Group 21
Carolina Baptista, ist195529 ana.p.baptista@tecnico.ulisboa.pt
André Correia, ist1102666 andre.m.correia@tecnico.ulisboa.pt
Tiago Firmino, ist1103590 tiago.esteves.firmino@tecnico.ulisboa.pt

Genre Classification of Movie Plots Using Natural Language Processing

ABSTRACT
This project aims to develop a machine learning model to classify movie genres based on textual data from movie descriptions. The focus was on using **Natural Language Processing** (**NLP**) techniques to create a genre prediction model. Initially, the data was loaded and preprocessed. The core methodology involved implementing a multimodal solution that combines model predictions based on weighted probabilities. Finally, the trained model was used to predict genres on an unlabeled test set, with results saved for further analysis.

Keywords
Machine Learning. Natural Language Processing. Genre Classification, TF-IDF, SVC, WordNet, Naive Bayes, Logistic Regression, Weight Probabilities, Confusion Matrix.

1. MODELS
The development process involved significant data preprocessing and the use of text vectorization techniques to convert movie plots into a format suitable for machine learning. Below, the key steps of the model development process are described, including preprocessing techniques and model selection.

1.1 Model Selection
To determine the best model to implement, we evaluated four approaches: **SVC**, **Multinomial Naive Bayes**, **logistic regression**, and a deep learning model using **BERT**. We began by implementing the three non-deep learning models. The Naive Bayes model was used with a simple CountVectorizer, which counts word frequencies rather than weighting them by importance, while both SVC and logistic regression used a TF-IDF vectorizer. After evaluating these models, we experimented with a BERT-based approach. Although BERT showed slight improvements, the extensive training time required led us to abandon this approach, due to hardware constraints. In the end, we chose to utilize a **multimodal approach** using **weighted probabilities** and the models SVC, Multinomial Naive Bayes and Logistic Regression.

1.2 Preprocessing
Preprocessing the raw text data was essential for improving model performance by cleaning the data and reducing noise. The raw text data was first preprocessed by removing **punctuation** using **regular expressions** and converting all characters to **lowercase**. This helped reduce the complexity of the text and ensured consistency. Then, stopwords, such as "and," "the," or "is," which do not contribute significant meaning to genre classification, were removed using the **NLTK** library's list of **English stopwords**. This step ensured that only relevant words remained for analysis.

The next step was to take the cleaned text and **tokenize** it, splitting it into individual words. After that, lemmatization was applied using NLTK's **WordNetLemmatizer**, which reduced words to their root forms, ensuring that words like "running" and "ran" were treated as "run" for a standardized vocabulary. Finally, unnecessary spaces were removed, leaving the cleaned and simplified text ready for the vectorization. For both the training and test datasets, we concatenated relevant fields, such as *title*, *from*, *director*, and *plot*, into a single *text* field to create a unified textual input for the model.

1.3 Vectorization
To convert the preprocessed text data into a numerical format for machine learning, two vectorization techniques were utilized: **CountVectorizer** and **TF-IDF (Term Frequency-Inverse Document Frequency)**. For the Naive Bayes model, **CountVectorizer** was employed to generate a bag-of-words representation by counting the occurrences of each word across all movie plots. This method provides a simple representation of word frequency but does not consider the relative importance of words in different documents. Additionally, the number of features was limited to the top **10,000 most frequent words** using the **max_features** parameter to optimize performance. For the remaining models, **TF-IDF Vectorizer** was applied to capture not only the frequency of terms but also their importance within each document relative to the entire corpus. This approach assigns a weight to each word based on its term frequency in a specific document and its inverse document frequency (we could call it rarity) across all documents, in doing so, it highlights significant terms while reducing the impact of common words. Both vectorizers were configured with a **maximum document frequency (max_df) of 70%**, meaning words appearing in more than 70% of the documents were ignored, as they were unlikely to provide useful information for genre classification for being too common and non-informative.

2. EXPERIMENTAL SETUP AND RESULTS
2.1 Experimental Setup
The experimental setup aimed to evaluate the effectiveness of different machine learning models for classifying movie plots by genre. A labeled training dataset with information on movie titles, genres, and plots was used for model training. Additionally, a test set without labels was used for generating predictions, while a development set, created by splitting the data into **90% training and 10% validation**, allowed for intermediate evaluation. Both datasets underwent the same preprocessing steps described in 1.2, the text data was then vectorized and the three models were tested. Finally, the confusion matrix for each model was calculated, along with accuracies per genre and model.

2.2 Results
Naive Bayes ended with an accuracy of **66.46%**, while SVC showed the most promise with **68.45%** and Logistic Regression with **65.34%**. To further understand each model's performance, a confusion matrix was generated for each approach (Figure 1).

These matrices provide a detailed breakdown of how often each genre was correctly predicted and where the model made errors by confusing one genre with another. From these matrices, we can see that each model displayed different advantages and disadvantages, performing better in different genres. All models perform well in horror and western, but while SVC and Logistic Regression struggle with sci-fi, crime and romance, these perform much better in drama and horror when compared to Naive Bayes.

Since the models vary in strengths, a multimodal approach was hypothesized, merging the predictions of all three models based on their accuracy at predicting each genre. By calculating its **confusion matrix** (Figure 1) we can see that its performance appears more balanced across genres.

