KROLIKOWSKI Linda
MONTMEAT Florian
linda.krolikowski@ensae-paristech.fr
florian.montmeat@ensae-paristech.fr
School year 2019-2020

Machine Learning for Natural Language Processing 2020

Analysis of product reviews Alexa - Sentiment study

Report

Reviews

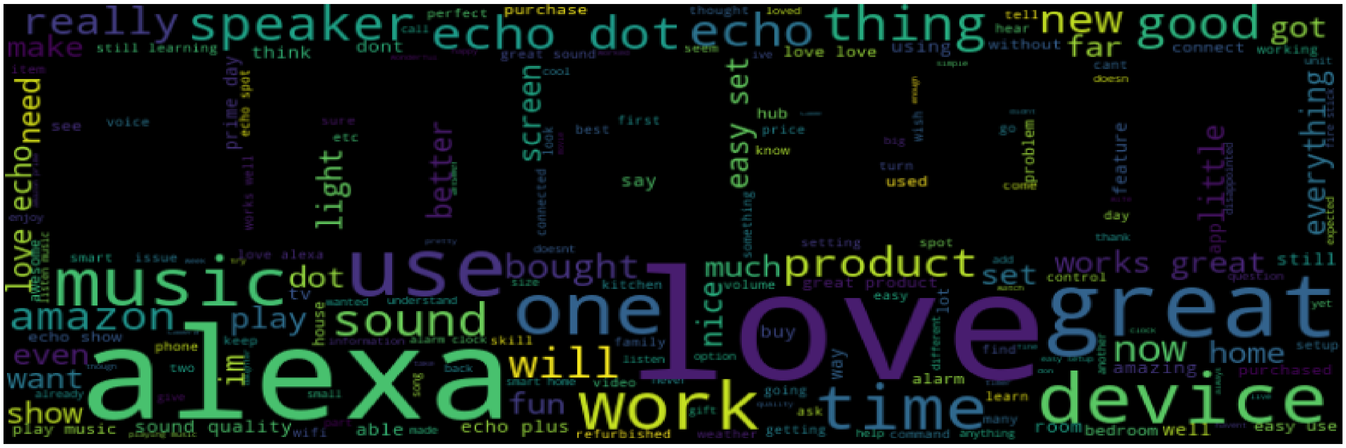


Table des matières

1 Abstract

We are part of amazon's marketing department and we try to understand the needs of consumers regarding the "Alexa" product. We have a database of customer comments on this product by range with the general "feedback" (positive or negative comment) and the rating given (from 1 to 5 stars).

We want to understand the reasons for a negative comment and a positive comment and implement a business plan for a new product launch created from Alexa. Finally, building an automated "feedback" prediction model will later help in the study of new product comments.

The protocol is as follows :

1. We clean our database according to the description in the "corpus cleaning" section in the appendix.
2. We make a first study of the corpus in order to give the intuition of the final results we will find and judge their coherence.
3. We train our models (RF and svm with and without word2vec, tfidf, tuning, smote...) and select those with the best f1 scores (1).
4. We analyze the decision making of the three selected models.Here we judge the coherence of the model, if it can help understanding interesting points about alexa's sells.
5. A final model is selected based on the findings of the previous step.We found that the svm model is the best one.

2 Corpus study and problem framing

2.1 Product details

The study focuses on the "Black Dot" and "Charcoal Fabric" ranges, which have more than 400 comments, rather than ranges such as "Oak Finish" and "Walnut Finish", which have less than 100 comments.

2.2 Base composition and sampling problem

Our database consists of 2893 positive comments versus 257 negative ones, which means that we have a sampling problem. For our baseline study we separated randomly our base into training and development set with the same proportion of negatives and positives as in the initial base. Our training set is constituted of 158 negative reviews (7.5%) and 1952 positive reviews. Our development set is constituted of 99 negative reviews (9.51%) and 941 positive reviews.

Our unbalance problem can be seen in the figure where 1 or 2 star commentators are considered outliers. However, we would like to develop our product by developing its strong points but also by taking into account its defects. To solve this problem :

— We focus on the f1 score associated with the negatives and we can favour a model that degrades the number of badly predicted positives to reduce the number of false positives.

— The under-representation of negatives will be addressed with the SMOTE (2) algorithm, which uses the nearest k neighbors to provide the basis for learning new negative individuals.

