Data Mining: Fundamentals

Group 12

Bruno Barbieri, Noemi Dalmasso, Gaia Federica Francesca Ferrara

The aim of this report is to display an analysis carried out on the IMDb dataset; the analysis has been conducted making use of data mining methodologies. After the data understanding and preparation phase, clustering, classification, and pattern mining techniques have been applied.

Contents

1. Data Understanding and Preparation

1.1 Data Semantics

The dataset *train.csv* contains 16431 titles of different forms of visual entertainment that have been rated on IMDb, an online database of information related to films, television series etc. Each record is described by 23 attributes, both numerical and non-numerical. All the variables of the dataset are introduced and explained in Table 1.1 and Table 1.2.

Attribute	**Type**	**Description**
originalTitle	Nominal	Title in its original language
rating	Ordinal	IMDB title rating class The range is from (0,1] to (9,10]
worstRating	Ordinal	Worst title rating
bestRating	Ordinal	Best title rating
titleType	Nominal	The format of the title
canHaveEpisodes	Nominal (Binary)	Whether or not the title can have episodes True: can have episodes; False: cannot have episodes
isRatable	Nominal (Binary)	Whether or not the title can be rated by users True: it can be rated; False: cannot be rated
isAdult	Nominal (Binary)	Whether or not the title is for adults 0: non-adult title; 1: adult title
countryOfOrigin	Nominal	The country(ies) where the title was produced
genres	Nominal	The genre(s) associated with the title

Table 1.1: Description of non-numerical attributes

1.2 Distribution of the variables and statistics

This section will give an overview about the distribution of variables that has been carried on to understand patterns, detect meaningful statistics and assess their relevance to the project.

1.2.1 Discrete attributes

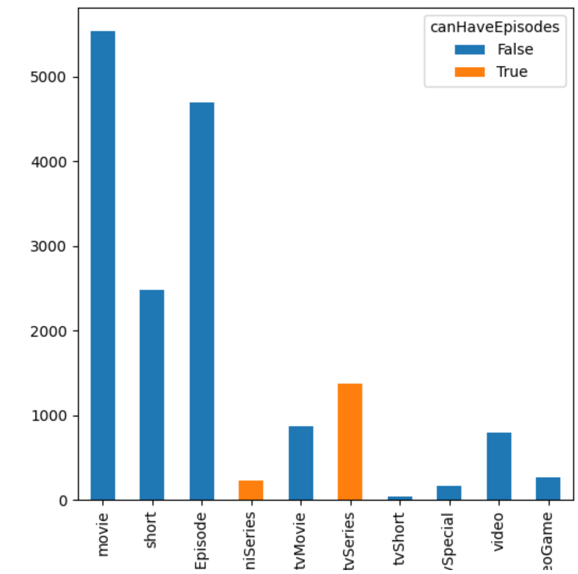
DA CAMBIARE IN BASE AI GRAFICI CHE DECIDIAMO DI TENERE In this paragraph, the most informative discrete attributes of the dataset are examined to provide an overview of their statistics and frequencies.

From Fig.1.1(a), it is observed that the classes of the `titleType` attribute are unbalanced, with *movie*

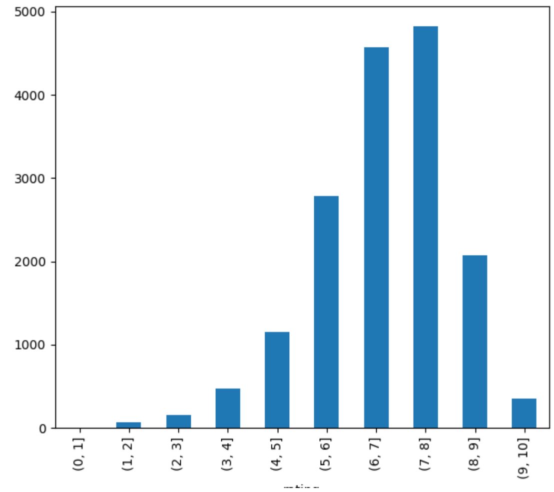
Attribute	**Type**	**Description**
runtimeMinutes	Numeric	Runtime of the title expressed in minutes
startYear	Interval	Release/start year of a title
endYear	Interval	TV Series end year
awardWins	Numeric	Number of awards the title won
numVotes	Numeric	Number of votes the title has received
totalImages	Numeric	Number of Images on the IMDb title page
totalVideos	Numeric	Number of Videos on the IMDb title page
totalCredits	Numeric	Number of Credits for the title
criticReviewsTotal	Numeric	Total Number of Critic Reviews
awardNominationsExcludeWins	Numeric	Number of award nominations excluding wins
numRegions	Numeric	The regions number for this version of the title
userReviewsTotal	Numeric	Number of User Reviews
ratingCount	Numeric	The total number of user ratings for the title

