Movie Genre Classification

Théo Druilhe, Pierre Larose, Sigurd Saue and Killian Steunou

February 2024

Abstract

This project aim is to use the methods of principal component analysis, clustering, discriminant analysis and of classification and regression trees on a text dataset. In the end, we used the class-map concept to inspect in more details the performance of the discriminant analysis. Our peculiarity is the usage of natural language processing methods to create a tokenized and vectorized dataset of movies descriptions. We obtained an accuracy of 60% with discriminant analysis method for predicting the genre of a movie based on text input (vectorized).

Contents

1 Introduction

In this project, we want to show that we can apply the methods learned in the high-dimensional data analysis and machine learning course to deal with textual data. Our aim is to create a machine learning model able to predict the genre of a film based on its textual description. The data we use comes from IMDb (Internet Movie Database). The initial data consists solely of text, so we need to carry out appropriate processing to transform this data into tabular (and numerical) data.

After pre-processing, we will implement a PCA to reduce the dimensionality of our dataset. Next, we will try an unsupervised clustering method, namely K-means, to see if our data is separable and discover partiular groups of description. Next, we will use two supervised methods to predict the genre of a movie. We will use a linear discriminant analysis model, followed by a classification tree to achieve our goal.

Finally, we will present and use the Classmap method to visualize the results of the discriminant analysis and interpret our classification.

2 Work allocation

Before talking about the division of tasks, it is important to know that everyone was involved in every stage of the project. Of course, depending on each person's tastes and aptitudes, we each worked on specific tasks. These are as follows:

- **Data Pre-processing:** Killian & Théo
- **PCA:** Killian & Pierre
- **Discriminant Analysis:** Théo
- **K-Means:** Killian & Sigurd
- **CART:** Pierre
- **Classmap:** Sigurd
- **Presentation Suppport** Théo

3 Data sources and preprocessing

3.1 Data Source

Our dataset, originating from IMDb, contains a total of 108,414 records, each representing a distinct film or show, and the following information: title (the title of the movie), description (a short description of the movie) and genre (the genre of the movie, for example documentary, adventure, etc).
There are 27 different genres, which distribution is shown in table 1a.

We can see the initial distribution is very unequal, with some genres being over-represented. To overcome this, we regrouped the genres in the following way: genres with similar themes or audience appeal were combined into broader categories to achieve a more balanced representation and to simplify the analysis. For instance, "Thriller" and "Horror" were merged into "Thriller/Horror" to encapsulate the full spectrum of suspenseful and scary content. Similarly, genres that often share elements, such as "Action," "Adventure," "War," "Sci-Fi," and "Western," were all consolidated under "Action" to represent dynamic content. The "Drama" and "Romance" genres were kept within "Drama," reflecting their focus on emotional narratives and character development. "Family" and "Animation" were

grouped together to cater to content that is generally family-friendly, while "Music" and "Musical" were combined to cover all music-related content. The documentary field, including "Documentary," "Biography," and "History," was unified under "Documentary" to encompass all non-fiction and educational content. Live entertainment and informative content, such as "Game-Show," "Sport," "Reality-TV," "News," and "Talk-Show," were grouped into "Live," highlighting their real-time or reality-based aspects. "Mystery," "Fantasy," and "Crime" were categorized as "Police" to focus on genres typically involving investigation or fantastical elements. Lastly, "Comedy" and "Short" remained in their own distinct categories due to their unique characteristics that do not neatly fit with others.
We also removed the genre "Adult" from the data to avoid having sensitive content.
After this regrouping step, we obtain the distribution shown in table 1b.

Genre	**Count**
Drama	27225
Documentary	26192
Comedy	14893
Short	10145
Horror	4408
Thriller	3181
Action	2629
Western	2064
Reality-TV	1767
Family	1567
Adventure	1550
Music	1462
Romance	1344
Sci-Fi	1293
Adult	1180
Crime	1010
Animation	996
Sport	863
Talk-Show	782
Fantasy	645
Mystery	637
Musical	553
Biography	529
History	486
Game-Show	387
News	362
War	264

(a) Initial Genre Distribution

Genre	**Count**
Drama	28569
Documentary	27207
Comedy	14893
Short	10145
Action	7800
Thriller/Horror	7589
Live	4161
Family	2563
Police	2292
Music	2015

(b) Regrouped Genre Distribution

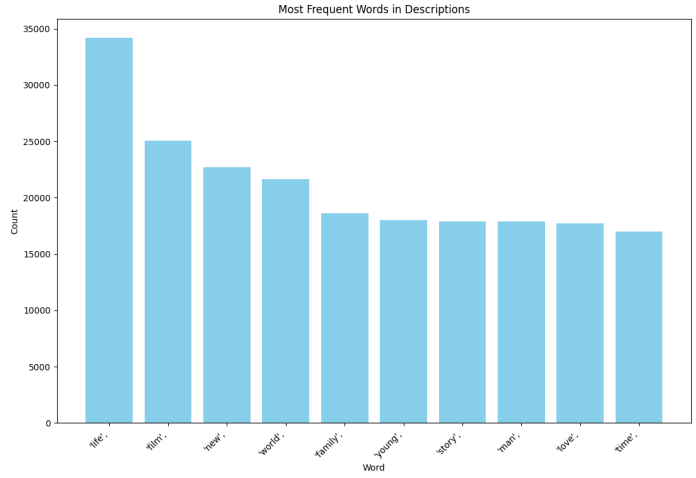
Table 1: Comparison of Genre Distributions

3.2 Data Preprocessing

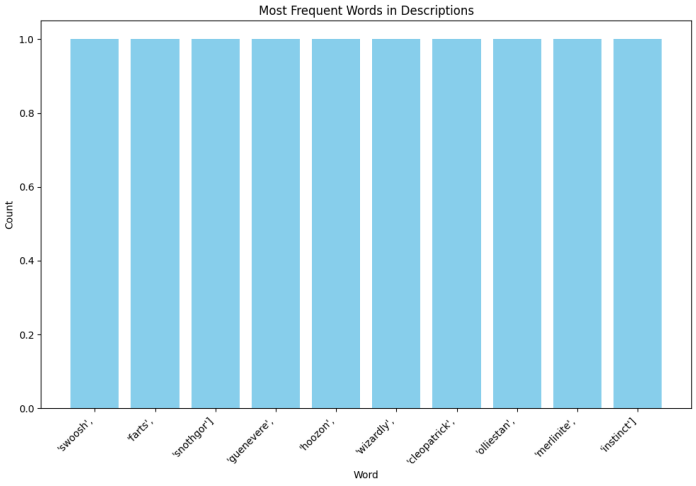
In this study, we outline our methodological approach for preparing and transforming raw textual data into a numerical form suitable for machine learning tasks. Specifically, given the initial dataset, we executed several sequential transformation steps described below.

