ML Scientist Internship - Technical Challenge

27th October 2022

AUTHORSHIP

Name: João Tiago Almeida; email: joaotiago99@gmail.com

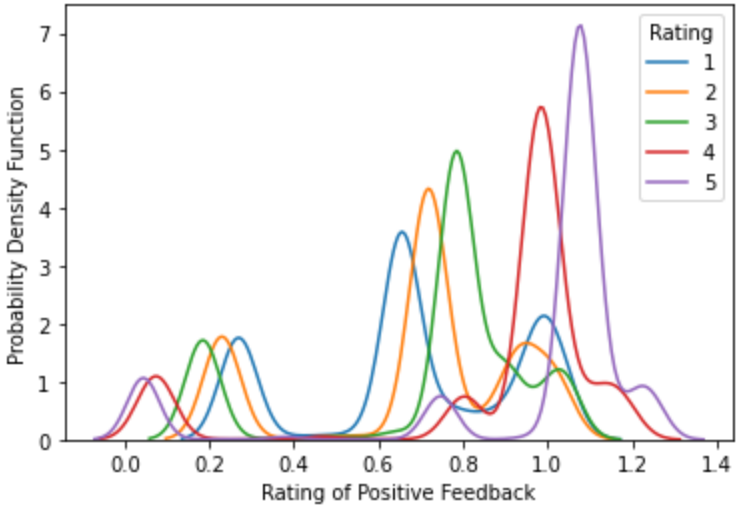
GOALS

1. Data analysis of a dataset regarding Amazon products' review;
2. Modelation of a Classifier to predict the helpfulness of a new Review.

Analysis

Question: Is there a correlation between the rating of the product and the helpfulness of the review?

Yes, there is a correlation between the rating of the product and its helpfulness. However, the relation is not crystal clear, since we have to manipulate the data at least two times. To prove the correlation between both variables I had to calculate the *rate of positive feedback*: $\frac{feedcak[yes]}{feedbcak[total]}$, and could only consider reviews with at least one rating. Additionally, the higher the rating of the product, the higher the number of feedback, therefore the data is not balanced. So, it is used a non-parametric statistical test between a categorical and continuous variable: Kruskal and a *posthoc* Dunn's test. We confirm statistical independence ($p<0.05$) between all conditions, proving the correlation between both variables.



Question: Who are the most helpful reviewers?

The computation of this metric is divided into two steps.

1. **Relevance of the review on the product**

Since the dataset is not balanced between products of different ratings, an absolute metric would overrate higher ratings and would be biased toward reviewers who reviewed more these products. The applied formula collects the impact of the review based on all reviews made. Therefore weighs how a review stands out over all reviews of a specific product. So, for a specific product p, the formula is

$$\frac{\#(feedback[yes)_p}{\#(reviews)_p}, \; with \; \#(x)_p \; being \; the \; total \; number \; of \; < x > \; related \; to \; product \; < p >.$$

At this point, I have the impact of each review (column: **helpful_yes_per_product**). With these results, we can call the **helpful_reviews** when for reviews in which this metric is equal to or higher than one. Likewise, it is called **unhelpful_reviews** for reviews for which this value is less than 1 but higher than 0. These reviews are considered labeled and the is unlabeled (result equal to 0).

2. **Balance between all reviews of each reviewer.**

At this stage, I evaluated the earlier computation to consider also the negative feedback. Therefore I multiplied the previous value by its correspondent *rate of positive feedback*. This step was not performed earlier because it does not look into the context of the review on all reviews. The result is the column **help_weight**.

Finally, for each reviewer, I used three different metrics to balance their influence on the platform, on the **help_weight** values.

a. **mean** - The mean of all values gives the consistency of the reviewer;

b. **max** - The max value gives the top peek of their most helpful review;

c. **sum** - Their effort of all comments (as the expression goes _step by step ..._).