Table 1.2: Description of numerical attributes

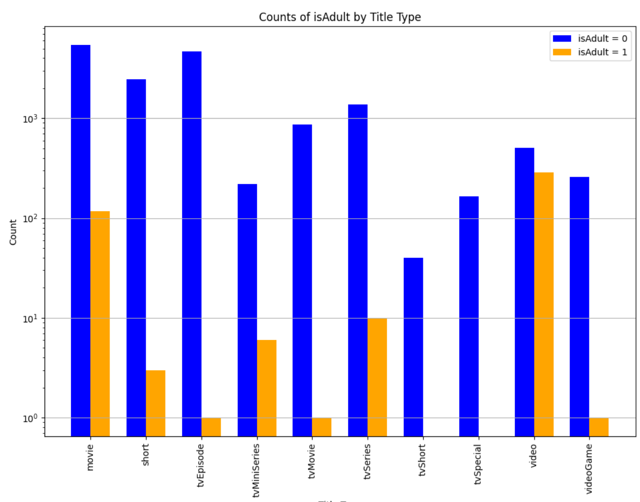
being the most frequent class (5535 records) and *tvShort* the least frequent (40 records). By analyzing the `canHaveEpisodes` attribute within these `titleType` values, it is found that only *tvSeries* and *tvMiniSeries* can have episodes, as expected. As shown in Fig.1.1(b), the frequency of rating classes is slighly skewed toward higher values, with the most frequent rating class being (7, 8], which is the rating of 4822 titles. Another important aspect is that all 16341 titles are ratable and the vast majority of them (16005) are non-adults contents, as shown in Fig.1.1(c) Finally, as indicated in Fig.1.1(d), an analysis of the `genres` variable across different `titleType` values reveals that *Drama* and *Comedy* are the most common genres, as they appear in the top 3 genres of nearly every *titleType* category.



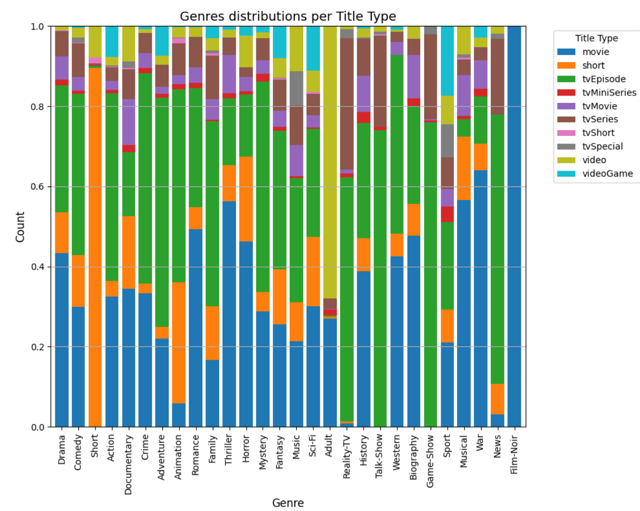
(a) Counting of the title types frequencies



(b) Counting of ratings frequencies



(c) Counting of the adult and non-adult per type



(d) types per genre

Figure 1.1: Bar chart of the discrete attributes.

1.2.2 Continuous attributes

...content...content...content...content...content...content...

1.3 Data Quality

In this phase, a proper evaluation of the observed data was conducted in preparation for the analysis. Once having checked that there are no duplicates and no incomplete rows in the dataset, attention was given at identifying missing values and outliers within the columns.

1.3.1 Syntactic Inconsistencies

In the exploration of the dataset it has been noticed that `awardWins` was the only feature having missing values identified with NaN. However, there were missing values also in other columns (`endYear`, `runtimeMinutes` and `genres`), but they were indicated with the string "\N". To avoid this inconsistency causing problems during data preparation, these values have been replaced with NaN. By doing so, any cell in the `endYear`, `runtimeMinutes` and `genres` column that previously contained the string "\N" is now considered a proper missing value, detectable and manageable using Pandas' functions.