3.2.1 Tokenization

Initially, natural language processing techniques were applied to convert unstructured text entries into structured numerical representations via tokenization and removal of frequently occurring stop words. We utilized the SpaCy NLP library to achieve this goal. Further preprocessing included eliminating unnecessary white spaces and special characters while also converting all characters into their lowercase equivalents. These procedures were implemented across the entirety of the movie descriptions.
After this step, we can look at the words distribution among the tokenized entries. Figure 1 shows the extreme quantiles.
In figure 1a we can see the most present words are the one we expect to find in a movie description, such as 'life', 'story', 'film', etc.
In figure 1b we can see the least represented words are not very often used words, or not commun proper nouns.



(a) Top 10 most present words



(b) Top 10 least present words

Figure 1: Words distribution in the tokenized data from movie descriptions.

3.2.2 Vectorization

Subsequently, we constructed dense vector representations for each distinct term existing in the processed corpora through application of the Word2Vec algorithm [1]. By averaging these term-specific embeddings per record, we obtained comprehensive sentence-level embeddings that captured intricate relationships between terms embedded therein. Afterwards, said embeddings were concatenated to the original dataset constituting additional variables.

4 Dimensionality Reduction: Principal Component Analysis (PCA)

After the data is preprocessed, each observation has now 100 new numerical variables, one for each embedding dimension. We perform a Principal Component Analysis to try to reduce the dimension of the data.
After running the PCA, we obtain the scree plot in figure 2.

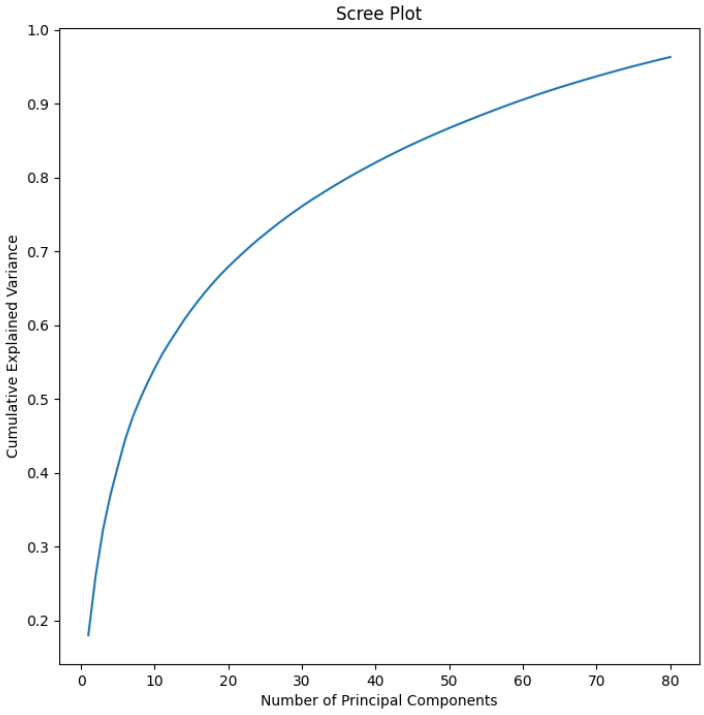


Figure 2: Scree Plot depicting Cumulative Explained Variance along the vertical Y-axis against the Number of Principal Components plotted on the horizontal X-axis. The visualization stems from conducting Principal Component Analysis on the IMDb description dataset comprising text descriptions of films. As illustrated in the diagram, higher Eigenvalues correspond to greater variance captured by subsequent principal components until reaching diminishing returns at around 37 components, indicating that these adequately summarize most meaningful patterns within the dataset while reducing dimensionality.

We can see that 37 components keep 80% of the variance, so we will proceed with using these 37 principal components for our further data manipulation. By reducing the dimensionality to these components, we can simplify our data without losing significant information, making our models more efficient and potentially improving computation time while retaining the essence of the dataset's variability. This approach strikes a balance between complexity and performance, allowing us to focus on the most informative aspects of the data for predictive or clustering tasks.
Note that we use PCA only for dimensionality reduction on the 100-dimensional sentence vectors to improve the performance of further models and analysis. Due to the nature of our data, it is hard to have a qualitative interpretation of the PCA.
We can visualize our data on the first two components, as shown in figure 3.

