# *Sentiment Classification using BERT*
# Machine Learning for Natural Language Processing 2020

**Yannis Airouche**
Ensae
yannis.airouche@ensae.fr

**Paul Braulotte**
Ensae
paul.braulotte@ensae.fr

## 1 Problem Framing

The aim of this project is to build a model capable of deducing from a movie review if this movie is worth watching or not. Indeed, as the importance of online critics increases, one should focus on the sentiment given by the comment itself and not on the given rating which could be misleading to the reader. In fact, people tend to be less accurate in their rating as in their thoughts. For instance, some people are excessive in their scoring as they would give at least 8/10 to every movie they liked and less than 3 to the rest. On the other hand, some others wouldn't give a score outside a 4 to 6 interval.
By expressing their thoughts in a more argued way, these critics are more nuanced and the information we could retrieve from it would therefore be more accurate in order to choose the right movie.

Our study will be based on reviews from the famous Internet Movie Database (IMDb). This website being in the top 100 of the most visited websites in the world, we can assume that the 50000 reviews we collected from it, will be representative.

## 2 Experiments Protocol

After some initial exploration of our database, we will train two models. The fist one will try to learn how to classify movies as "good" or "bad" depending on its review. The second model will also be a classification model but with way more precision as it will estimate the rating given by the review's author. This rating goes from 1 to 10, 10 being an awesome movie and 1 an awful one.
It is important to note here that the only features contained in the dataset are the reviews. There is no feature for the movie, the producer, the main actors or the release year for instance, therefore preventing the model from "learning" the successful movies, producers or actors and estimating the scores according to them.
In order to maximize our results, we decided to use Google's BERT transformer, pre-trained on the BooksCorpus and Wikipedia, and that we will fine-tune to render it more adapted to movie critics language. We added to this encoder a linear layer to be able to make classification.

Quite naturally, we will have first split our database in three different samples for training, validation (to hyper-tune the model's parameters during training) and testing, in order to evaluate our models on an independent subset from the one used during training. Furthermore, this evaluation will be made through several metrics: a quite common confusion matrix which will give us a first picture of the performances; linked to this matrix, we will produce a classification report presenting the precision, recall and f1-score for each label as well as their averages and the global accuracy of the model. A more peculiar metric we will use for the binary classification is the Matthew's correlation coefficient, often described as the best way of describing an unbalanced confusion matrix with just one number as it is a kind of correlation between the predicted and true labels, +1 (resp. -1) meaning that the classifier is perfect (resp. terrible).

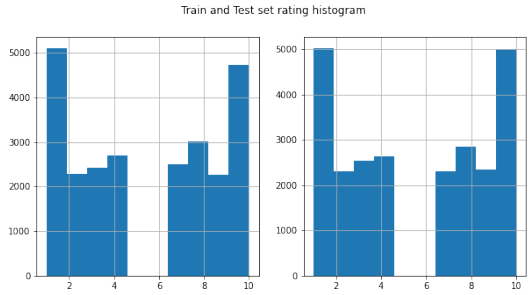
$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (1)$$

Finally, in order to obtain satisfactory results from our models, the reviews put as inputs will have been cleaned beforehand. By cleaning, we here mean the removal of HTML tags, punctuation and multiple spaces, so that there will be no misinterpretation by the tokenizer and the models.

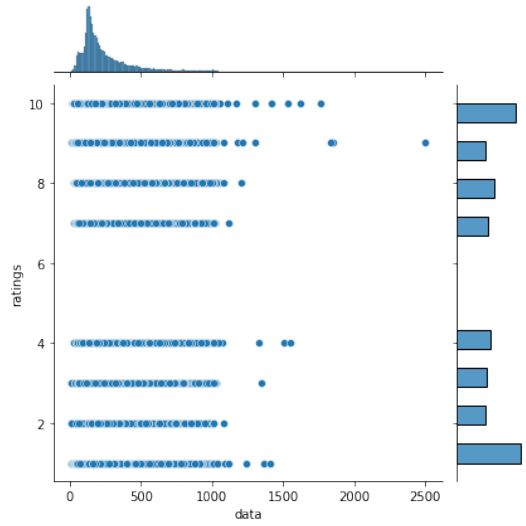
## 3 Results

We talked earlier about the importance of focusing on reviews instead of ratings to gain more precision. This phenomenon is clearly visible in both our training and testing datasets as we can see that critics clearly exacerbate their opinions when scoring a movie, the labels 1 and 10 being clearly predominant. It is noticeable that there are no "average" movies

