.

Hierarchical Clustering

In hierarchical clustering, data points are organized into nested clusters, showing the hierarchical relationship between content items (Figure 3).

DBSCAN Clustering

DBSCAN is a density-based algorithm that groups closely packed points into clusters. However, in higher-dimensional text data, many points may be considered noise owing to sparse distribution, as shown in Figure 4.

Performance Evaluation

The applied clustering algorithms were evaluated using internal validation metrics that measure cluster compactness and separation (Table 2).

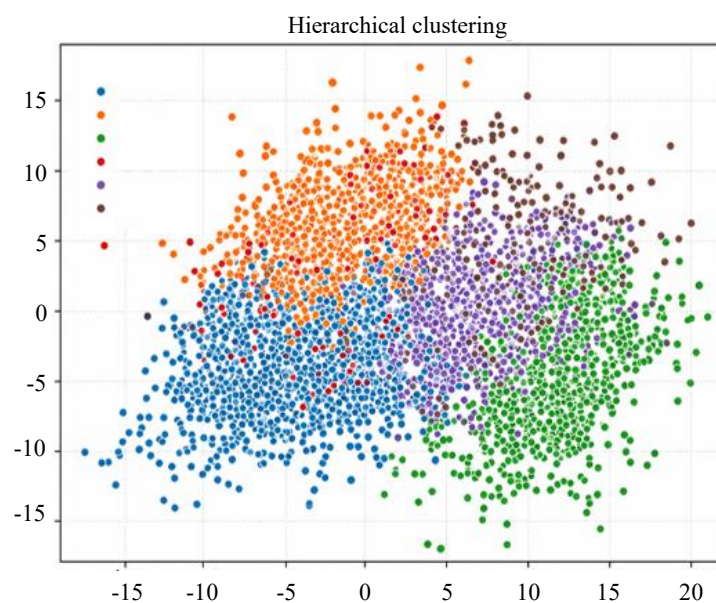


Figure 3. Hierarchical clustering visualization of the Netflix dataset.

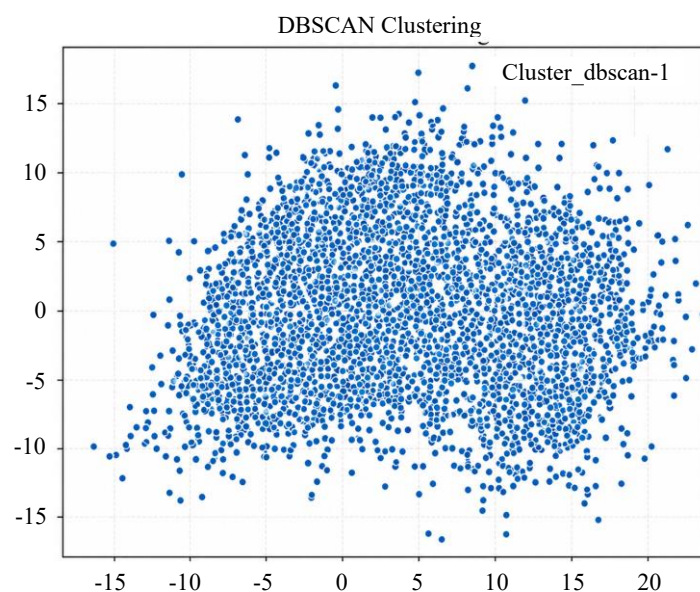


Figure 4. DBSCAN Clustering visualization highlighting dense clusters and noise points.

Table 2. Performance comparison of clustering algorithms.

Algorithm	Silhouette score	Davies–Bouldin index	Calinski–Harabasz index	Within-cluster SSE
K-Means	0.188383	1.530945	2378.566650	371068.0625
Hierarchical	0.176870	1.596520	2025.949463	NaN
DBSCAN	NaN	NaN	NaN	NaN

Comparative Analysis of Clustering Algorithms

K-means performs well when the data are structured and produces distinct clusters; however, we must predetermine the number of clusters according to a comparative analysis of the three clustering algorithms. DBSCAN is effective at identifying noise and outliers and does not require the number of clusters beforehand; however, its performance depends on choosing the right parameters, and it may struggle when the data density is uneven. Hierarchical Clustering helps in better understanding the relationships between data points through its tree-like structure; however, it is slower and not suitable for large datasets. Overall, no single algorithm is the best in every case, and the choice depends on the type of data and the problem being solved.

CONCLUSION

In this study, we compared three clustering algorithms—K-means, DBSCAN, and hierarchical clustering—using the Netflix Movies and TV Shows dataset. The aim was to understand how each algorithm performed on real-world content data. The results indicate that K-means performs well when the data are structured and produce clear clusters; however, it requires the number of clusters to be defined in advance. DBSCAN is useful for detecting noise and outliers and does not require a predefined number of clusters, although its performance depends on parameter selection. Hierarchical Clustering helps in understanding the relationships between data points but is slower and less suitable for large datasets. Overall, no single algorithm is universally optimal for all situations, and the choice of algorithm depends on the nature of the dataset and problem requirements. This study emphasizes the importance of using the appropriate clustering technique for efficient content segmentation and recommendation systems.

Disclosure of Interests

The authors declare no competing interests relevant to the content of this study.

REFERENCES

1. Aggarwal CC. Recommender Systems. Cham: Springer International Publishing; 2016. doi:10.1007/978-3-319-29659-3.
2. MacQueen J. Some methods for classification and analysis of multivariate observations. In: Le Cam LM, Neyman J, editors. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Vol. 1. Berkeley (CA): University of California Press; 1967. p. 281-297.
3. Lloyd S. Least squares quantization in PCM. IEEE Trans Inf Theory. 1982;28:129-137. doi:10.1109/TIT.1982.1056489.
4. Ester M, Kriegel HP, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. 1996. p. 226-231.
5. Johnson SC. Hierarchical clustering schemes. Psychometrika. 1967;32:241-254. doi:10.1007/BF02289588.
6. Rousseeuw PJ. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math. 1987;20:53-65. doi:10.1016/0377-0427(87)90125-7.
7. Davies DL, Bouldin DW. A cluster separation measure. IEEE Trans Pattern Anal Mach Intell. 1979;PAMI-1:224-227. doi:10.1109/TPAMI.1979.4766909.
8. Jain AK. Data clustering: 50 years beyond K-means. Pattern Recognit Lett. 2010;31:651-666. doi:10.1016/j.patrec.2009.09.011.

9. Zaïane OR, Foss A, Lee CH, Wang W. On data clustering analysis: Scalability, constraints, and validation. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Berlin: Springer; 2002. p. 28-39.
10. Han J, Kamber M, Pei J. Data Mining: Concepts and Techniques. Waltham (MA): Morgan Kaufmann Publishers; 2012.