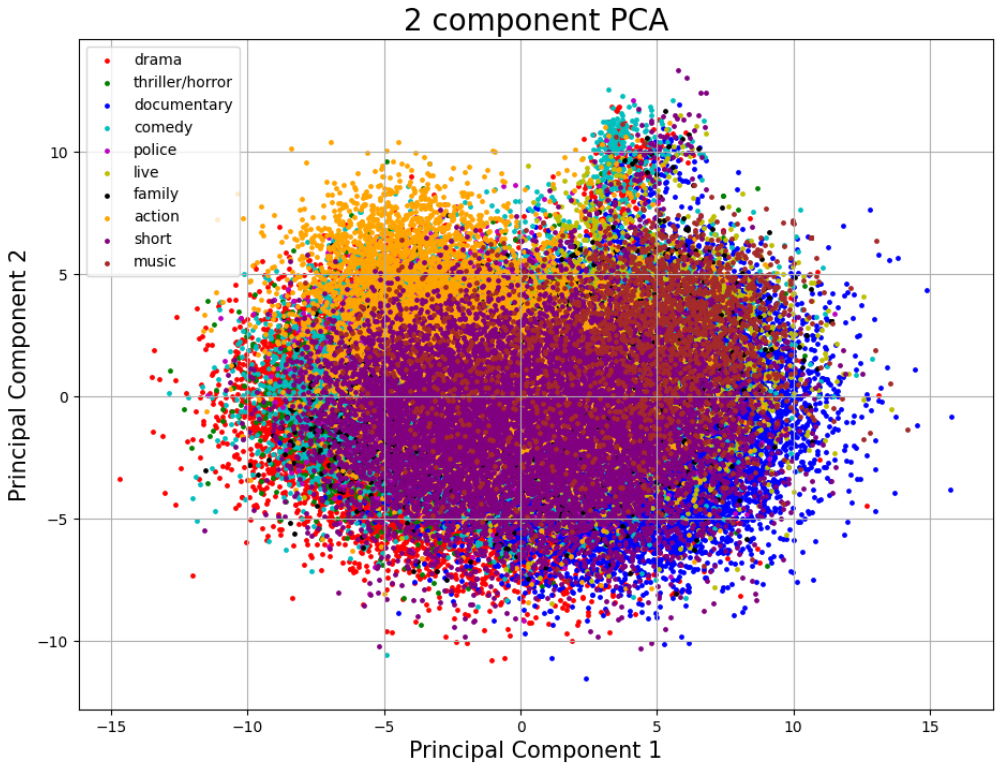


Figure 3: Two-dimensional representation of film descriptions obtained via text feature extraction techniques projected onto the initial two principal components derived through PCA transformation of the imdb description dataset consisting of narrative summaries for various motion pictures. Color-coding accompanying this visualization illustrates the different genres.

We can see that the different groups seem quite separable on the first two components. This motivates the interest in clustering and is encouraging for the future, as having separate groups could improve the quality of our classification models.

We will now perform a clustering on the obtained principal components.

5 Clustering (K-Means)

As we have seen in figure 3, some movie genres seem to be separable along the first dimension, motivating the use of a clustering algorithm to discover groups among the movie descriptions. For this, we will implement K-means on our principal components.

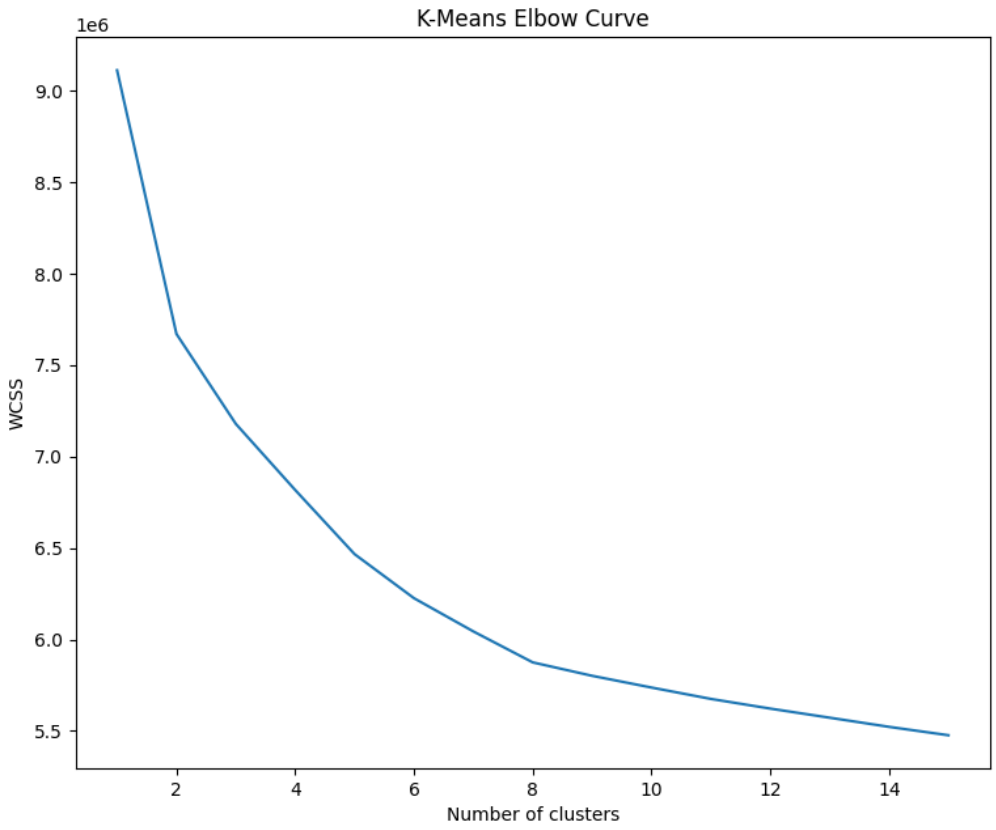


Figure 4: Elbow curve displaying the relationship between within-cluster sum of squares (wcss) and increasing cluster counts in K-Means algorithm. Decisively declining inflections indicate points where additional clustering offers minimal improvement, serving as guides for selecting suitable cluster divisions balancing model fit and parsimony.

Figure 4 shows an inflexion point at 8 clusters, so we will run K-Means with 8 clusters. Yet, retrospectively, one could have taken a more educated guess at the best k to choose using the Silhouette method in collaboration with the traditional Elbow one. Although the Silhouette method is not more accurate than the Elbow method, being mainly visual too, it adds some valuable intel about thee distance between clusters that for sure can help.
The data seem quite well separated as shown in figure 5, even though we cannot see the cluster 0 represented on this figure.