Finally, the top 10 is composed by:

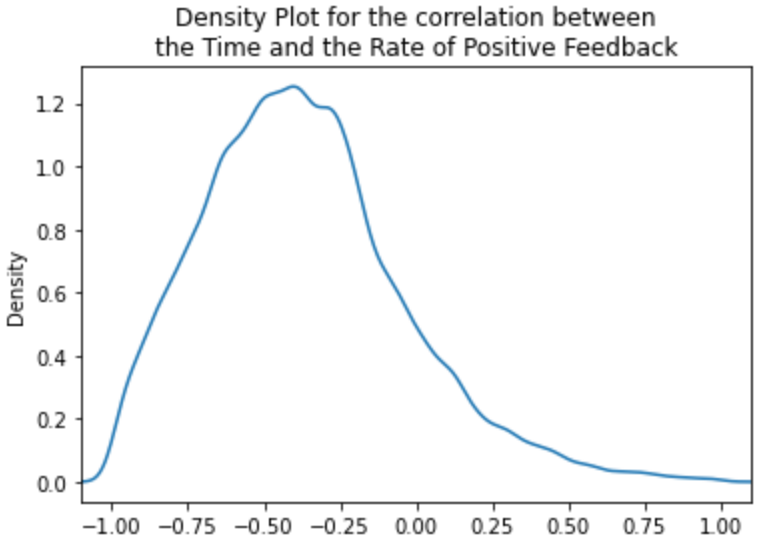
```
Top 10 of the Most Helpful Reviewers:
1st: MariaSmiles
2nd: Dirk J. Willard "Dirk Willard"
3rd: Andrew Conley "Andrew"
4th: Sandra Hender "Master Writer and Poet"
5th: C
6th: Joanna Daneman
7th: Tallgirl77
8th: Gadget Girl
9th: Michael G. Lustig
10th: Matthew G. Sherwin
```

Question: Have reviews been getting more or less helpful over time?

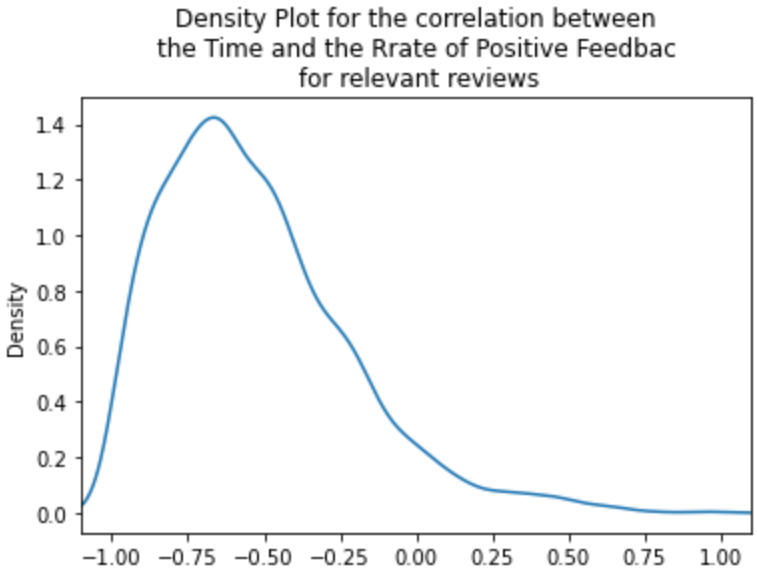
Reviews have been getting less helpful over time. To support this answer, I ran a correlation between the time of a posted review and its positive rate of positive feedback for each product. However, this metric is not the most accurate, since treating the rate of positive feedback is constant every time (contrarily to the question "...over time...").

Nevertheless, the results show that the correlation between the time of the post and the review's rate of positive feedback is a medium negative correlation. To show graphically our results, I plotted 3 figures showing:

1. The density of the correlation between the time and the rate of positive feedback, for comments with at least one feedback.