2.3 First analyses of customer returns

The words that come up most often in 5-star reviews are : "love", "great", "use", "new", "alexa", "device", "music", "sound", "one" ; and also :"prime day", "good", "work","easy", "dot", "speaker". We can deduce from this, by also reading some comments with these words, that the people who appreciated the product listen to music and Alexa provides a good sound, is easy to use and has a good quality/price ratio (especially at the time of the "prime day").

The most common words used in 1-star comments are : "device", "work", "amazon", "product", "echo", "time" ; and also :"back", "bought", "light", "screen", "refurbished". It can be deduced, also by browsing through some comments with these words, that people who did not like the product inform that the after-sales and delivery service is not up to their expectations and/or because some elements of Alexa work badly or not at all ; others find it useless compared to other amazon products (hence the presence of the word amazon) and we should ask ourselves about the target of this product.

3 Experiments protocol

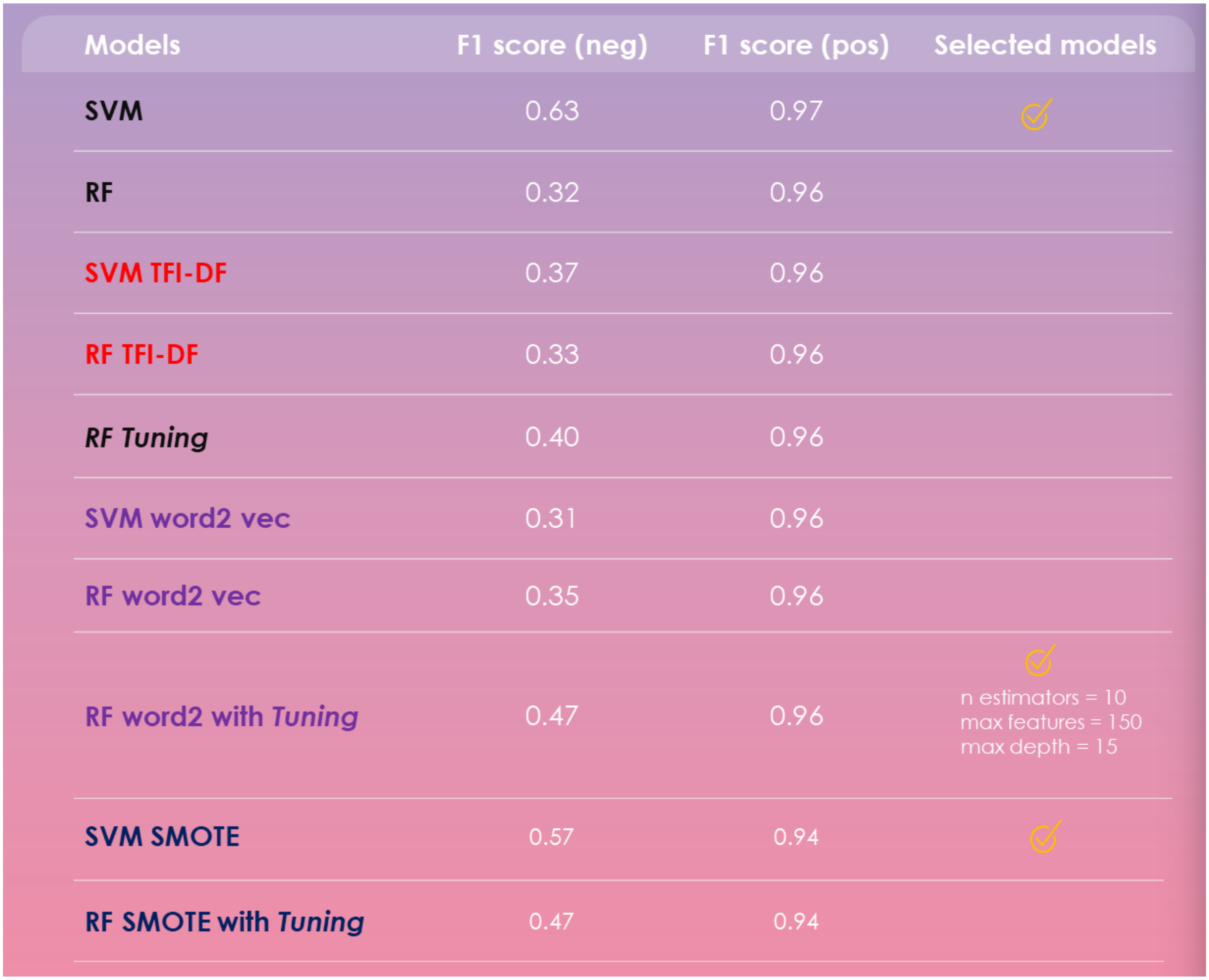
Two models are at the heart of our study, namely the SVM (classification by a hyperplane separator constructed from determining keywords) and random forests (segmentation according to the presence of certain words in the commentary). Around these models, attempts of improvement will be proposed, tf-idf (4) and word2vec (3) :

