Data Mining II Project:

Analyzing Data Insights from the IMDb Platform

Authors:

Bruno Barbieri, Chiara Ferrara, Ankit Kumar Bhagat

Academic Year 2024/2025

Contents

Introduction

The goal of this report is to illustrate the caracteristics of the given IMDb dataset, and to show key insights that can be obtained from it. In particular, the focus of many of the observations is based on aspects that could be useful for a product's creation and marketing, in order to optimize the chances of success of a product in the market.

1 Data Understanding and Preparation

TODO: distrib graphs The dataset contains around 1.6 million of titles of different types. For each title, the dataset contains information regarding many different aspects. Table 1 lists the initial categorical features.

Feature	**Description**
`originalTitle`	Original title, in the original language (?)
`isAdult`	Whether or not the title is for adult
`canHaveEpisodes`	Whether the title can have episodes
`isRatable`	Whether the title can be rated by users
`titleType`	Type of the title (e.g., movie, tvseries)
`countryOfOrigin`	Countries where the title was primarily produced
`genres`	Genres associated with the title
`regions`	Regions for this version of the title
`soundMixes`	Technical specification of sound mixes
`worstRating` (ordinal)	Worst title rating
`bestRating` (ordinal)	Best title rating
`rating` (ordinal)	IMDB title rating class

Table 1: Initial categorical features of the IMDb dataset

Of the initial categorical attributes, the following were removed:

- `originalTitle`, as it did not provide particularly useful information;
- `isAdult`, as it was almost completely correlated with the *Adult* genre, so a logical OR operation was performed, and the genre only was kept; Interesting to note the fact that in our representation, being that the genres are represented through freq enc, we don't have the info
- `canHaveEpisodes`, as it was completely correlated with the title type being *tvSeries* or *tvMiniSeries*;
- `isRatable`, as it was always true;
- `worstRating` and `bestRating`, as they were always 1 and 10, respectively;
- `rating`, as it was obtainable from the `averageRating` continuous attribute, through a simple discretization.

`soundMixes` was also removed, as it required some domain knowledge to be understood, as well as having issues with the values it contained.
Because of their very similar meaning, `regions` and `countryOfOrigin` were merged through a simple union operation. The resulting feature was then represented trhough frequency encoding on the entire list, as well as counts of the number of countries from each continent. This resulted in eight new features (six continents, one for unknown country codes, and the last for the frequency encoding).
While inspecting the `genre` attribute, it was observed that each record contained up to three genres, listed in alphabetical order—indicating that the order did not convey any semantic information about the title. To represent this information, three separate features were created, each corresponding to one of the genres. These features were encoded using frequency encoding, sorted in descending order of frequency across the dataset. A value of 0 was used to indicate missing genres—either when no genres were present or to fill the remaining slots when fewer than three were available.

The initial numerical features are listed in Table 2.
`endYear` was removed due to it not being meaningful for non-Series titles, and having around 50% of missing values for *tvSeries* and *tvMiniSeries*.
`totalImages`, `totalVideos` and `quotesTotal` were merged through a simple sum operation into a single feature

Feature	Description
startYear	Release year of the title (series start year for TV)
endYear	TV Series end year
runtimeMinutes	Primary runtime of the title, in minutes
numVotes	Number of votes the title has received
numRegions	Number of regions for this version of the title
totalImages	Total number of images for the title
totalVideos	Total number of videos for the title
totalCredits	Total number of credits for the title
criticReviewsTotal	Total number of critic reviews
awardWins	Number of awards the title won
awardNominations	Number of award nominations excluding wins
ratingCount	Total number of user ratings submitted
userReviewsTotal	Total number of user reviews
castNumber	Total number of cast individuals
CompaniesNumber	Total number of companies that worked for the title
averageRating	Weighted average of all user ratings
externalLinks	Total number of external links on IMDb page
quotesTotal	Total number of quotes on IMDb page
writerCredits	Total number of writer credits
directorCredits	Total number of director credits