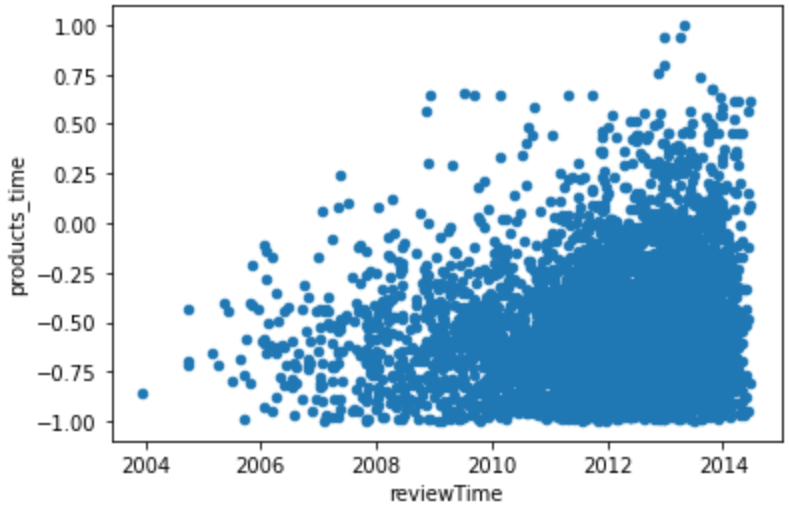


2. The density of the correlation between the time and the rate of positive feedback, for comments with a higher number of positive feedback than comments on all products.



Comparing the two previous analyses, when using only the relevant most relevant comments, the mean and the standard deviation decreased, showing more data gathering around a higher negative correlation.

3. The scatter plot shows that more and more reviews on products have been made over time, although they have not gotten better over time. The y-axis reflects the correlation between the time and the helpfulness of the review.



Modeling

After someone writes a review, will it be considered helpful by other users?

After some attempts o bringing together to process sentences and merge the outputs with LSTMs, the results turned out to be barely better than the 50/50 threshold mark, which I considered to be random. However, the second attempt consisted of using a **Random Forest classifier**, which was already tested on similar datasets of Amazon reviews [1].

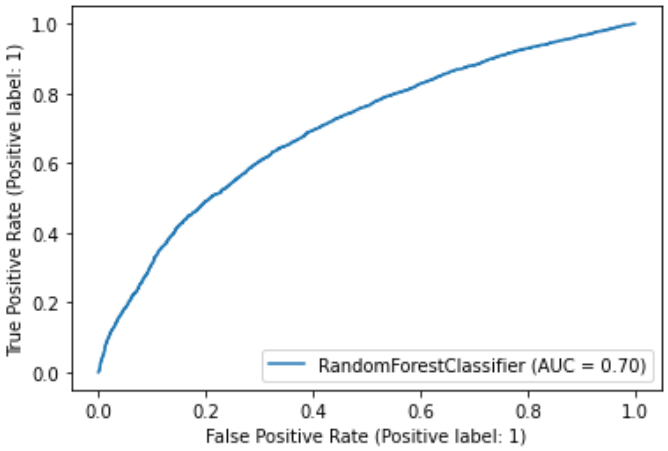
I used the previous helpful/unhelpful review classification to label the data corresponding to 2.2% and 25%, respectively. Thus resulted on 72.8% of the data is unlabeled. In the training process, I balanced the dataset between the helpful and unhelpful reviews, sampling unhelpful reviews randomly to make the same number of helpful reviews. Finally, I trained the model with 80% evaluated with 20% of the data.

I used the scikit-learn *RandomForestClassifier* to build the classifier and its predictions with the default parameters as the baseline values. Then, I run a RandomSearch varying some parameters (n_estimators, min_samples_split, min_samples_leaf, max_depth), resulting in the final model parameters of (n_estimators=1250, min_samples_split=2, min_samples_leaf=2, max_depth=5), respectively. To avoid overfitting the model, it was compared the prediction values on the training data as well as on the 20% untouchable data until this point. The results are the following:

	Accuracy	Recall	Precision	F1-score
Baseline - Test	0.644	0.625	0.647	0.636
Final model - Train	0.697	0.672	0.715	0.693
Final model - Test	0.653	0.632	0.658	0.645

