Cluster Analysis, ANN and Text Mining

The provided dataset was the imdb_dataset.csv

Cluster Analysis:

This report focuses on applying K-means clustering and hierarchical clustering to the imdb dataset, aiming to discover the patterns among movie attributes such as runtime, rating, and audience score.

K-Mean Clustering

K-Means organizes data into groups where items in the same group are more similar to each other. We used:

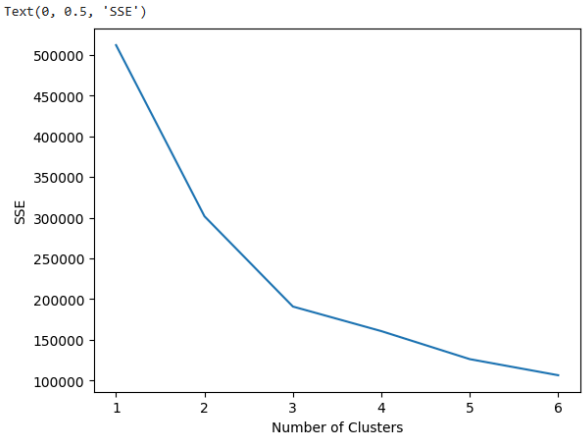
- Feature Selection: where we selected the numeric features ‘runtime’, ‘imdb_rating’, and ‘audience_score’ for clustering.
    - Non-numeric attributes were excluded (‘title’ and ‘genre’)
    - We applied k-mean with k=2 clusters
- Results:
    - The dataset was divided into two clusters, indicating movies with:
        - Higher audience scores and imdb ratings (cluster 0)
        - Lower audience scores and shorter runtimes (cluster 1)

The centroids of the clusters indicate that Cluster 0 contains movies with certain feature similarities (such as longer runtimes or higher audience scores), while Cluster 1 contains movies with different characteristics. The centroids of these clusters represent typical feature values for each group and can be applied to other movies to determine their cluster assignments based on similar attributes.

	runtime	imdb_rating	audience_score
0	90.5	6.4	77.0
1	139.0	7.2	76.0

Determining the number of clusters in the data:

To determine the number of clusters in the data, we applied k-means with varying numbers of clusters from 1 to 6 and computed their corresponding sum-of-squared errors (SSE). The "elbow" in the plot of SSE versus number of clusters was used to estimate the number of clusters.