1.3.2 Missing Values

Once having solved the above-mentioned inconsistency, the resulting total amount of the missing values are the following, also represented in percentages for a better understanding:

- `endYear`: it is the feature with the highest number of NaN values (15617; about 95%). To handle them it has been decided to **COSA FARE?**;
- `runtimeMinutes`: it has 4852 missing values (29.5%) that have been handled by grouping the records by `titleType` and substituting the NaN value with the median of each group;
- `awardWins`: this feature has 2618 NaN values (about 16%). Since the mode associated with this variable is 0, it has been decided to substitute the missing values with 0;
- `genres`: it is 382 missing values (2.3%). Having dealt this variable with a multi-label one-hot encoding process (as will be described in the *Variable Transformation* section), a vector of all zeros is assigned to record with missing genres values.

1.3.3 Outliers detection

While examining the dataset, it became apparent that some attributes have outliers. The important aspect to highlight is that since `awardWins`, `totalVideos` and `awardNominationsExcludeWins` have many values as 0 (respectively 89%, 14821, 14427), these might be considered variables with many outliers (as seen in Figure **METTERE GRAFICO che rappresenti in qualche modo il result di DETECT_OULIERS_MULTI_ATTRIBUTES in data_quality noemi**) but they actually have less outliers compared to the other variables. For the other attributes **CONTINUARE...... VALORI OUTLIERS SU TRAIN IN %: 86.7 , 90.2 , 87.8 VALORI OUTLIERS SU DF_PP IN %: 88.7 , 90.2 , 87.8**

1.4 Variable Transformation

As the first step in the variable transformation process, the `CountryOfOrigin` and `genres` variables (datatypes: strings) were converted into lists of strings to facilitate further analysis. This transformation was necessary because some records contain multiple genres or countries as values for these variables. Multi-label one-hot encoding was applied to the `genres` column; each unique genre was represented as a binary feature, allowing records that belong to multiple genres simultaneously to maintain this information. Furthermore, it has been decided to extract the ceiling value for each entry in the `rating` column, in order to use it as an integer for further analysis.

For the numeric attributes, it was observed that some variables required a stronger transformation due to their highly positively skewed distributions. Specifically, when required by the data mining method, a log-transformation was applied to all the numeric attributes, since their skewness was highly greater than 1. **DECIDERE SE SPECIFICARE EFFETTIVAMENTE IN QUALE TECHNIQUE L'ABBIAMO APPLICATA O SE RIMANERE VAGHI**. Following the log-transformation, standard normalization techniques - `MinMaxScaler` and `StandardScaler` - have then been applied (when scaling was necessary); respectively to scale each feature to a given range and to standardize features by removing the mean and scaling to unit variance. The decision to apply one or the other was again made based on the specific requirements of each data mining technique.

1.5 Pairwise correlations and elimination of variables

The plot in figure 1.2 is a Pearson's correlation matrix that takes into account the continuous numerical variables of the dataset. For what can be observed, `numVotes` and `ratingCount` have a perfect positive correlation, and so it would be redundant to keep them both. For this reason it has been decided to drop `ratingCount`. On the other hand, regarding categorial attributes, after having analized them, `bestRating`, `worstRating`, and `isRatable` have been discarded because they **endyear(???)** were found to have limited contribution based on their distributions. As a matter of fact, their unique values were respectively 10, 1 and True for all attributes.

