LN Report MP2

Group 20
Ana Carolina Batista Filipe, 89723 | Gonçalo Filipe Pereira Mateus 93713

The objective of this project is to build a model that classifies reviews of beauty products according to 5 labels: =Poor=, =Unsatisfactory=, =Good=, =VeryGood= and =Excellent=, from the worst rating to the better.

Models and Experimental Setup

Our best model is based on a Support Vector Classifier and a TF-IDF vectorizer implementation which works by proportionally increasing the number of times a word appears in a review but is counterbalanced by the number of reviews in which it is present.

Firstly, we import every single library that is needed to make the model work properly. Then, in the pre-processing phase, we remove every punctuation mark, the numbers and all the duplicate whitespaces, and lowercase words.

Moreover, we remove the English stopwords from the nltk library. We also created a not_remove list so we could decide which words we don't want to remove because we believe that are important for the learning process. The major differences in our efficiency rate happened due to the addition or removal of certain words from this list as we tried to figure out which ones were the most important. We decided to keep all words that express negation or opposition of ideas, e.g., ‘not’, ‘no’ or ‘but’. Because these words invert the sentiment of the review.

To finalize this phase, we use the tokenize and lemmatize functions from the nltk library so we can perform the lemmatization upon every word.

After that, we use TF-IDF vectorizer with unigrams and bi-grams in order to increase the contextual information and avoid underfit our data.

Finally, we built different classifiers such as Support Vector Machines (SVM), Naive Bayes (NB), Stochastic Gradient Descent (SGD), K-Nearest Neighbors (KNN) and Decision Tree (DT). To avoid overfitting, we used k-fold cross validation with k = 10.

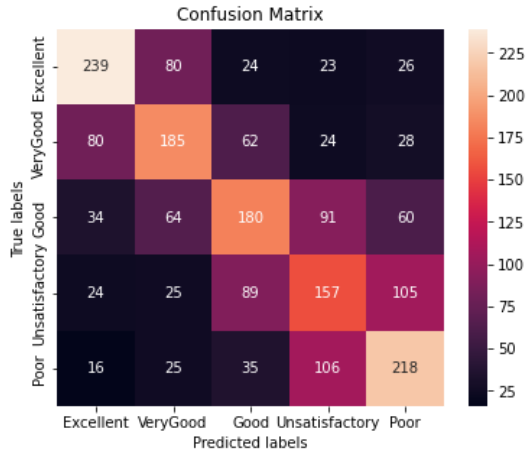
For this work, two baselines were considered. The first one is based on Jaccard similarity. Our second baseline is based on a Support Vector Classifier and a CountVectorizer implementation which uses a tool with the objective of counting how many times a word appears in a certain document. In our case, this will count each word from the beauty products reviews at a time (unigram).

Results

Table 1 contains the classification task results for all the methods developed and the comparison with the two baselines. In Figure 1 we can find the metrics for our top performing model, SVM.

Table 1: Classification results.

	Baseline 1	Baseline 2	SVM	NB	SGD	KNN	DT
Accuracy (%)	36,8	43,0	48,1	45,2	47,8	42,3	44,2



a)

	Precision	Recall	F1-score
Excellent	0.59	0.6	0.59
Very good	0.47	0.47	0.47
Good	0.47	0.41	0.44
Unsatisfactory	0.38	0.39	0.38
Poor	0.51	0.55	0.53

b)

Figure 1: Evaluation metrics for best model developed (SVM): a) Confusion matrix and b) Precision, Recall and F1-score per label.

Discussion

The first aspect to notice is the low accuracy values obtained by all the models. This is due to the inherent difficulty of sentiment analysis tasks. When analyzing the dataset we found several inconsistent assignments, for example: “Huge disappointment” is classified as unsatisfactory, yet the model classifies it as poor, which would be more appropriate given the review. In other cases, the context is not fully provided in the review which ends up becoming ambiguous, i.e., only the negative aspects such as high price or preference in another color are indicated but a positive rating is given, because the customer is nevertheless satisfied with the other aspects of the product. But from the information provided by the text in the review the model has no way of knowing that the customer is overall satisfied and so classifies it as negative.

As expected, the best model was the SVM because of its generalization ability in high dimensional feature spaces, as is the case of text classification.

As we can conclude by the metrics, the main difficulties of our model are to distinguish between poor and unsatisfactory and between good, very good and excellent. In other words, the model can identify if the review is positive or negative, but it fails to label correctly inside those categories. For example, the model will classify “Not a fan” as unsatisfactory instead of poor and “Works shockingly well” as Very good instead of excellent.

Another “problem” is when it must classify reviews that are positive but have negative words on it. For example, “what!? no brush!?” is classified as unsatisfactory but should be very good and “Difficulty staying in” as poor while it should be also very good.

To finish this discussion topic, I believe that one of the biggest barriers of this model is that it removes every punctuation mark, and as a result, it doesn’t consider that aspect when classifying each review.

Future Work

If we had more time, the main thing that we would like to implement was the punctuation aspect, especially de question mark and exclamation mark recognition, so the model could easily recognize the context of each review.

Beyond that, we also would like to create classes of words to each review category in order to solve the problem with mixing all the positive reviews and all the negative reviews and by doing this we would increase the accuracy of each classification. We can also add the named entity recognition (NER) to our preprocessing task.