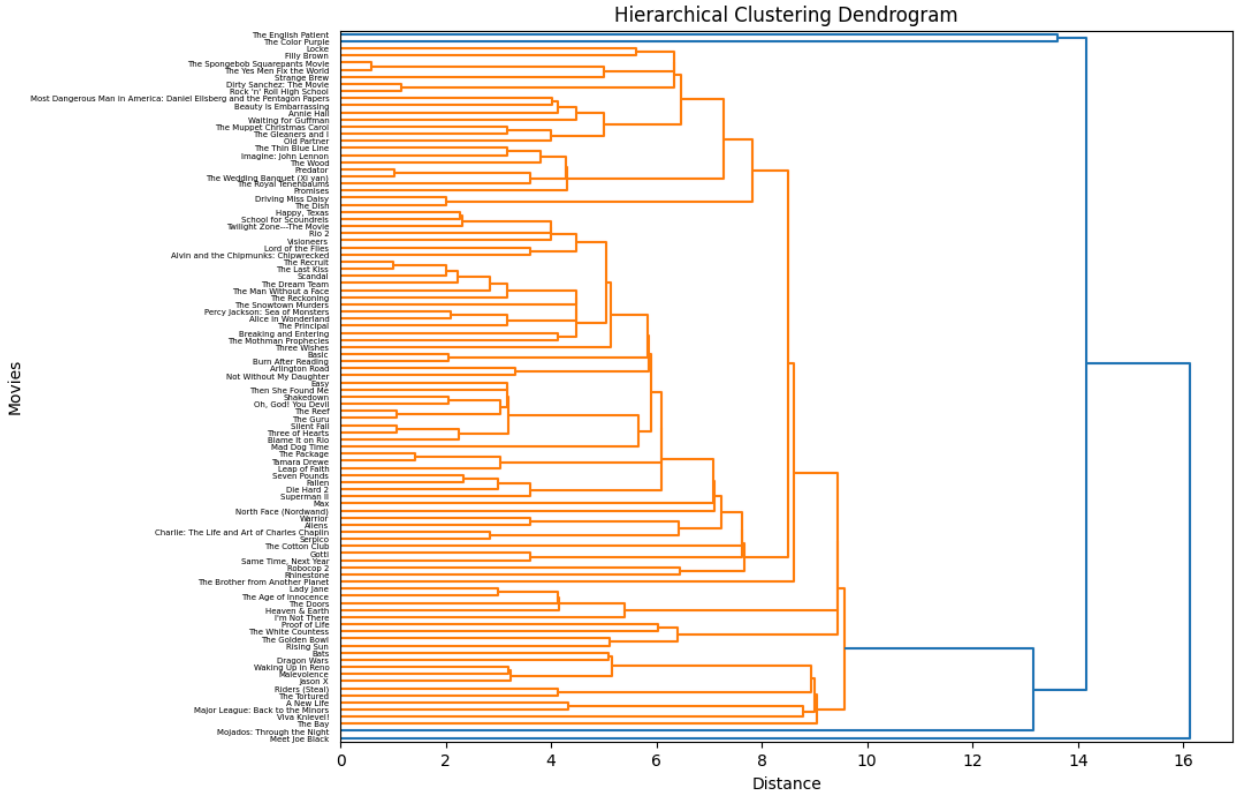


The SSE plot showed an "elbow" at k=2, confirming this as the ideal number of clusters

Hierarchical Clustering

Hierarchical clustering builds a hierarchy of clusters using a distance matrix. We used the following linkage methods: single linkage (MIN), complete linkage (MAX), and group average. We included:

- Feature Selection: where we used the same numeric features as k-means: ‘runtime’, ‘imdb_rating’, and ‘audience_score’.
- Dendrograms to visualize the clustering process.
    - Each dendrogram revealed clusters of movies grouped by similarity.

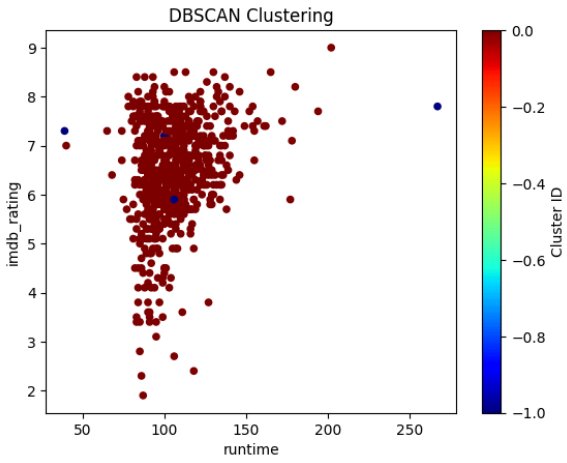


To visually display an example, this image provides a single-link clustering dendrogram. This dendrogram groups movies based on how similar they are. The closer the branches are (horizontally), the more alike the movies in those clusters.

Density-Based Clustering

Density based clustering is a method of grouping data points into clusters by identifying dense regions in the dataset and separating them from areas of lower density. In our case, we grouped movies based on their 'runtime' and 'imdb_rating'.

- Progress: We apply the DBScan clustering algorithm on the data by setting the neighborhood radius (eps) to 15.5 and minimum number of points (min_samples) to be 5. The clusters are assigned to IDs between 0 to 8 while the noise points are assigned to a cluster ID equal to -1.
- Results:
    - The scatterplot identified those noise points with a cluster ID of -1, representing outliers, movies that do not fit within our cluster.