- Tf-idf gives an importance score to the words in a text.
- In word2vec, the word has a representation in a space that positions it in relation to adjacent words. The links are not between close elements of speech but between elements that are used in the same way. We use a plunge carried out by google which has more important resources at its disposal.

Finally, we're tuning in to random forests so that we can choose our parameters more efficiently.

4 Results

The results are as follows :



Models	F1 score (neg)	F1 score (pos)	Selected models
SVM	0.63	0.97	✓
RF	0.32	0.96	
SVM TFI-DF	0.37	0.96	
RF TFI-DF	0.33	0.96	
RF *Tuning*	0.40	0.96	
SVM word2 vec	0.31	0.96	
RF word2 vec	0.35	0.96	
RF word2 with *Tuning*	0.47	0.96	✓ n estimators = 10 max features = 150 max depth = 15
SVM SMOTE	0.57	0.94	✓
RF SMOTE with *Tuning*	0.47	0.94	

Figure 1 – Models selection

Tuning improves the score of random forests, just like TF-IDF and word2vec. On the other hand, SMOTE causes a big loss on the positive side while performing less well than other models on the negative side. SVM scores best on both sides compared to TF-IDF and word2vec. Although there is a degradation in the positive score, the svm negatives on the re-sampled basis score the second best.

5 Discussion around the three best models

- The Word2Vec + Random Forest model captures words better than the SVM model (this can be observed with good predictions on short comments), but is clearly less efficient on long sentences.
- The SVM model with resampling captures words better than the SVM model without resampling.
- Between Word2Vec + random Forest and the SVM model after resampling, the differences in probabilities are partly observed when the probability of predicting "positive" is high, so we are less interested in the prediction part of negative opinions.

The random forest model with word2with and optimal settings sets a higher importance score for the word "return" to predict that a comment is negative. "Amazon" (the customer compares alexa to other products that are considered more useful) appears in second place. We can guess that "half" intervenes to say that the customer does not use half of the gadgets of the product (uselessness). "Terrible" is also among the first segments. The model is interpretable.

The two best models are the SVM model and the SVM model on a resampled basis. To compare these models we observed the coefficients (see graphs in appendix(5) : 3 and 4) allowing us to calculate the prediction. Thus we notice that the SVM before resampling and after resampling show that positive comments will be predicted thanks to words such as "love" and "great product" in particular. These words seem important to us in understanding a model.
Looking at the words helping to predict negative comments, for SVM models before and after resampling there are "terrible" and "awful", "didn't" (which marks negation). On the other hand some words like "overdriven" or "slow" are present for one of the models but not the other. We therefore decided to choose the SVM model as the final model before re-sampling, as it has a better score and is interpretable.

6 Conclusion

After making a compromise between accuracy in predicting negative and positive feedback, we were able to select 3 models. Then we opted for the one that achieved the best compromise between interpretability and accuracy. None of the three models turned out to be inconsistent. We were able to conclude that our model predicted negative comments with key words such as "intrusive", "terrible" and "awful", and positive comments with "love", "perfect", "great" which are words related to subjective opinions. Other words cited above have allowed us to draw up a list of alexa-related malfunctions. The negative comments can therefore be used by Amazon to improve its product in the future by relying on the words that stand out the most in the negative comments. We could have obtained better results, especially with regard to the word2 model, if we had used a more consistent learning sample at our base.