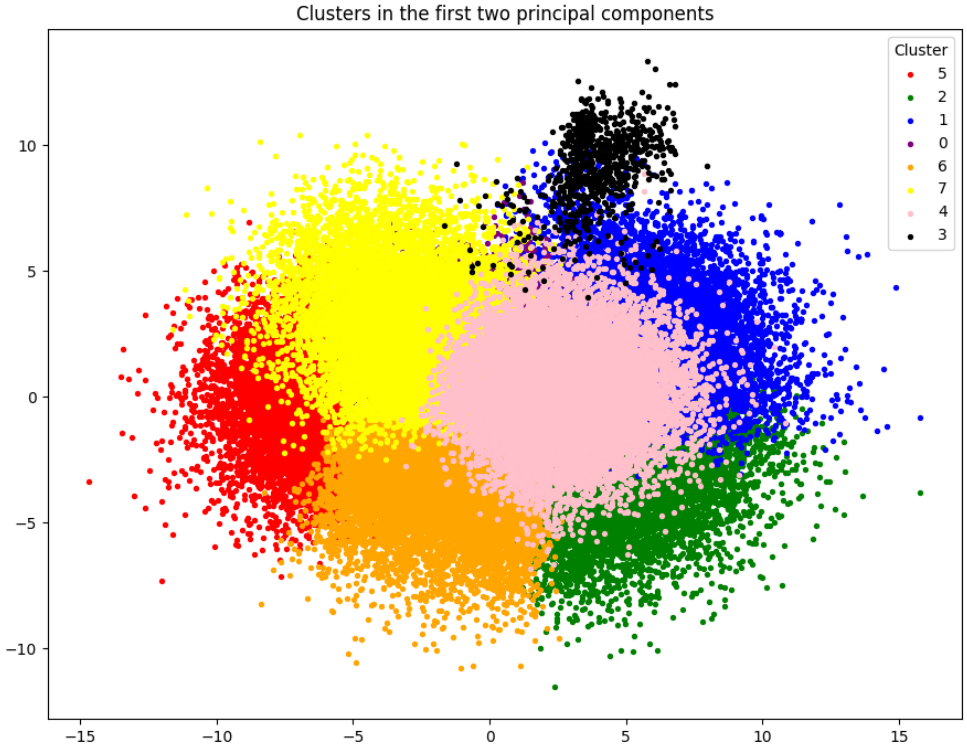


Figure 5: Two-dimensional representation of film descriptions obtained via text feature extraction techniques projected onto the initial two principal components derived through PCA transformation of the imdb description dataset consisting of narrative summaries for various motion pictures. The data is colored according to the 8 clusters discovered by the K-Means algorithm.

The data distribution among the clusters are shown in table 2. While all clusters are relatively evenly distributed (between 12806 and 19110 movies per cluster), one cluster stands out with only 795 movies. Further examination of the descriptions of the movies in this smaller cluster is needed to determine its relevance and importance.

Cluster Number	**Number of Movies**
0	795
2	12806
5	12975
4	14172
7	14570
6	16186
1	16620
3	19110

Table 2: Distribution of movie clusters resulting from K-means clustering. One row per cluster with the corresponding count of movies assigned to each group represented in the second column.

Table 3 shows that compared to the other clusters, cluster 0 allocates a comparatively larger share of its total movies to the comedy genre, accounting for roughly 36.28%, even though cluster 6 shows a similar proportion of comedy movies. It has a quite large proportion of drama movies (29.58%), but this does not differentiate it from the other clusters. This table does not help us that much to interpret the clusters.

genre cluster	action	comedy	documentary	drama	family	live	music	police	short	thriller/horror
0	2.65	36.28	12.90	29.58	2.40	2.28	0.88	1.01	10.11	1.90
1	4.19	7.76	12.32	45.14	2.26	0.84	0.34	2.62	13.09	11.43
2	2.26	9.85	42.99	5.18	3.08	14.30	10.66	0.61	9.47	1.60
3	2.68	21.81	1.21	55.20	2.19	0.30	0.63	1.67	6.98	7.34
4	1.75	3.49	59.66	10.75	1.43	4.50	0.83	0.83	15.79	0.98
5	32.56	11.89	2.00	21.05	1.90	0.39	0.26	6.64	3.67	19.64
6	4.23	33.83	9.80	20.36	4.67	7.19	1.67	1.62	10.60	6.02
7	7.71	2.52	61.98	14.18	1.03	1.80	0.30	1.45	6.26	2.77

Table 3: Distribution of genre categories across the different clusters of movie. Each row represents a distinct cluster, while the columns show the percentage distribution of various genres within those clusters. For instance, Cluster 0 has approximately 2.65% of its movies categorized under 'action', 36.28% under 'comedy', and so forth.

Eventually, we can sample from our data some examples of movies from cluster 0, presented in table 4. Among the sampled 5 elements, we have two shorts, two comedies and one drama.

Most of them contain a blend of humor and real-life issues, making them engaging and relatable. Some examples of this mix include "Ask Will," which deals with Shakespeare trying to succeed as a romance advice columnist despite having trouble navigating relationships himself, and "Girl Please!," a satirical talk show discussing topics ranging from pop culture to social norms.

Additionally, there is a strong presence of artistic expression and creativity in these descriptions. Works like "Diskzokej" and "Pigan Brinner!" rely heavily on symbolism and creative visualizations. Meanwhile, "Andrew Sawyer's Ichabod" brings a unique twist to classical literature through its adaptation of "The Legend of Sleepy Hollow."

The description are also quite imaged and poetic, which could explain that this cluster contains few movies.

