TÉCNICO LISBOA

NLP 24/25: Finding the Genre of Movie Plots

Author: Carolina Pinto 100463, Francisco Abreu 110946, Vasco Gameiro 110881

1 Introduction

This report details the development of a **movie genre classification pipeline**. Two preprocessing approaches were used: classical **Natural Language** (NLP) techniques and **Sentence Transformer**-based embeddings. The best-performing solution was an **Histogram Gradient Boosting Classifier** trained on the movie description sentence embeddings, achieving **67.5% accuracy** on the test set.

2 Exploratory Data Analysis (EDA)

An initial **exploratory data analysis (EDA)** was performed. Various issues were identified, including variations in **director names** due to regional differences and misspellings, entries with multiple or unknown directors, and movies with multiple genres. Additionally, there were **18 duplicate movies**, and additional duplicates (same description) but with variations in director or title. The **genre distribution** was highly imbalanced, with *"Drama"* significantly overrepresented compared to genres like *"Animation"* and *"Sci-Fi"*. The *description* column had a **right-skewed token distribution**, with a meaningful number of descriptions exceeding **500 tokens**, and multiple outliers according to boxplots. When applying **PCA** to *description*'s **TF-IDF vectorizations and embeddings**, significant **genre overlap** was found, although some genres formed **denser clusters** in certain regions.

3 Models

Two distinct approaches were used to prepare the data for the models. The **first approach** involved **classical NLP** techniques with feature engineering. **Director names** were cleaned and standardized using string operations and fuzzy matching, then encoded with a *CountEncoder* based on the frequency of each unique director-description pair. For entries with multiple directors, only the most frequent one was retained, and duplicates were removed. **Titles** and **descriptions** were processed by expanding contractions, tokenizing, and lemmatizing. *Stop word* removal and *noun phrase* addition were tested but not used further. Additionally, **log-ratio analysis** was used to identify the most discriminative tokens in the *description* by comparing the likelihood of a token appearing in one genre versus another, retaining up to **25,000 tokens** per genre. A *region* feature was also created by grouping the *from* column into regions like *"Western"* and *"East Asian"*, and encoded with a *similarity encoder* as a proxy for geographic similarity (e.g., *"East Asian"* and *"South Asian"* are closer than *"Western"*). **Text data** was vectorized using *TF-IDF* for individual tokens, excluding n-grams to reduce computational costs. Since some models do not support sparse matrices, *TruncatedSVD* was applied to create dense representations. For this approach, three models were tested: (1) a **Histogram Gradient Boosting Classifier (HGBC)** with dense vectors; (2) a **Support Vector Classifier (SVC)** with sparse vectors; and (3) an **SVC** with *feature selection (FS)* applied to sparse vectors.

The second approach used neural *description* embeddings from the *GIST small Embedding v0* [2], a top-performing model for classification tasks under **100M parameters** on Hugging Face. For descriptions longer than **512 tokens**, recursive chunking split the text into chunks of up to **512 tokens** with a **50-token overlap**. The chunk sizes were kept similar by recursively adjusting the split points. Each chunk was encoded individually, and a weighted average of the embeddings, giving more weight to longer chunks, was computed. Both **HGBC** and **SVC models** were trained using these embeddings.

4 Experimental Setup and Results

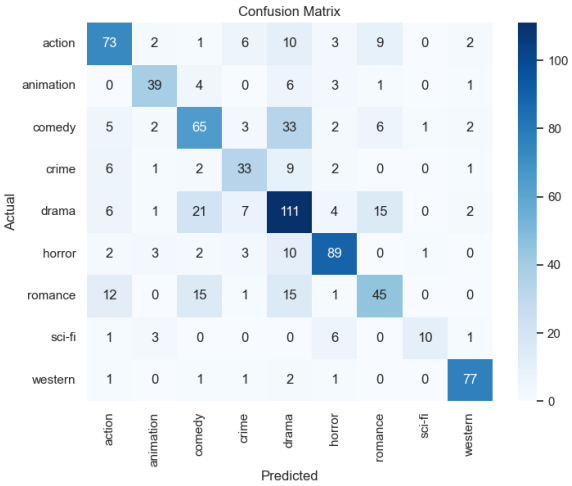
The data was split into 90% training and 10% testing, with a fixed random seed for reproducibility. *Sklearn* pipelines handled preprocessing for classical models, simplifying cross-validation and preventing data leakage. Hyperparameter tuning was done with *RandomSearchCV* (4 folds, 50 iterations), but HGBC with SVD was limited to 10 iterations due to computational cost. After tuning, models were retrained on the full training set and tested on the test set. Results were validated using accuracy, the classification report, and confusion matrix. The table 4 below presents the test performance, F1-Scores for each class, and the confusion matrix of the best-performing model, which will be discussed next. It was observed that **10** plot descriptions were repeated between the original training and test sets. Although a rule-based approach could have been used to classify these cases, it was considered beyond the project scope. In the final predictions, it was confirmed that the model's predictions matched those that a rule-based system would have produced for these instances.