7 Appendix

7.1 Sampling problem

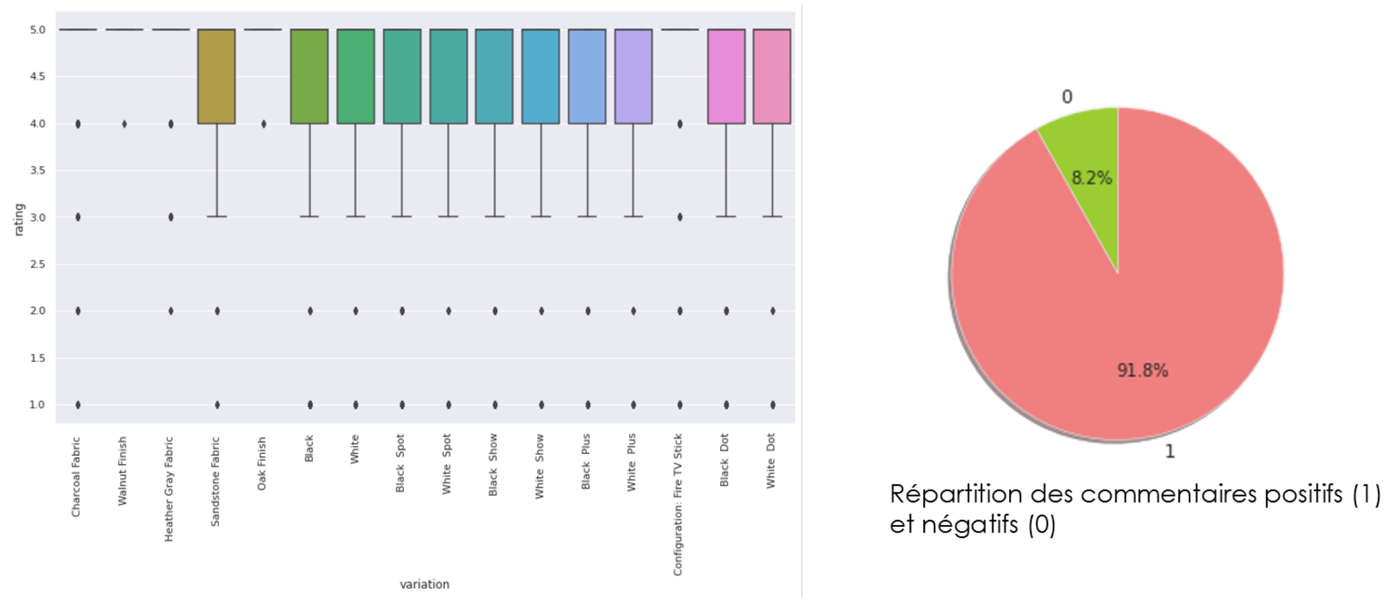


Figure 2 – An understaffing of negatives

7.2 Corpus cleaning

The database is treated as follows :

- We're removing the eishtag, url, beacons.
- Take out the stopwords. ("in", "an", "the") only keep one copy of the duplicate words...
- We choose the TreeBank Tokenizer over the TokTok Tokenizer so we can match words like "prime day"