Title	Genre	Description
Diskzokej (1980)	short	An alarm clock wakes a man who washes his face, has breakfast, drives his car to work, spins records, returns home, and takes his pills. It's a world of circles - often seen from above: an espresso cup, a stairwell, the pills, and the records spinning. At the dance where the music plays, the rhythms evoke images of a butcher slicing head cheese, gears driving other wheels and levers, a combine churning out bales of hay, a butcher cutting chunks of meat for a stew, and boxers punching. The circle of music and life.
"Ask Will" (2017)	comedy	Will Shakespeare was never any good at romance. With no job, no relationship, and no way to pay rent, he reluctantly takes a contract as a romance advice columnist for a tacky local magazine. Things take a turn for the awkward when his neighbors become his anonymous contributors. As Will tries to keep his identity secret, he is also struggling to publish his script and gain recognition as a serious writer. ASK WILL is a quirky web comedy about navigating relationships and pursuing your calling.
Pigan brinner! (2008)	short	"Pigan brinner!" ("Maid on fire!") is a "modern silent short", shot in black/white 35 mm with the hand cranked camera Pathé, from 1914, (actually the same camera used by D.o.P. Julius Jaenzon for "Phantom Carriage") telling the story of a Maid that slaves in a Swedish family high-etc kitchen in the year of 2008, serving some twin brats, a hungry Nosferatu-teenager and a father "dying" in a cold. All tensed up, she suddenly starts to burn and SELFCOM-BUST!... Maid on fire! is a torch in the Swedish debate about maids.
Andrew Sawyer's Ichabod (2007)	drama	In this classic retelling of Washington Irving's, "The Legend of Sleepy Hollow" director, Andrew Sawyer and writer Allison Lahikainen bring a stage feel to the screen. When Ichabod Crane, the new schoolmaster, moves to Sleepy Hollow he encounters more than rambunctious children and curious town folk. Someone or something is after his head. He must confront the Legend of Sleepy Hollow in order to find peace with the woman he loves, if he can survive the night.
"Girl Please!" (2006)	comedy	A talk show adding a hip urban vibe that is guaranteed to make you laugh. They tell it like it is! Nothing to hide - No fronts. Just a fresh opinion. They talk about it all! This show concept is like the View meets Sex in the City and a dash of In Living Color and okay...a little QUEER EYE for the Straight Girl!!! Can it get any better than that?

Table 4: Sample of five randomly drawn elements from cluster 0 in the dataset.

As anticipated, interpreting vectors originating from text descriptions proves challenging since the extracted features tend to lack explicit meanings and intuitive explanations, unlike structured quantitative variables.
Let us move on toward a more predictive task by using Linear Discriminant Analysis (LDA), a technique employed for elucidating the underlying covariance structure and maximizing class separability.

6 Discriminant Analysis (DA)

We divided our data into two parts: one called the training set, which we use to build our model, and the other known as the test set, which comprises 20% of the entire data. We reserve this portion for checking how accurately our model performs on unseen data. This division guarantees that our conclusions are trustworthy and applicable in practice, preventing problems like overfitting and bias.

We train the Linear Discriminant Analysis (LDA) model on the training data (on the variables recovered from PCA), specifying 4 components to the model, since it kept more than 80% of the cumulative explained variance. We assess its performance on the test data, that we show in table 5. We can also visualize the report as a heatmap (figure 6).

Class	Precision	Recall	F1-Score	Support
action	0.57	0.54	0.55	1521
comedy	0.55	0.47	0.51	2954
documentary	0.73	0.77	0.75	5574
drama	0.60	0.71	0.65	5714
family	0.39	0.25	0.30	509
live	0.43	0.52	0.47	794
music	0.36	0.59	0.45	397
police	0.26	0.12	0.16	465
short	0.49	0.30	0.37	2008
thriller/horror	0.56	0.57	0.57	1511
accuracy			0.60	21447
macro avg	0.49	0.48	0.48	21447
weighted avg	0.59	0.60	0.59	21447

Table 5: Classification Report on Linear Discriminant Analysis

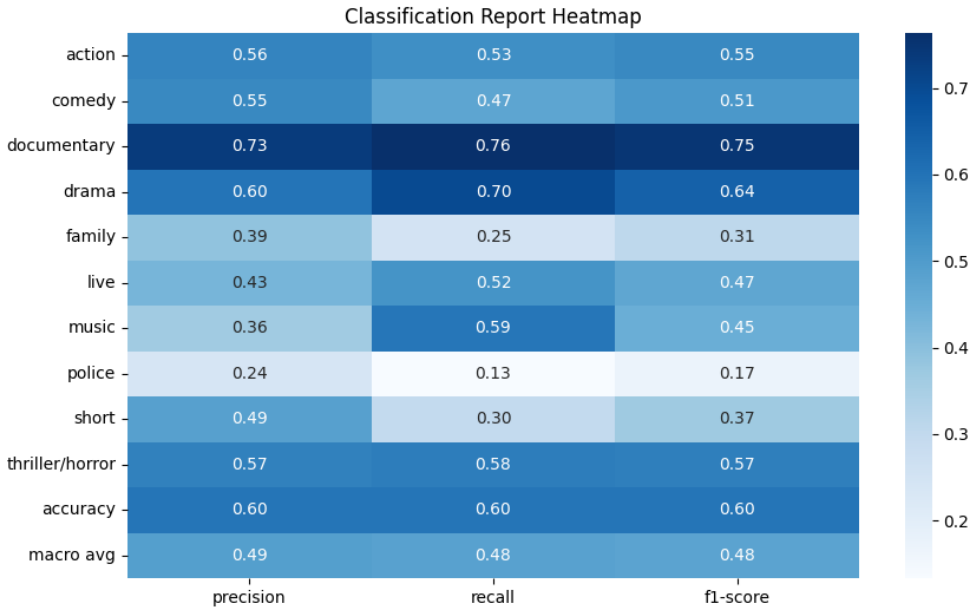


Figure 6: Heatmap for Discriminant Analysis classification result

The table lists the precision, recall, and F1 scores for each genre. Generally, a higher score denotes superior model performance in handling that specific class.

The model is best at predicting the documentary genre with a precision, recall and f1-score of respectively 73%, 77% and 75%. It is also good at predicting the genre drama, action, comedy and thriller/horror. This is not a surprise since the number of movies of these genres is higher than the one of the genres family, live, music and police. We can see the genre short is not predicted well by the model, which seems reasonable since a short movie can have a description like any other genre (we could think of a short horror movie or a short comedy..).