ANN (Artificial Neural Network):

- We implemented the ANN by splitting the dataset into training and testing sets using the train_test_split method. This ensured that the training and testing sets were completely separate. Every time the process is run, a different combination of training and testing sets will be used.
- The target variable for this project was the audience_rating column, which originally contained categorical values: Upright and Spilled. These values were converted into numerical formats (binary) to ensure they would be executed in ANN and other classification models such in the Logistic Regression Model, and the Decision Tree Classifier.

Comparison with Other Classification Models

- On average the ANN accuracy score was around 0.63 which means that ANN correctly predicted audience_rating 63% of the time of the test data.
- The logistic Regression Accuracy score was around 0.67 average which means that the Logistic Regression model correctly predicted 'audience_rating' 67% of the time of the test data. It suggests that for this dataset Logistic Regression Accuracy predicts it a bit better.

- The Decision Tree Classifier correctly predicted the target variable for 65% of the test data.
- Overall the highest accuracy is the Logistic Regression Model with ~67%, the Decision Tree Classifier with ~65% and last is ANN with ~63%

Attribute Value Binarization

Encoding categorical features

- The encode_text_dummy was used to encode categorical columns. This ensures that categorical data is appropriately formatted for ANN.

Converting the target variable transformation

- The 'audience_rating' column was mapped into binary values in order to align with a classification structure.
  - 1 for "Upright"
  - 0 for "Spilled

	audience_rating
1	Upright
2	Upright
3	Upright
4	Upright
5	Spilled
...	...
647	Spilled

*** BONUS * Data processing for ANN**

Feature Selection

- We selected a few relevant columns to focus on 'genre', 'runtime', 'mpaa_rating', 'studio', 'thtr_rel_year', 'imdb_rating', 'audience_rating'

Removing the missing values

- Rows with missing values were dropped, ensuring data consistency and preventing training biases.

Data splitting

- The variables X and Y matrices were generated using the to_xy function

Outcome/Result

- The target variable Y was converted to align with the output layer of the ANN

Text Mining

Text mining is essential when it comes to data analysis with textual data. Text processing involves converting raw text into a structural format for computational analysis. For the data provided, we tested out multiple text processing techniques:

CountVecorizer:

- Transforms text into a sparse matrix where each row corresponds to a document (sentence), and each column corresponds to a unique word.

```
[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0]
[0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1
 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0
 1 0 0 0 0 0 1 0 1 0 0 1 0 0 1 0 0 2 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0
 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0]]
['admire' 'afford' 'agreed' 'allowance' 'am' 'an' 'and' 'announcing' 'as'
 'believe' 'company' 'comparison' 'contented' 'continue' 'conveying' 'day'
 'declared' 'described' 'devonshire' 'did' 'direction' 'dissimilar' 'do'
 'easy' 'excellent' 'excuse' 'exercise' 'explained' 'for' 'formed'
 'former' 'garden' 'had' 'has' 'hastened' 'he' 'head' 'hearts' 'humanity'
 'in' 'is' 'its' 'just' 'least' 'literature' 'manners' 'men' 'most' 'my'
 'necessary' 'no' 'nor' 'now' 'of' 'oh' 'on' 'otherwise' 'out' 'parish'
 'parlors' 'party' 'perfectly' 'principle' 'property' 'put' 'relied'
 'resolving' 'result' 'room' 'say' 'set' 'she' 'simple' 'sir' 'sister'
 'so' 'such' 'supplied' 'supposing' 'suspected' 'sweetness' 'terminated'
 'therefore' 'to' 'travelling' 'uncommonly' 'use' 'vicinity' 'warmth'
 'while' 'who' 'wrote' 'yet' 'you']
```

As shown, the values in the matrix represent the frequency of each word in a document.

TfIdfVectorizer:

- Similarly, TfidfVectorizer transforms text into a sparse matrix where rows are text and columns are words, and values are the tf-dif values

Although this section focuses on a different dataset, the principles of text processing are crucial for preparing textual data, since it enables the transformation of unstructured text into usable data for modeling or other applications.

Team Contributions:

Both team members, Marilyn Sarabia Ortiz & Isabel Santoyo-Garcia, contributed equally to all aspects of this project, including data preprocessing, model development, analysis, and documentation.