Table 2: Initial numerical features of the IMDb dataset

(totalMedia) because of their similar semantic meaning, as well as heavy right skewness. The same was true for awardWins and awardNominations, as well as userReviewsTotal and criticReviewsTotal, merged with the same procedure into totalNominations and reviewsTotal, respectively.
castNumber, writerCredits, directorCredits with and without totalCredits; deltacredits
runtimeMinutes had a very high number of missing values (add %). Since the feature had high relevance in the domain, it was imputed with random sampling from a interquartile range, separately for each title type.
eventually, add description of the imputation procedure for tasks which involved titleType

2 Outliers

2.1 COF

2.2 Isolation Forest

2.3 ABOD

3 Imbalanced Learning

3.1 Undersampling

3.2 Oversampling

3.2.1 SMOTE

4 Advanced Classification

In this section, classification results are showcased for two target variables: averageRating (properly binned into 5 classes), and titleType (with 6 classes).

4.1 Ensemble methods

4.2 Neural Networks

We then applied multiple classification models using the data as preprocessed ... The first target variable is titleType, which includes six categories: *movie*, *short*, *tvEpisode*, *tvSeries*, *tvSpecial*, and *video*. The classes are not equally distributed, with *tvSpecial* and *video* being significantly under-represented compared to the others.

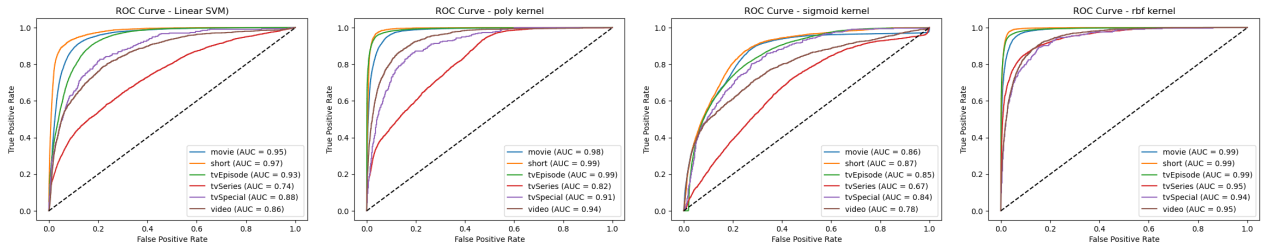
Entries labeled as *videoGame* were removed from both the training and testing sets, as they were too few to be useful for classification. The remaining categories were merged into broader groups according to the following mapping: *movie* and *tvMovie* were grouped as *movie*, *short* and *tvShort* were grouped as *short*, *tvSeries* and *tvMiniSeries* were grouped as *tvSeries*, while *tvEpisode*, *tvSpecial* and *video* were left unchanged.
All feature columns were standardized using a `StandardScaler`.
In addition, the variable `canHaveEpisodes` was removed prior to training, since it provides direct information about the target `titleType` and could therefore introduce data leakage.

4.3 Support Vector Machines

Support Vector Machines (SVM) were applied to the IMDb classification task to distinguish among six title categories: *movie*, *short*, *tvEpisode*, *tvSeries*, *tvSpecial*, and *video*. Both linear and non-linear kernels were investigated to assess how the flexibility of the decision boundary affects classification performance.
The first experiment used a Linear SVM trained on the full dataset. A grid search with five-fold cross validation was carried out on the parameters $C \in \{0.01, 0.1, 1, 10, 100\}$ and $max\_iter \in \{1000, 5000, 10000\}$. The optimal configuration, with $C = 100$ and $max\_iter = 1000$, achieved a test accuracy of 0.81. While precision and recall were high for majority classes (*movie*, *short*, *tvEpisode*), the classifier failed on *tvSeries*, *tvSpecial*, and *video*, indicating that a linear decision boundary is insufficient for this problem.
Non-linear kernels were then evaluated. Because a full parameter search on the complete training set would be computationally expensive, an exploratory grid search was performed on a stratified 10% sample of the training data. RBF and polynomial kernels performed well, while the sigmoid kernel consistently underperformed and was discarded. The selected configurations were then retrained on the full dataset. Both RBF and polynomial kernels achieved approximately 0.90 test accuracy, substantially outperforming the linear model.
ROC curves were used to evaluate class separability (Figure 1), showing excellent separation for majority classes, although minority categories remained problematic. Confusion matrices (Figure 2) illustrate that *tvSpecial* and *video* were frequently misclassified. I will change the text and explain the figures better.
To address class imbalance, the RBF kernel was retrained with `class_weight=balanced`, which penalizes misclassification of under-represented classes. This model reached a slightly lower overall accuracy of 0.84, but recall for *tvSpecial* and *video* improved, providing a more equitable classification across categories.
Analysis of the support vectors confirmed this effect. In the unbalanced RBF, nearly all points of minority classes became support vectors, while in the balanced model the total number of support vectors increased and was more evenly distributed across classes, indicating a more complex but fairer decision function.
Table 3 summarizes the main results, including the parameters used for each kernel and the corresponding test performance. This information allows the experiments to be fully reproducible.

