Classification of movies genres
Machine Learning for Natural Language Processing 2022

Chehouri Lea
3A DSSA
lea.chehouri@ensae.fr

Froment Theophile
3A DSSS
theophile.froment@ensae.fr

1 Problem Framing

The purpose of this project is to predict the genre of a movie thanks to multiple variables present in our database. We were interested in the database, *The movies dataset IMB*, that we retrieved from kaggle. We are in the context of a supervised problem, a classification problem, where the different classes correspond to the possible genre of a movie. Our database contains 45466 movies, and 20 genres. We have numerical, categorical and textual variables.

2 Descriptive analysis and observations

2.1 Target variable : genre

The classes are strongly unbalanced: the most frequent genre, Drama, is present in 45% of the films (Fig.1). Some genres are much rarer like TV movies, and therefore, will be much harder to predict.

Moreover, a movie can have up to 6 genres assigned, but this is very rare (Fig.2). In general, movies have between 1 and 3 genres assigned. This information is interesting because if a movie can have several genres one can wonder if a subset of genres can describe the whole database. This would be useful as too many classes would be quite impractical to get satisfactory results for classification task. We found that by keeping only the 8 most frequent genres we can describe 90% of the database (Fig.3).

Another interesting aspect to study is the proximity between the genres: for a given genre, which are the other genres that are often associated with it. This could potentially allow the grouping of genres, and thus reduce the number of classes to be predicted. Our analysis showed that it is not relevant to group genres (Fig.4).

2.2 Textual variable : overview

To clean this variable (to make it usable by a model), we first checked if all the overviews were in english. Then, we lower the text, removed links, numbers, stop words, punctuations, words with less than two characters, we normalized white spaces, corrected spelling mistakes, relax the contractions, lemmatized, and tokenized.

The average length of the overviews before cleaning is 323 words, while after cleaning it is 220 words. We have calculated word clouds by genre and we observe that they allow to easily distinguish the different genres (there are really words specific to genres). This is very interesting because it justifies the use of a bag of words to encode our overviews, as a baseline model (fig.6,7). The zipf law computed on the text cleaned shows that only 30% of the vocabulary words represent 95% of the words in the corpus (Fig.5).

After cleaning the base, we have retained 93% of the base. By keeping only the movies assigned to the 8 most frequent genres, we keep 87% of the database.

3 Experiments Protocol

The challenges of our project are multiple:

- combining all types of variables in order to best predict the genre(s) of a given film.
- Manage the multiclassification task on 20 classes, single-label and multi-label.
- Manage unbalanced classes.
- Find embedding methods and efficient models for our task.

First, we will try to find efficient models for a **single-label classification task** (for a given movie, we predict the most frequent associated genre, among the eight preserved ones). We will test three types of models:

- Baseline model: a random forest with different embedding techniques: bag of word, and glove embeddings with different aggregation techniques: simple average, or tf-idf average.
- Deep learning model: LSTM. We will test an LSTM with embedding layer, and an LSTM with fastext embeddings.
- Pre-trained model: DistilBertForSequenceClassification from HuggingFace.

In a second step, we will try to do **multi-label classification**. Indeed, initially, several genres are attributed per film. Moreover, we can think that by keeping only one genre per movie, we can lose information potentially useful for genre prediction. We will test:

- A One-vs-Rest classifier on a Random forest.
- An LSTM adapted to multi-label classification.

Finally, the classes being very unbalanced, we will perform **data augmentation**.

4 Results

4.1 Single-label classification

The baseline models of the random forest with bag of words and glove, give the same performances : an accuracy of 51%, but with a model that predicts Drama all the time (92% of the predictions for bag of words are Drama) (Fig.8,9). For the LSTMs, we see that the accuracy with Fastext is better, with 60% accuracy (against 52% accuracy for the LSTM with embedding layer). The LSTM predicts better than the baseline models in the sense that the predictions are more varied: it predicts Drama Comedy and Documentary well. However, we see that the LSTM with Fasttext overfits much less than the one with the embedding layer, even though we had built the LSTM to limit the overfitting : drop-out, dimension reduction, weighting of the loss by the size of the class. (Fig.9,10). For the BERT model, we succeeded, by training our model on 1 epoch, to obtain an accuracy of 65%. Bert's performance is quite disappointing, as the accuracy is good, but he only predicts two classes well. The main conclusion of these models is that we will not be able to improve the performance as long as the classes are not a reequilibrated.

With CPU, randoms forest train in a few minutes, while LSTMs take 40-50 minutes, and BERT up to 3 hours for an epoch. So there is a real compromise between time and performance.

4.2 Multi-label classification

For multi-label classification, it seems more relevant to us to look at recall and accuracy per class to evaluate our models. For the one-vs-rest classifier, the precisions are correct (above 60%) for all genres except Horror, Crime and Romance, for which the model never predicts. The recalls are bad except for Drama. The LSTM seems to perform better: precisions and recalls are between 50% and 75% for all classes. However the model seems to overfits. Finally the 'micro average' accuracy for the LSTM is 57% against 31% for the one-vs-rest (Fig.11,12).