Overall, the accuracy of our model is 60%, which is a good performance. Indeed, even we're trying to predict ten classes, so if the algorithm chose one class at random, the accuracy would be 10%.

If we go back to the beginning of the report, when we presented the distribution of the different genres, we can see that it wasn't uniform. Some genre were abundantly present in our data set, while others were not. This is the case for the documentary and police genres. This may be one of the reasons for the difference in accuracy between the two genres. To test this hypothesis, we created a sample of our data set containing 1000 randomly selected observations of each genre. We then performed the Linear Discriminant Analysis model, and we end up with a lower overall accuracy (50%).

This can probably be explained by the fact that the model needs a lot of observations to learn properly, and so to increase accuracy we'd need to increase the number of observations. We'd need a larger dataset.

We can visualize the transformed data from the LDA projected on the first 2 components of the dimensionality reduction result in figure 7.

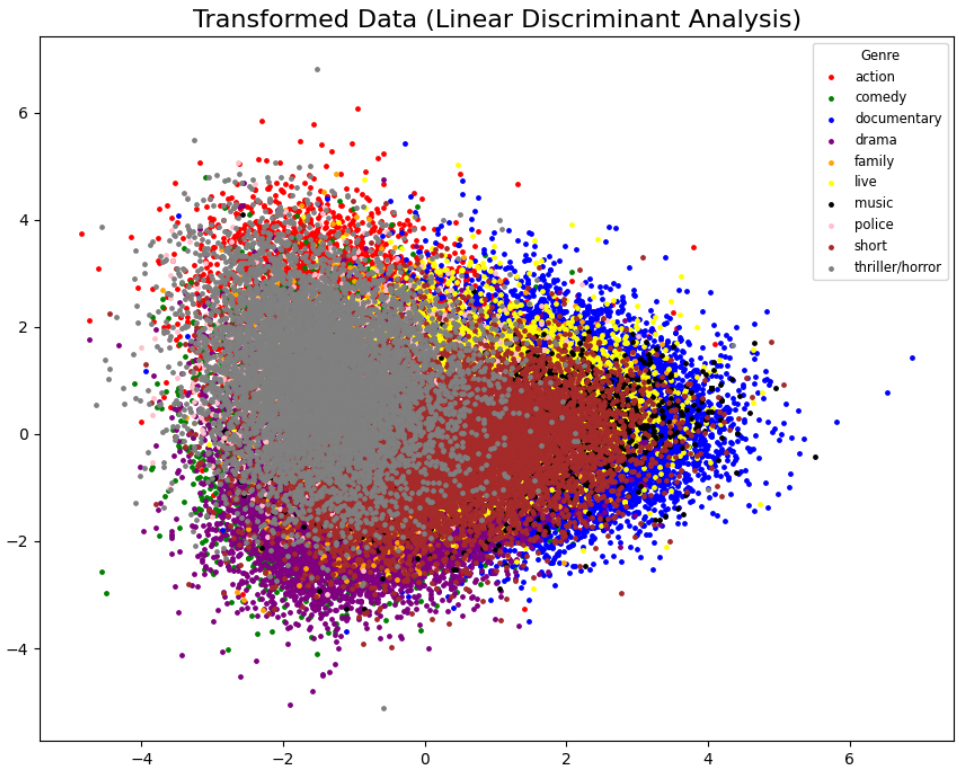


Figure 7: Transformed data after Discriminant Analysis

Now we will implement a different method for classification: classification tree.

7 Classification and Regression Trees (CART)

Below we used the CART method to predict the genre of movies using the 37 previously created components and the belonging to any cluster. We tried at first a tree on the tokenized dataframe before pca and got a 0.36 accuracy score. Then we created a second tree this time using the dataframe with pca implemented and improved the accuracy score to 0.43. In a third attempt to improve our accuracy, we used the belonging to any cluster as a new intel for the cart algorithm to use otherwise, it did not improve the accuracy score at all.

Retrospectively, this was quite a dumb idea. Indeed, clusters allocation is determined using the information provided by our 37 components alone. But here we already have all that information for our tree to use. Adding the column with clusters added no extra information, explaining its null-impact.

Finally, we pruned the tree and incrementally found 8 to be the best depth to choose. With all that was mentioned we get the tree in figure 8 with accuracy score equal to 0.53.

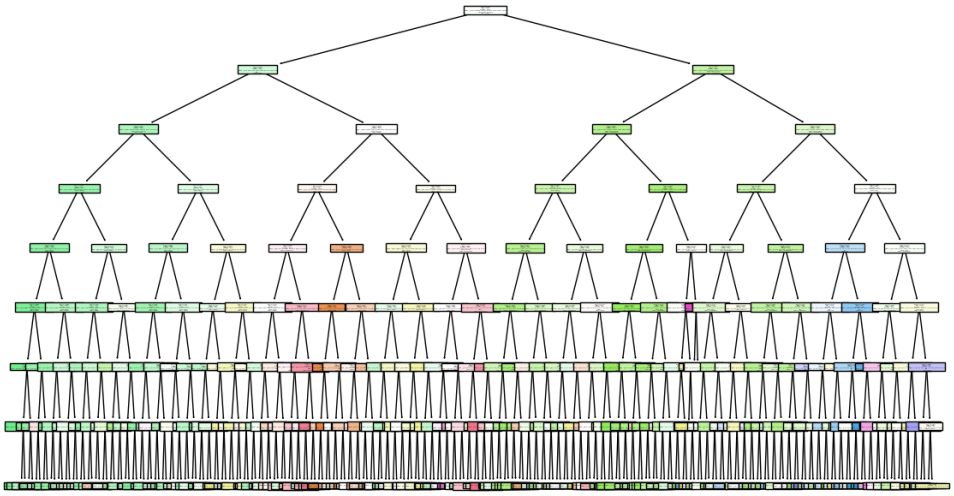


Figure 8: Decision Tree Classifier

We won't interpret the nodes of the tree because of the nature of our data and the fact that we are using components which are even more difficult to interpret. Moreover, the only thing that interest us is the prediction capability of this model and how it compares with its counterparts like discriminant analysis.