Table 3: Comparison of SVM models on the IMDb classification task.

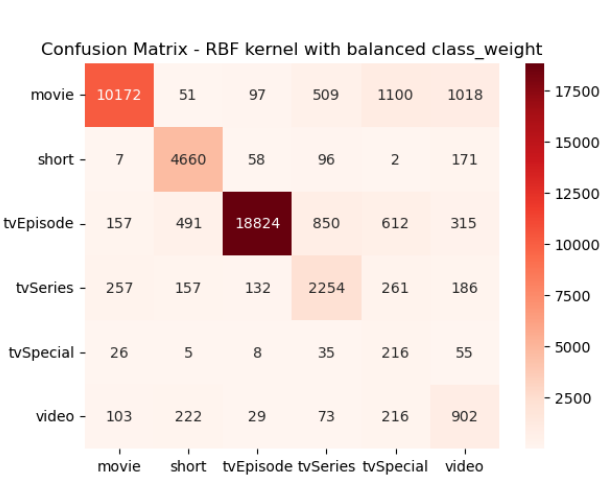
Model	Best Params (main)	Test Accuracy	Macro F1-score
Linear SVM	$C = 100$, $max\_iter = 1000$	0.81	0.45
RBF kernel	$C = 10$, $\gamma =$ scale	0.90	0.64
Polynomial kernel	$C = 10$, degree=3, $\gamma =$ auto	0.90	0.64
Sigmoid kernel	$C = 0.1$, $\gamma =$ auto	0.65	0.36
RBF (balanced)	$C = 10$, $\gamma =$ scale, balanced	0.84	0.65



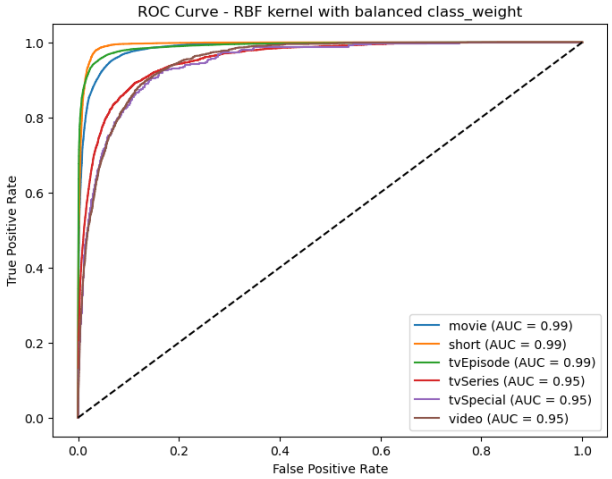
(a) Linear SVM (b) Polynomial kernel (c) Sigmoid kernel (d) RBF kernel

Figure 1: ...

In conclusion, non-linear kernels were clearly superior to the linear SVM, with RBF and polynomial achieving comparable accuracy. The RBF kernel with balanced class weights provided the best compromise, maintaining strong performance on majority classes while improving recognition of minority ones.



(a) Confusion Matrix RBF balanced



(b) ROC RBF balanced

Figure 2: ...

5 Advanced Regression