Note that here we are working on films where at least one of the genres is among the eight most frequent. The performances are much less good when we do not filter the films, that is to say when we consider all the genres and all the films.

4.3 Data augmentation

At this stage, what could greatly improve our model is data augmentation: indeed, the least frequent classes have only about 200 movies. We found that the most relevant method is data augmentation by back translation: we translate the text into French (Spanish, and German), then we translate it back into English. The tests we conducted showed us that this method is very efficient. Because of time constraints, we simply tested to increase all our classes to 3000 movies per class (35% of the full dataset), and we trained it on our best model so far: Lstm with Fastext. The performance is very satisfactory : the performance of small classes are much better (all classes are predicted with a correct accuracy), predictions are much more diversified. The overall performance is quite good (57% accuracy) and could be even better if we use the whole dataset. Thus, one can expect very good performance by increasing the base further more (Fig.13,14,15).

5 Conclusion

In the context of single label classification, we find that the best model taking into account the computation time and performance obtained is the LSTM with Fastext. The performances are strongly improved by performing data augmentation. On the multi-label classification task, the LSTM's the performances are quite good considering the difficulty of the task.

References

1. https://towardsdatascience.com/journey-to-the-center-of-multi-label-classification-384c40229bff

2. https://medium.com/towards-data-science/data-augmentation-in-nlp-using-back-translation-with-marianmt-a8939dfea50a

3. **Shivampanwar.** 2019. https://github.com/Shivampanwar/Bert-text-classification

4. **Chris McCormick.** *"BERT Fine-Tuning Tutorial with PyTorch"*. 2019. https://mccormickml.com/2019/07/22/BERT-fine-tuning/.

5. **Francois Chaubard, Rohit Mundra, and Richard Socher.** *"Deep Learning for NLP Part-I"*. 2016.