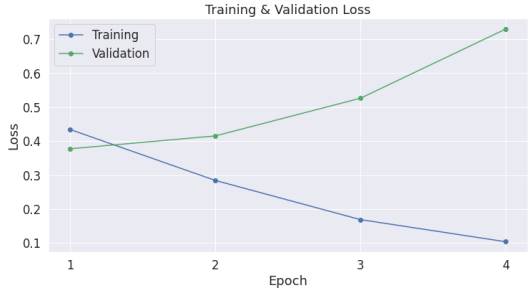
in our database as labels 5 and 6 have been ignored. During our exploration, we asked ourselves if there



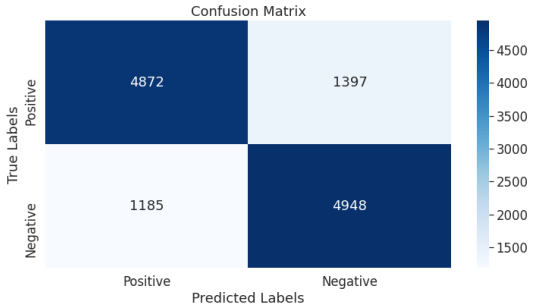

could be a link between the length of a review, represented by the number of tokens, and its associated rating, as one might think that people enthusiastic about a movie might have more to write about it than others. In fact, we can see on the next figure that there is no clear dependence structure. Once again, our model won't be able to "cheat" through this trick. We first tried to fine-tune our sentiment model with




4 epochs but observed that this would surely lead to some overfitting as, while the training set's loss was decreasing, the validation one kept climbing, as we can see beneath. That's why we chose that the best trade-off would be 2 epochs of training. The

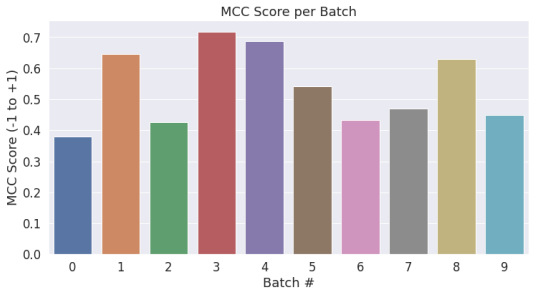



results we obtained on the testing sample are quite satisfactory as our confusion matrix tends to be close to the identity matrix, with few false positives and false negatives. Given that the database is nearly perfectly balanced between good and bad reviews, we can focus here on the accuracy of our model, which is of 0.7918 on the testing dataset, meaning that 4 times out of 5, our model predicts correctly if a movie is worth watching or not. Finally, the




| Classification Report: | precision | recall | f1-score | support |
|---|---|---|---|---|
| 1 | 0.8044 | 0.7772 | 0.7905 | 6269 |
| 0 | 0.7798 | 0.8068 | 0.7931 | 6133 |
| accuracy | | | 0.7918 | 12402 |
| macro avg | 0.7921 | 0.7920 | 0.7918 | 12402 |
| weighted avg | 0.7922 | 0.7918 | 0.7918 | 12402 |

Matthew's correlation coefficient shown below by batch, is clearly a sign of success for our model because not only all its values are positive but also mostly above 0,5.




## References

https://colab.research.google.com/drive/1eQtcEC_-AqklJTbK4CBPbxizSqamDybZJ?usp=sharing
https://github.com/FlyingDutchman97/ENSAE_-ML_NLP_AIROUCHE_BRAULOTTE