Ardigò Guglielmo, Vitali Giorgio, Apavaloaei Eduard

Assignment - Web Scraping and Data Analysis

The main goal of our project is to employ web scraping and sentiment analysis techniques in order to extract, process and analyse data.

In the first phase we have implemented a function in order to scrape the most important information about the analysed product and to organize them into an understandable table. For the identification and the extraction phase, we have chosen CSS identifiers for the review extraction part and Xpaths for the others. In the second phase, we have cleaned and described the data. To graphically represent the data, we have used the *ggplot2* package for the bar plots and the histograms and the *wordcloud* package for word clouds visualizations. Then, we have implemented a dictionary-based analysis, choosing the *Bing* lexicon for the tidy approach. Later, we also used the udpipe one. Finally, for topic modelling we employed the LDA (Latent Dirichlet Allocation) text mining technique.

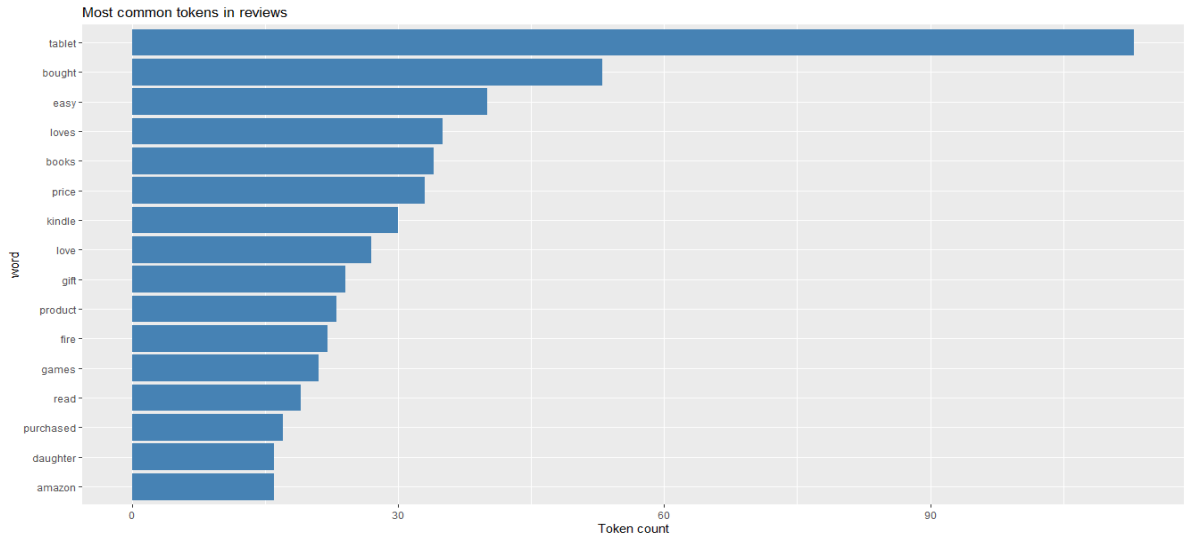
For this purpose, we have chosen a product from Amazon UK. It is a fire 7 tablet, 7'' display and 16GB storage, released in 2022. Basically, it allows the user to read, browse the web, watch movies and listen to music, among other things. We focused our attention on product information (product details, number of costumers rating, fastest delivery date) and reviews.

We have run into a problem with scraping data directly from the website[1], so we decided to use a related dataset that it was provided to us. This dataset is composed by two-hundred and one observation and twenty-four variables. However, for our analysis we have used only four of them: title, text, username, and rating(star_score). To these, we have added a fifth variable, doc_id, which makes it possible to uniquely identify reviews.