Knowing that discriminant analysis provide more accurate predictions in our particular case, we won't dwell on the CART, let just say that for the sake of comparison, implementing it is always worth it.

Finally, the interpretation issue mentioned earlier leads us to the last section, describing a method that help us visualize our data.

8 Class map

To better understand and visualize our data, we will use the Class map method [2]. The class map method can be used on the train set when we do a classification. To do this, we use a model which, for each individual, outputs a probability of belonging to each class of the target variable. We therefore decided to use class map with the discriminant analysis method. The graphics obtained with class map can be used to visualize many things. We can see the probability of the individual being classified in an alternative class, which means not his own, the distance of the individual from his given class, if the individual is an outlier, that is far from all the classes, and above all, we can see in which category the model has classified the individual if it has made a wrong prediction, which lets us know how close two classes are to each other.

We used this method on our train set, and obtained ten graphs, one for each movie genre. This allowed us to observe a lot of interesting things.

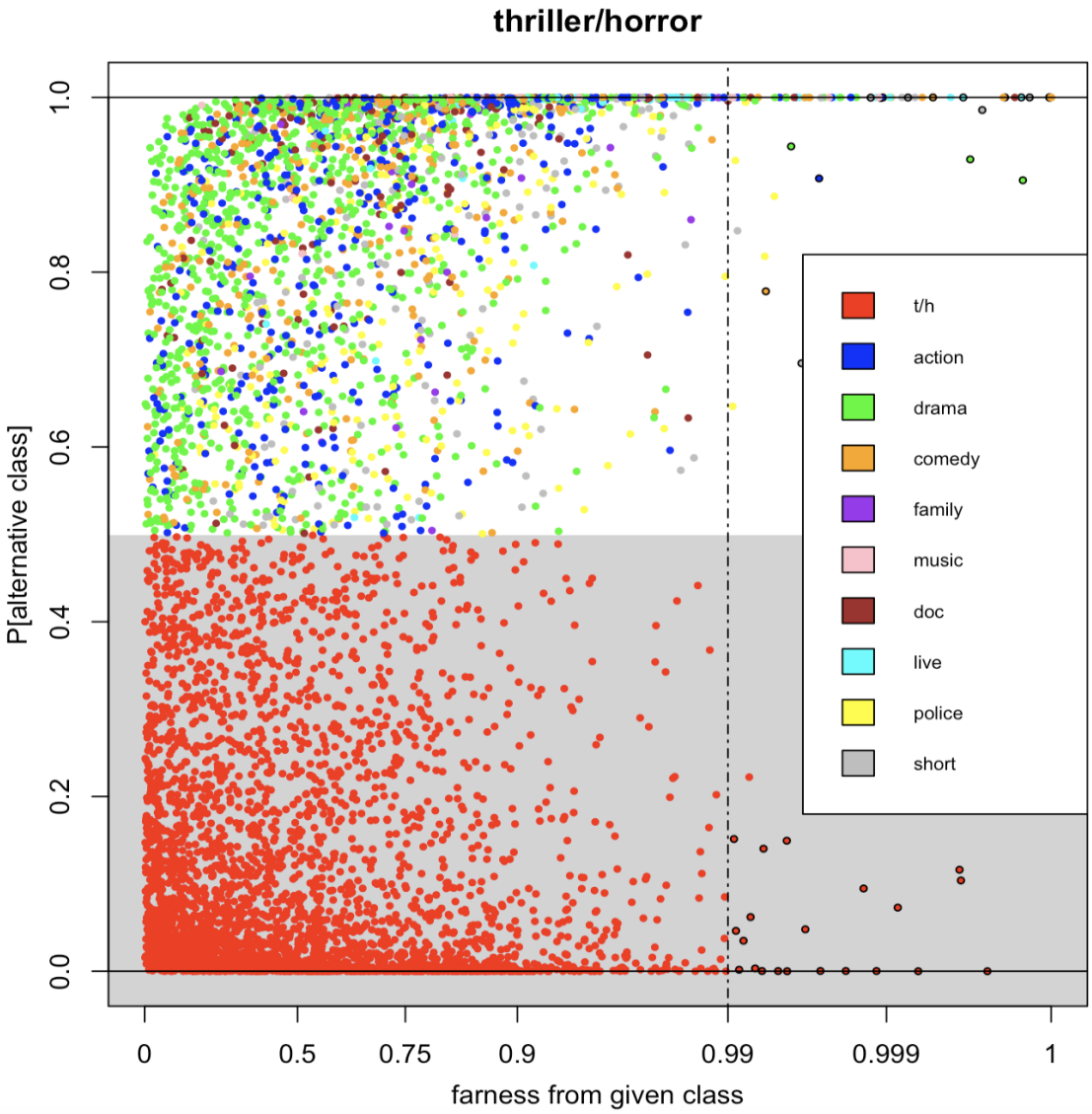


Figure 9: Class map of "thriller/horror"

On the figure 9, we can observe all the movies of the genre "thriller/horror". All of them represented by a red dot in the light grey rectangle below were well classified by the model, unlike those above the rectangle. The most interesting thing to note is that the majority of individuals misclassified by the model are in light green, which stands for the genre "drama". We can therefore assume that these two movie genres have pretty close descriptions. But as we saw earlier, the "drama" genre is over-represented in our dataset. This over-representation influences the model to make more predictions of the drama genre than of less represented genres. One solution would be to equalize our dataset by taking samples of equal size for each genre, but since we have textual data, we need a very large dataset, and this method greatly reduces the performance of our models. A possible solution to this problem might be to use a SMOTE to generate artificial data for our least represented genres.