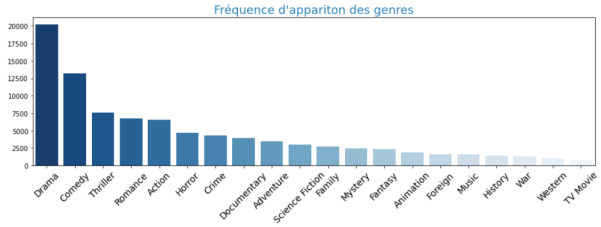

Figure 1: Frequency of occurrence of genres

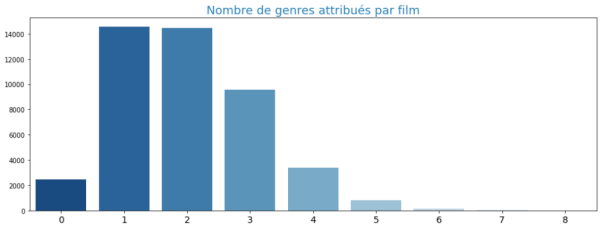

Figure 2: Number of genres assigned per movie

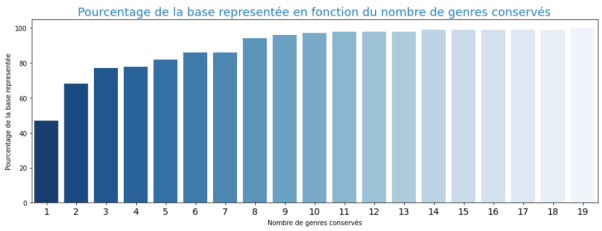

Figure 3: Percentage of number of films retained by number of genres retained

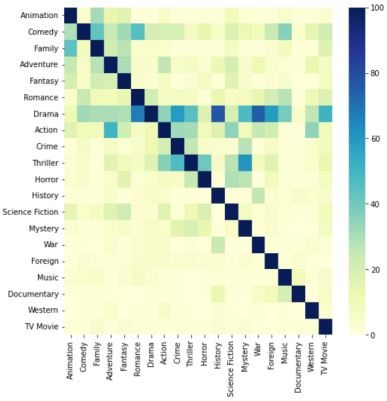

Figure 4: Proximity between genres

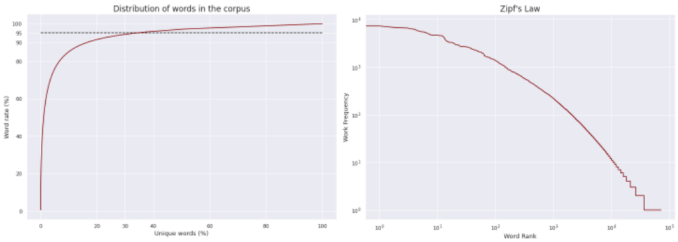

Figure 5: Zipf law

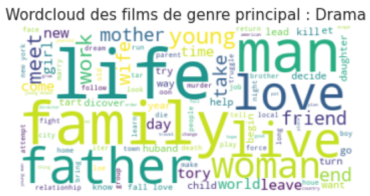

Figure 6: WordCloud : Drama

Figure 7: WordCloud : Thriller

	precision	recall	f1-score	support
Action	0.29	0.01	0.01	352
Comedy	0.39	0.10	0.16	1702
Crime	0.00	0.00	0.00	46
Documentary	0.63	0.10	0.18	678
Drama	0.52	0.97	0.68	3938
Horror	1.00	0.00	0.01	326
Romance	0.00	0.00	0.00	83
Thriller	0.57	0.01	0.01	737
accuracy			0.52	7862
macro avg	0.43	0.15	0.13	7862
weighted avg	0.51	0.52	0.39	7862

Figure 8: Random Forest + Bag of Words - Classification report

Figure 9: Single label classif - LSTM FASTEXT - Confusion Matrix

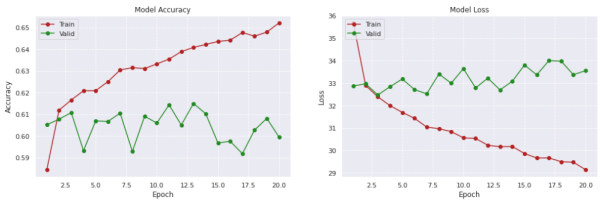

Figure 10: Single label classif - LSTM FASTEXT - Loss et accuracy plot over 20 epochs

	precision	recall	f1-score	support
Comedy	0.56	0.62	0.59	2476
Drama	0.66	0.64	0.65	3916
Thriller	0.47	0.41	0.44	1524
Action	0.54	0.44	0.49	1309
Horror	0.65	0.54	0.59	934
Documentary	0.72	0.54	0.62	781
Crime	0.44	0.26	0.33	813
Romance	0.42	0.36	0.39	1317
micro avg	0.57	0.53	0.55	13070
macro avg	0.56	0.48	0.51	13070
weighted avg	0.57	0.53	0.54	13070
samples avg	0.57	0.57	0.54	13070

Figure 11: Multi-label classif - LSTM - classification report

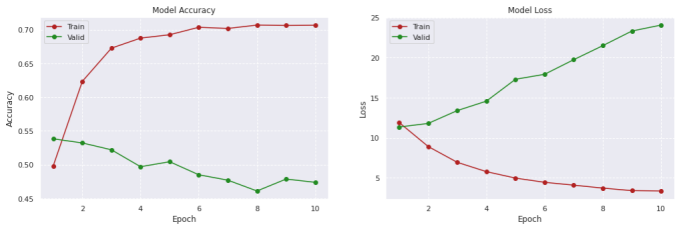

Figure 12: Multi-label classif - LSTM - loss and accuracy plot

	precision	recall	f1-score	support
1	0.43	0.52	0.47	329
2	0.83	0.71	0.77	600
3	0.47	0.24	0.32	600
4	0.36	0.41	0.38	600
5	0.42	0.56	0.48	600
6	0.67	0.56	0.61	636
7	0.75	0.98	0.85	274
8	0.78	0.87	0.82	349
accuracy			0.57	3988
macro avg	0.59	0.61	0.59	3988
weighted avg	0.58	0.57	0.56	3988

Figure 13: Data augmented dataset (35% of full dataset) : classification report

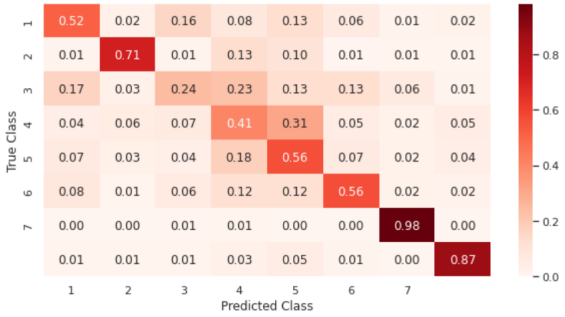

Figure 14: Data augmented dataset (35% of full dataset) : confusion matrix

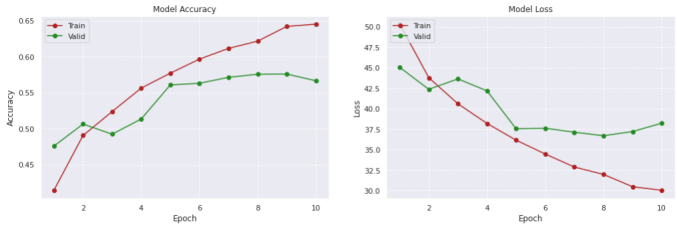

Figure 15: Data augmented dataset (35% of full dataset) : loss and accuracy plot