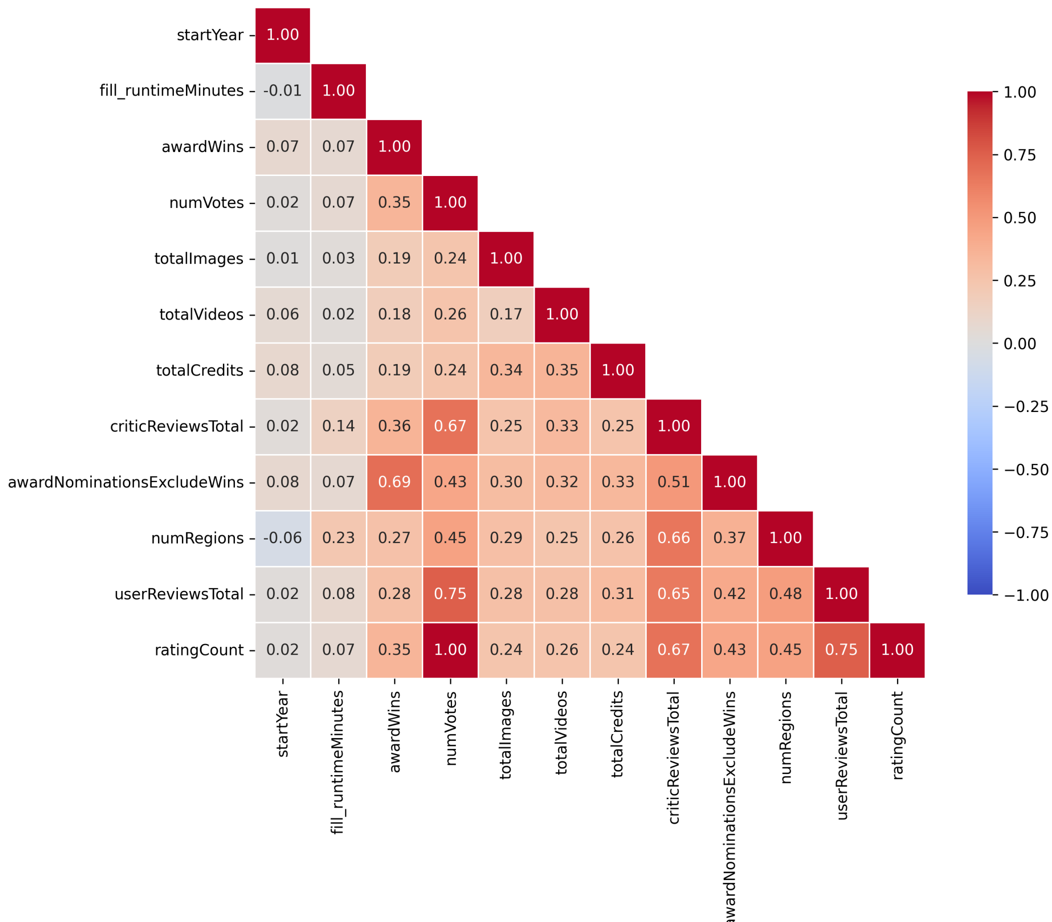


Figure 1.2: Correlation matrix

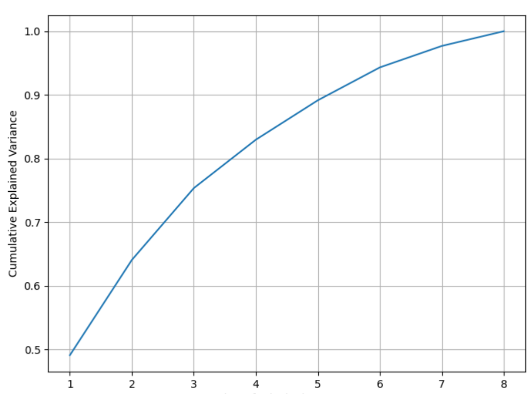
2. Clustering

This chapter of the report aims at illustrating the clustering analysis performed on the dataset at hand. The employed clustering techniques are K-means (Centroid-based), DBSCAN (density-based) and hierarchical clustering.

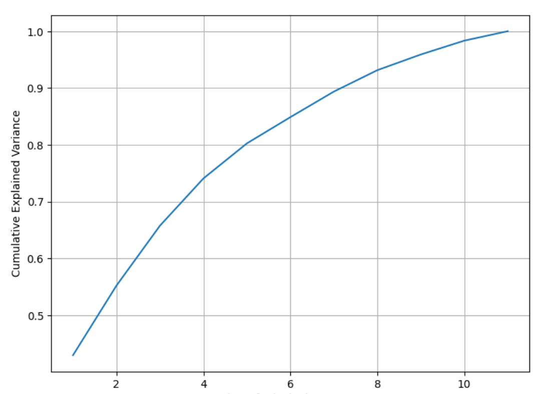
The analysis conducted using these methods focused only exclusively on the dataset's numerical attributes, which were appropriately log-transformed (as mentioned in the *Variable Transformation* section) and normalized using `StandardScaler`. For the K-means algorithm, `awardWins`, `awardNominationsExcludeWins`, and `totalCredits` were excluded due to their high proportion of zero values, which negatively affected cluster formation.

In addition, an attempt was made to incorporate categorical variables to the analysis with the K-means algorithm by converting them into binary attributes and constructing a mixed-distances matrix. Distances were then calculated using the Euclidean distance for numerical (log-tranformed and scaled) features and the Jaccard similarity for binary ones. However, this approach did not lead to any improvement in the results.