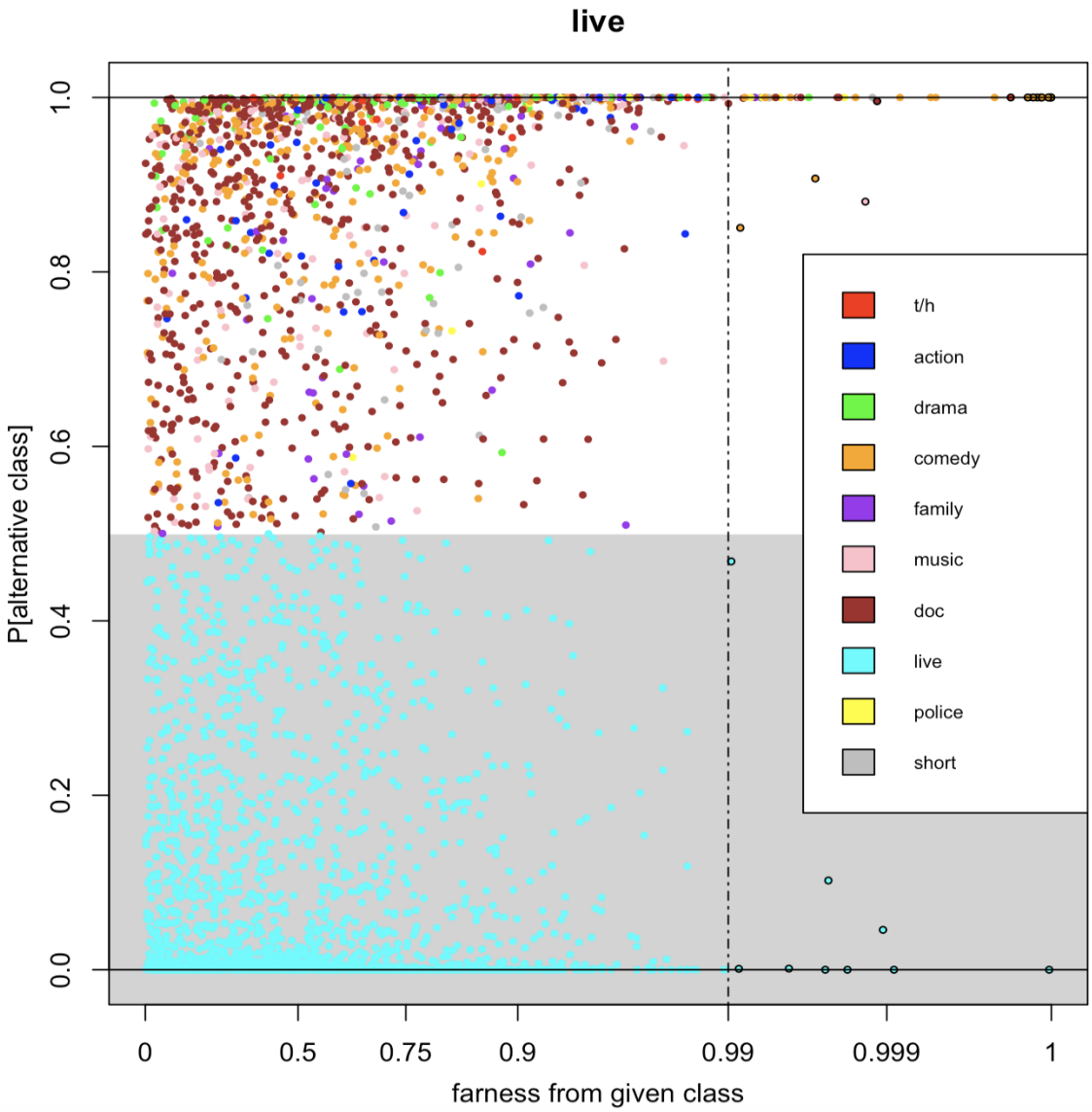


Figure 10: Class map of "live"

Despite our badly distributed data, we can draw some very interesting conclusions from our class maps. For example, the figure 10 shows the class map for the "live" genre, and we can see that bad predictions are almost never from the "drama" genre, despite the fact that the latter is over-represented. We can therefore safely say that the "live" and "drama" movie genres are far apart in terms of their descriptions. All the ten class maps are available on our GitHub repository.

9 Conclusion

This project aimed to explore the application of machine learning techniques to classify movie genres based on their descriptions from IMDb. Through a series of steps, including data preprocessing, dimensionality reduction, clustering, and predictive modeling, we attempted to tackle the challenge of

understanding and categorizing textual data.

Our initial efforts focused on preparing the text data, transforming it into a numerical format suitable for analysis. Using Principal Component Analysis (PCA), we reduced the dimensionality of our dataset, which helped in managing the computational complexity of the models we intended to use. The K-Means clustering provided some insights into the natural groupings within the data, although the direct relationship between these clusters and movie genres was not always clear.

In terms of predictive modeling, we employed Linear Discriminant Analysis (LDA) and Classification and Regression Trees (CART) to classify movies into genres. The LDA model achieved an accuracy of 60%, indicating a reasonable level of performance but also highlighting the difficulty of genre classification with textual descriptions alone. The CART model, with slightly lower accuracy, reinforced the idea that while it's possible to classify movies based on descriptions, there's significant room for improvement.

Throughout this project, we faced challenges associated with natural language processing and the interpretability of machine learning models trained on text data. These challenges underscore the complexity of working with text and the limitations of our current approaches.
Looking ahead, there's potential for improving the accuracy and effectiveness of movie genre classification by exploring advanced text processing techniques and incorporating additional data sources.

In summary, this project demonstrated the feasibility of using machine learning for classifying movie genres from text descriptions, provided insights into the challenges of working with textual data, and outlined directions for our future research.

References

[1] Radim Rehurek and Petr Sojka. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2), 2011.

[2] Jakob Raymaekers and Peter J. Rousseeuw. Silhouettes and quasi residual plots for neural nets and tree-based classifiers, 2021.

Appendix

A Code

You can find all our well documented code on this GitHub repository (https://github.com/theodruilhe/MovieGenreClassification).