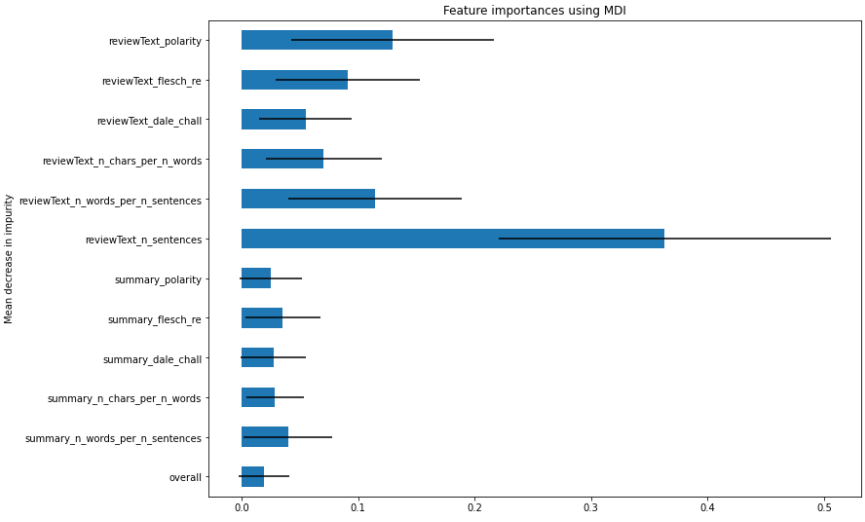
Finally, the final results are slightly better than the baselines, and not to distance from the train concluding that the model is not overfitted to the training data.

Additionally, I also took into consideration the confusion matrix (normalized) and the ROC/AUC of the model. In this case, it is particularly relevant taking into consideration the *false negative* or type II error, to decrease the probability of misclassification of a helpful comment.



	Predicted Label (Unhelpful/Helpful)	
True Labels (Unhelpful/Helpful)	0.674	0.326
	0.368	0.632

The model used 12 features in total: Overall (rating of the product by the reviewer), and then the number of sentences, number of words per sentence and characters per word, the Dale Chall readability score and the Flesh reading ease test, for the text in the review and summary separately. Note that it was not used the number of sentences in the summary as a feature of the final model since it showed little to no interference in early predictions of the model.



Additionally, there is a feature called polarity that consisted of analyzing whether the tone of the summary/reviewText was more positive or negative. For that, I used an open-source document with frequent positive and negative words and summed the ratio of those words in the utterances. After that, I subtracted the negative ratio from the positive. This feature turned out to be the second most important. Furthermore, as expected the features present in the review shapes more the outcome of the model compared to the features in the summary.

All in all, the results are far from being sufficient, and if I had more time I would revise the balance between features, as well as add more text classification. Moreover, this model does not look properly in context, orthography, coherency, or subjectivity, so I could bring it as features. Lastly, I might clean the review's text slightly more, together with past information about the reviewer or the product, bridging the gap with the information on the previous analysis of the dataset.

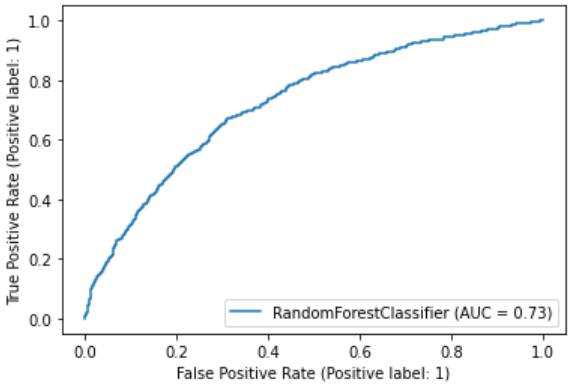
The final step was to label the previous data that could not be taken into consideration because it was unlabeled. The data should be majority unhelpful since most of them were ignored by other reviewers. The results indicate there are 83% unhelpful and 17% helpful reviews among the unlabeled reviews, which seems reasonable.