In addition, during error analysis, the model's performance on minority classes, such as **western** and **romance**, was found to be particularly sensitive to variations in plot length and description detail. Shorter descriptions often led to higher misclassification rates, as observed in **romance**, where plots with minimal context were misclassified into broader categories like *drama*. This suggests that, in future iterations, incorporating techniques like **text augmentation** or leveraging additional metadata (e.g., *director* or *release year*) could help mitigate the impact of sparse descriptions and further enhance model accuracy, especially for underrepresented genres.

	HGBC-SVD	SVC-noFS	SVC-FS	SVC-Emb	HGBC-Emb
CV Acc	0.614	0.663	0.658	**0.687**	0.669
Test Acc	0.641	**0.682**	0.667	0.676	0.675
F1 Macro-Avg	0.62	0.65	0.66	0.67	**0.68**
F1 Action	0.62	0.68	0.65	**0.69**	**0.69**
F1 Animation	0.76	**0.80**	0.78	**0.80**	0.74
F1 Comedy	0.47	**0.60**	0.59	0.57	0.57
F1 Crime	0.40	0.41	0.40	**0.62**	0.61
F1 Drama	0.63	**0.65**	0.61	0.60	0.61
F1 Horror	0.78	0.82	0.82	**0.83**	0.81
F1 Romance	0.57	**0.60**	0.57	0.50	0.55
F1 Sci-Fi	0.48	0.35	0.56	0.54	**0.61**
F1 Western	0.86	**0.92**	**0.92**	**0.92**	0.91

5 Discussion

All five models exceeded the benchmark test accuracy of 0.62, but HGBC-SVD underperformed, likely due to information loss and limited hyperparameter tuning. While SVC-noFS had the highest test accuracy (0.682), HGBC-Emb (0.675) is the more reliable choice. Its smaller gap between Cross-Validation and Test accuracy indicates better generalization, reducing the risk of overfitting. More crucially, HGBC-Emb achieved the highest F1 Macro-Average (0.68), showing stronger performance across genres. Given the unknown distribution of the unlabelled test set, relying solely on test accuracy is unwise.



For example, SVC-noFS struggles with Sci-Fi (F1 = 0.35), which could hurt performance if Sci-Fi movies dominate the unlabelled test set, while HGBC-Emb handles it better (F1 = 0.61), making it more balanced across genres. As a result, HGBC-Emb offers the best trade-off between accuracy and robustness, and was chosen for the final prediction.

The confusion matrix shows interesting class relationships. Comedies are often misclassified as dramas, likely due to overlapping themes and vocabulary. Drama was the most over-predicted class, expected as it's the majority. Romance was the most misclassified, while western was the best predicted, a pattern consistent across all models.

Using boxplots, we checked if there was a meaningfull difference between the original description size and missclassification, and we found out that smaller description had a meaningfull bigger tendency to be missclassifier, which is plausible since those descriptions have less information, this also raise the questions we should have removed the outlier descriptions that we found in EDA.

The UMAP visualization shows key patterns in genre classification. Confused genres like *comedy* and *drama* are clustered, suggesting overlapping features, while *western*, with the highest accuracy, forms a distinct cluster. **Romance**, the most misclassified genre, is more scattered, showing the model's difficulty distinguishing it. This supports the conclusion that HGBC-Emb's ability to capture subtle distinctions improves performance across genres.

Upon examining the misclassified instances, we noticed that despite high confidence (with some *max_proba* values reaching 1.0), several predictions were incorrect. For example, *drama* was misclassified as *animation*, and *comedy* as *horror*, indicating complexities in distinguishing certain genres. This suggests that even confident models can struggle with subtle distinctions in textual features. Further improvements, such as using more metadata or refining embeddings, could help address these challenges.

An interesting misclassification was *Bava Nachadu*, a **drama** predicted as **animation**, with a confidence of **proba = 1**. This may be due to keywords like "fracture" and "modeling," which overlap with themes in animated films where characters experience transformations. Additionally, the lighthearted elements in the plot may have caused confusion, as they are common in animation. This suggests that the model, even when highly confident, struggles with genres that share overlapping narrative elements.

6 Future Work

A key limitation was limited computational power, restricting hyperparameter iterations and cross-validation folds. More resources could allow broader parameter searches or bayesian optimization. Future work could explore models like Logistic Regression, Neural Networks, or fine-tuning transformers.

Further research could compare **TF-IDF** and sentence **embeddings**, independently or combined with features like *title* and *director*. Trying other embeddings, such as **GloVe** or **FastText**, might offer new insights. Addressing **class imbalance** via *SMOTE* or **class weighting** could improve minority genre representation. Lastly, deeper preprocessing analysis could refine the pipeline and boost HGBC-SVD's performance.

References

[1] Vasco Gameiro, *Natural-Language Repository*, GitHub, 2024. Available: `https://github.com/vascomgameiro/Natural-Language.git`. Accessed: 2024-10-18.

[2] Aivin V. Solatorio, *GISTEmbed: Guided In-sample Selection of Training Negatives for Text Embedding Fine-tuning*, arXiv preprint arXiv:2402.16829, 2024. Available: `https://arxiv.org/abs/2402.16829`.

[3] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux, *API design for machine learning software: experiences from the scikit-learn project*, in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.