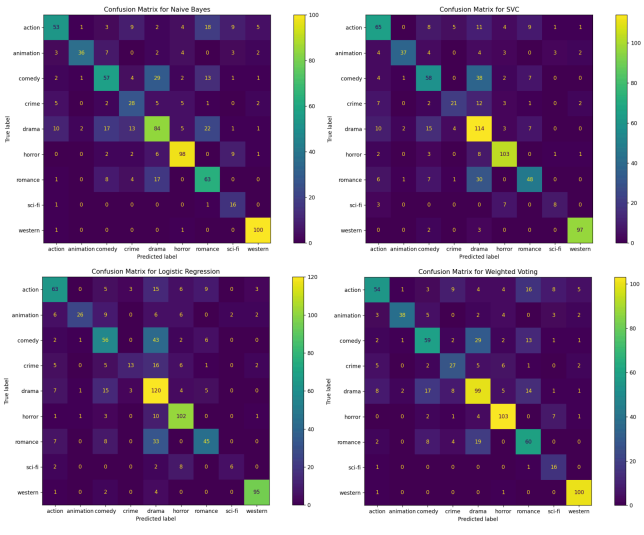

Figure 1. Confusion Matrices

Our multimodal process consists of two key concepts: **Ensemble learning** and **Weighted voting**. The core idea of ensemble learning is that different models will make different types of errors, and by merging their predictions in parallel we can leverage the strengths of multiple models to make more accurate predictions than any individual model could on its own, mitigating those errors. The weighted voting system further refines this by combining models' probability scores with **accuracy-based weights** based on their ability to predict specific genres. In this system, models that perform better in certain genres have more influence over the final prediction in those categories. This allows the strengths of each model to be maximized, while their weaknesses are minimized. For example, SVC's strong performance in drama and horror gave it more influence in those genres, while Naive Bayes, despite its overall lower accuracy, contributed more weight in sci-fi and romance. A key factor in our approach is the **probabilistic interpretation** of predictions. Instead of hard classifications, models output probability scores representing their confidence in each genre. By combining these probabilities with genre-specific accuracy weights, the weighted voting system aggregates **confidence-weighted predictions**. This ensures more informed decisions, reducing extreme misclassifications in uncertain genres.

This approach turned out to achieve the highest test accuracy with **69.32%**. To further illustrate the differences between the approaches, we plotted the genre-wise accuracies (Figure 2), which shows the more balanced performance achieved with the multimodal system. While individual models tended to excel in certain genres but falter in others, the ensemble method reduced extreme misclassifications, creating a more reliable and robust prediction system overall.

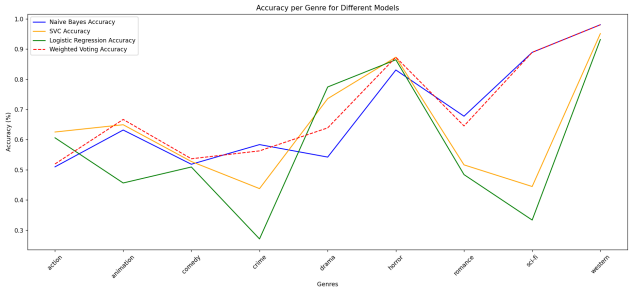

Figure 2. Overall accuracy graphic

3. DISCUSSION

In this section, we analyze the errors made by the model and examine the underlying reasons for the misclassifications. While the overall performance of the model was satisfactory, several patterns of errors emerged, particularly among closely related genres. The analysis below focuses on common error patterns, providing examples to illustrate the challenges faced by the model.

3.1 Common Error Patterns

One of the most prominent sources of error came from the misclassification between drama and romance genres and between drama and comedy. For example, "Intimate Relations", a British drama, was classified as a comedy. On the other hand, "The Angel Who Pawned Her Harp", a British comedy, was classified as a drama. This is likely due to similarities between these genres, as both pairs often share overlapping themes or emotional tones. For example, many romantic films incorporate dramatic elements, while comedies often blend humor with serious situations found in dramas. This overlap likely confuses the model, making it difficult to distinguish between genres with subtle differences in tone or plot progression. The romance genre is often confused with drama, comedy, and action due to genre overlap. Emotional depth links it with drama, while romantic comedies blend humor with romance. In action films, romantic subplots can lead the model to prioritize action elements over romance. Both SVC and Logistic Regression struggle with sci-fi, romance, and crime. On the other hand, Naive Bayes frequently misclassified drama, which significantly impacts its overall performance, as drama is a dominant genre in the dataset, worsening its accuracy.

4. FUTURE WORK

While our preprocessing pipeline already uses several techniques, feature extraction could be used for semantic understanding, or perhaps we could explore additional methods such as synonym replacement, and experimenting with domain-specific stopword lists or more sophisticated lemmatization algorithms could refine text inputs and further enhance performance. Additionally, the multi-model approach could be refined by experimenting with other classifiers, such as Random Forest, Gradient Boosting, or even Convolutional Neural Networks (CNNs). The goal would be to identify classifiers that excel where the current models underperform, potentially improving overall classification accuracy. On top of that, data augmentation could diversify the training set and improve model generalization.

5. BIBLIOGRAPHY

Jurafsky, D., & Martin, J. H. (2020). Speech and Language Processing (3rd ed.). Pearson.

Dong, X., Yu, Z., Cao, W., Shi, Y., & Ma, Q. (2019). A survey on ensemble learning. Higher Education Press and Springer-Verlag GmbH Germany, part of Springer Nature.