Principal Component Analysis (PCA) was applied to the preprocessed data to effectively visualize the clustering results. Analysis of the numerical attributes reveals that 4 principal components are optimal when excluding variables with many zero values, while 5 components are needed when including all variables. These numbers of components capture the maximum meaningful variance, as shown by the point in the plots where where the line starts to flatten, indicating that adding more components doesn't increase explained variance significantly. The differences between these two approaches are illustrated in Figure X. The plots in figure 2.1 show the differences between these two approaches.



(a) PCA excluding variables



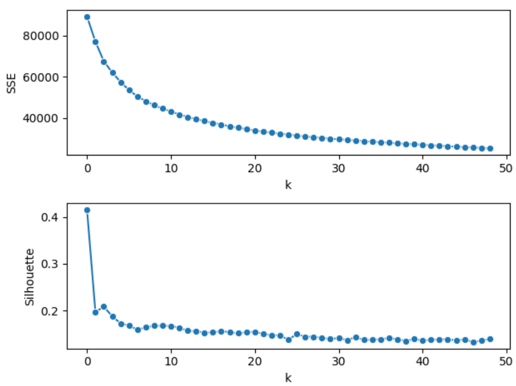
(b) PCA with all variables

Figure 2.1: Principal Component Analysis

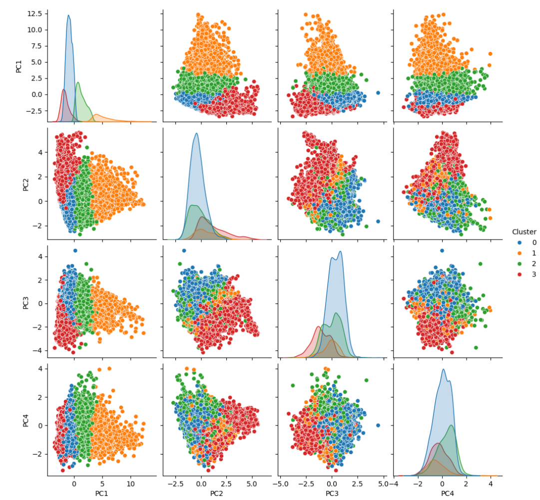
2.1 K-means

To identify the optimal number of clusters, both the SSE and Silhouette scores were computed. The goal was to find a configuration that minimizes the SSE while maintaining a robust Silhouette score. The plots in figure 2.2a demonstrate that $k = 4$ provides the optimal balance between these metrics. Choosing $k = 4$ returns a SSE score of 67496 and Silhouette score of 0.21.

The cluster results are presented in figure 2.2b.



(a) SSE and Silhouette scores



(b) K-means Visualization

Figure 2.2: K-means clustering analysis

The distribution of data points across the four clusters is as follows (shown in percentage of data points per cluster): Red (0): 51.68%, Blue (1): 7.42%, Green (2): 24.22%, Orange (3): 16.68%. The clusters are not as well-separated in most Principal Component combinations as they are with PC1. In fact, in the other combinations, the clusters tend to overlap and their boundaries are not always clearly distinct. This might be an indication that the true clusters have irregular shapes or different densities, resulting in boundaries between them being not clearly defined.

2.2 DBSCAN

To determine the optimal DBSCAN parameters, the k^{th} *nearest neighbors* method was used: this allows to identify *eps* (the maximum distance between two points for them to be considered neighbors) given the value of *Minpts* (minimum number of points in a neighborhood for a point to be considered a core point). Initially, *Minpts* was set to 22, following the rule of setting it above twice the number of dimensions. However, due to the dataset's unbalanced nature and the sparsity of high-dimensional data, reducing *Minpts* to 11 allowed the formation of smaller clusters while preventing the risk of detecting only one dominant cluster and classifying many minority groups as noise instead of distinct clusters. To determine *eps*, the k^{th} nearest neighbors plot (with $k = 11$) was analyzed. While the "knee" point suggested an eps of around 0.1, this value would have resulted in excessive noise and a single dominant cluster. To address this, eps was set to 1.564, allowing for meaningful connectivity while preserving the detection of smaller clusters without merging them into a single entity. By adjusting *eps* and *Minpts* appropriately, the clustering results achieved a Silhouette score of 0.139, indicating little improved cluster separation and reduced noise, which is considered good enough for an unbalanced, high-dimensional dataset.

2.3 Analysis by hierarchical clustering

2.4 General considerations

3. Classification

4. Regression

5. Pattern Mining