Transformer models

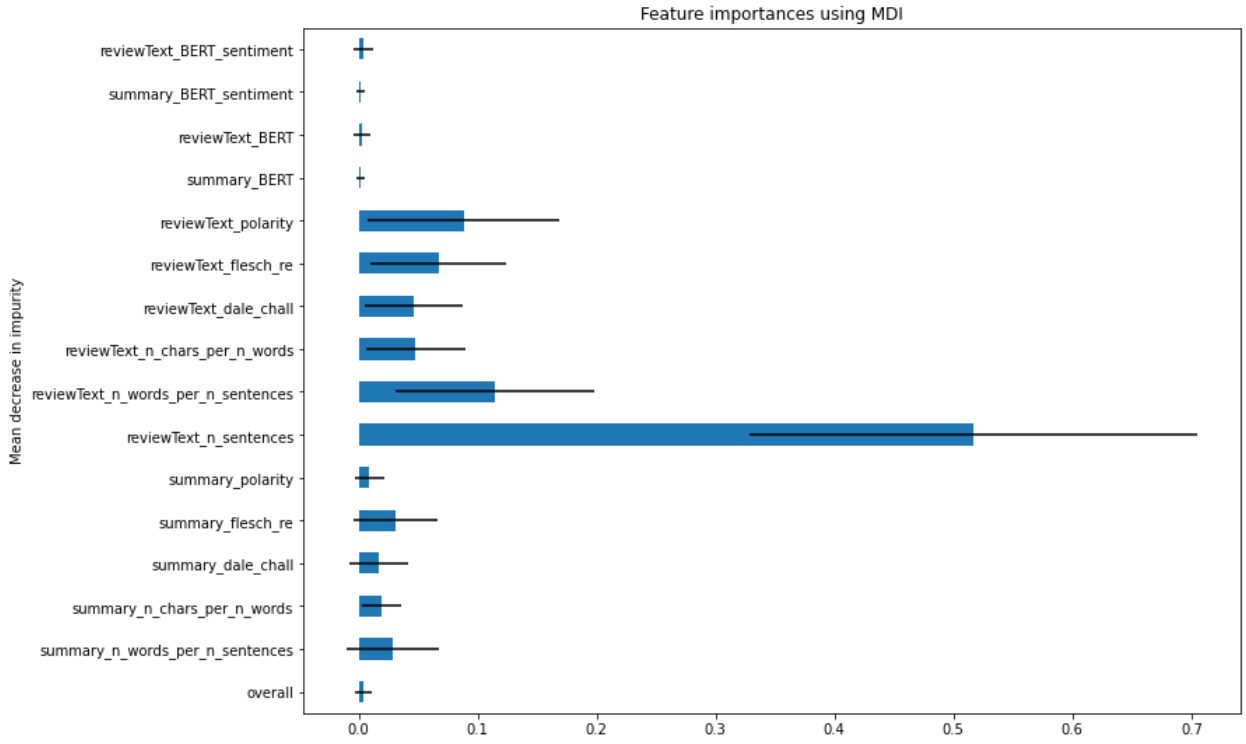
The usage of transformers has the possibility to boost the current work as it looks to the context of the input utterance throughout the word embeddings. With this context, we can associate it with the target value of the classification. As an example, I added some pre-trained BERT models to evaluate the sentiment analysis of both the review and the summary. The BERT models used, both from hugging face 🤗, were: *"finiteautomata/bertweet-base-sentiment-analysis"* with neg/neu/pos and *"nlptown/bert-base-multilingual-uncased-sentiment"* with 1-5 stars rating. Unfortunately, these features were not relevant to the model classification, obtaining the results:

	Accuracy	Recall	Precision	F1-score
Final Model w/ BERT - Test	0.673	0.623	0.685	0.653

	Predicted Label (Unhelpful/Helpful)	
True Labels (Unhelpful/Helpful)	0.721	0.279
	0.377	0.623



The incorporation of the transformers in this model to not reflect their capacities in text classification nor sentiment analysis. Moreover, in a transformer model, I expect it to understand the context mainly due to two things: the positions embeddings, where the information of the sentence word is embedded in the data instead of in the model, and the muti-head self-attention mechanism which takes into consideration the relationship between the other present embeddings. Therefore, compared to other models such as Random Forests, I expect them to be more feasible, and more trustworthy as they are less pruned to be tricked by quantitative measures (e.g. number of sentences, words, characters).



References

[1] Singh, J. P., Irani, S., Rana, N. P., Dwivedi, Y. K., Saumya, S., & Roy, P. K. (2017). Predicting the “helpfulness” of online consumer reviews. *Journal of Business Research*, *70*, 346-355.