7.3 Coefficient Analysis

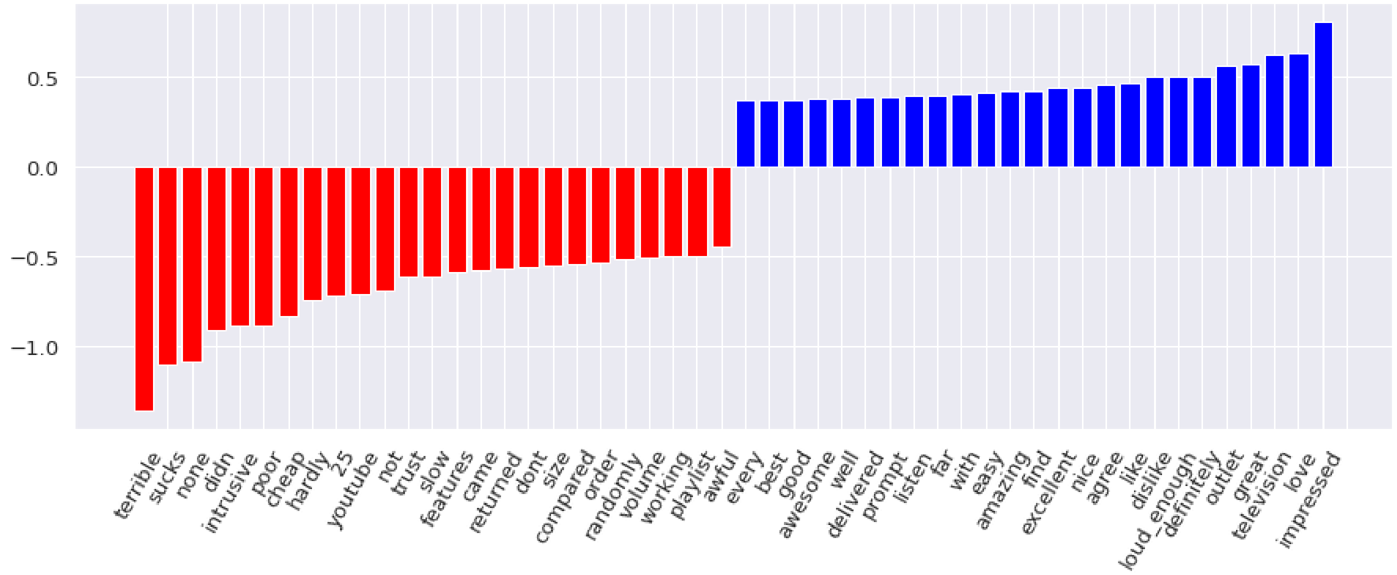


Figure 3 – Prediction Coefficients : SVM

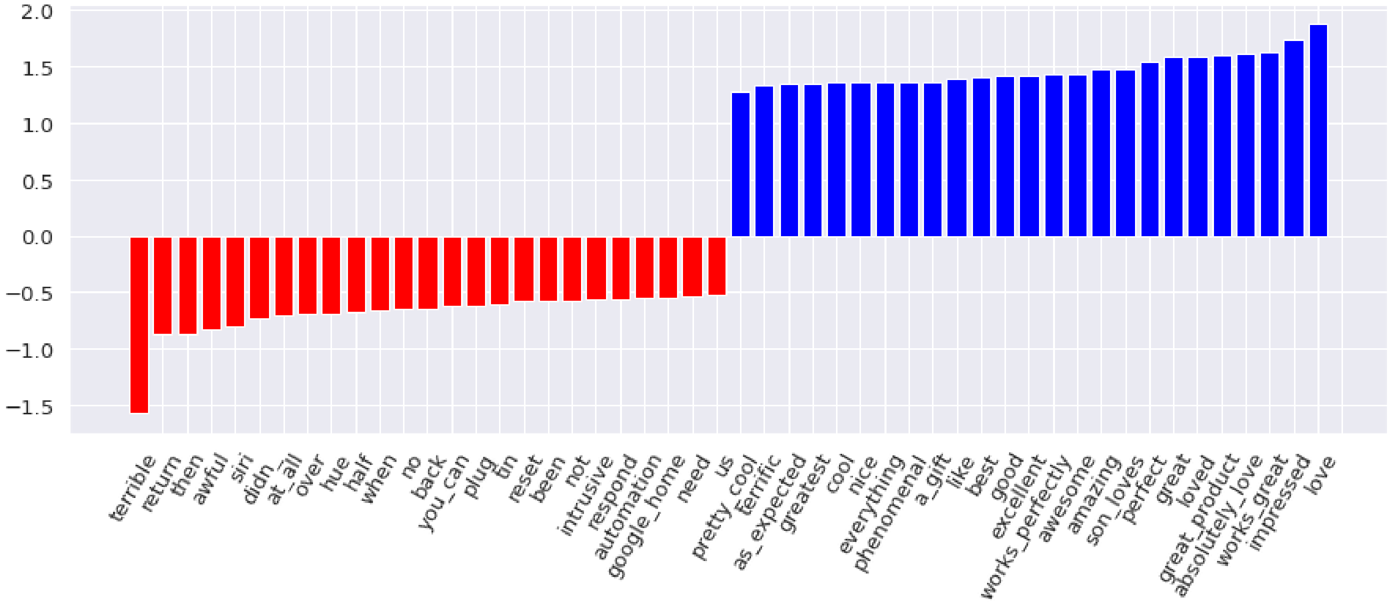


Figure 4 – Prediction Coefficients : SVM + Resampling

Références

[1] https ://en.wikipedia.org/wiki/F1score

[2] Jason Brownlee, *SMOTE Oversampling for Imbalanced Classification with Python*, January 17, 2020

[3] Chris Nicholson, *A Beginner's Guide to Word2Vec and Neural Word Embeddings*, 2019

[4] William Scott, *TF-IDF from scratch in python on real world dataset*, Feb 15, 2019

[5] Aneesha Bakharia, *Visualising Top Features in Linear SVM with Scikit Learn and Matplotlib*, Feb 1, 2016