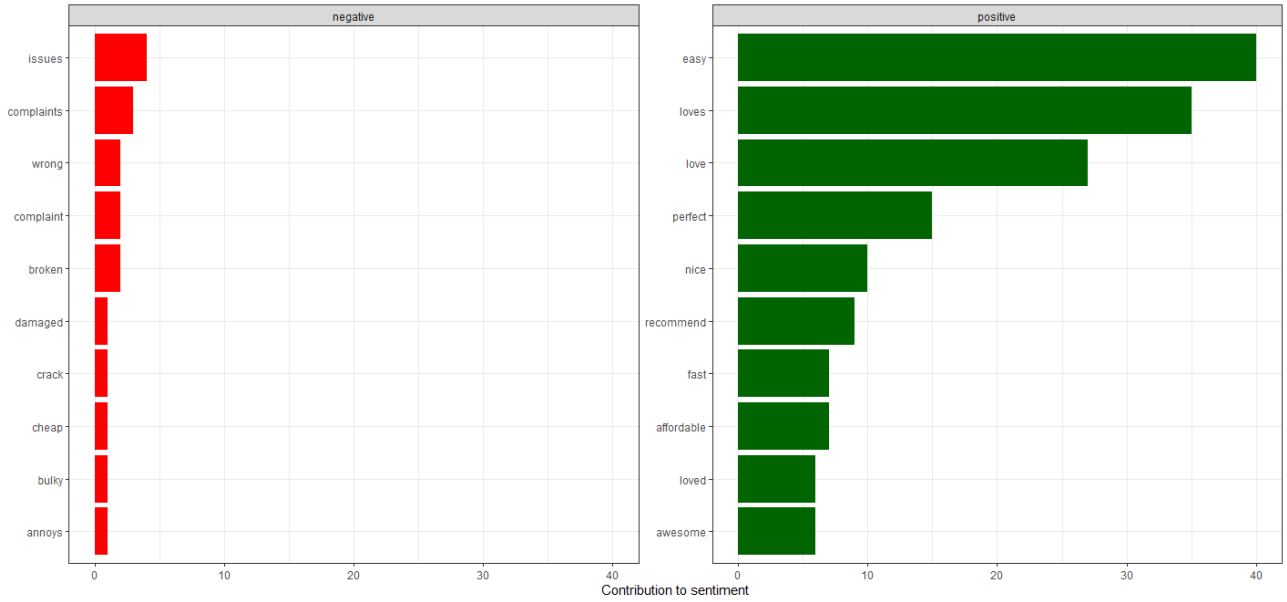
We proceeded by tokenizing the reviews using the *unnest_token* function and removing the stop words with the *anti_join* function. During the process of eliminating stop words, we have observed that 3281 words of the reviews were deleted. These are words that often do not carry much meaning or add significant insights to the analysis, on the contrary, they increment the bias or the noise introduced by common words, preventing us from focusing on the more meaningful and informative words in the dataset.

[1] In the specifics, when we try to extract the html page from page 2 of the review section onwards, the html provided by Amazon.Uk corresponds to a captcha code in html form, and not to the real web page displayed. This made it impossible to identify the html code segments, which corresponds to the parts containing the reviews text and stars rating.
The source of the stack overflow page referring to the scraping problem for python application, but also applicable for R:
https://stackoverflow.com/questions/75993627/web-scraping-reviews-from-amazon-only-returns-data-for-the-first-page

We have represented graphically the most frequently used tokens:

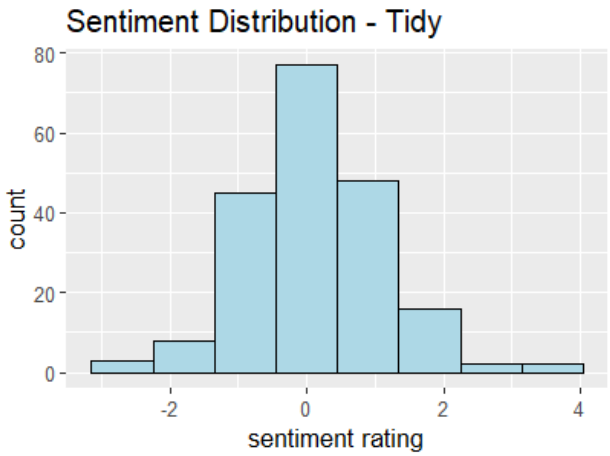


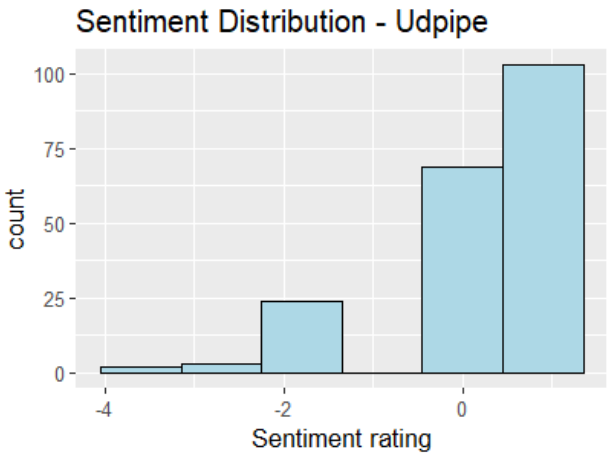
Next, we have performed a dictionary-based sentiment analysis on the dataset by calculating the sentiment score using the "Bing" lexicon. Then, we have graphically represented the results in a bar chart that contains the top ten contributing words for each sentiment category.



Considering the number of words in each category, we can precisely conclude that, of the 200 reviews analysed, the vast majority have a positive sentiment, while the ones with negative sentiment are focused on the problems that affect the integrity of the product.

We have run the same sentiment analysis using udpipe. The resulting scores are stored into two new columns that categorize the sentiment scores from the tidy and udpipe approaches. Using the scores from these new columns, we have created two histograms that visualize the distribution of sentiment scores.





After that, we created other two columns of categorical variables that divide the results into three categories: "positive", "neutral", and "negative". The *"tidy"* column is assigned the value "positive" if the "*tidy*" value is greater than 0, "neutral" if the value is equal to 0, and "negative" otherwise. Similarly, the "*udpipe* " column is assigned to "positive" if the value is greater than 0, "neutral" if equal to 0, and "negative" otherwise. The results are summarized in this table:

	UDPIPE		
TIDY	negative	neutral	positive
negative	4	3	4
neutral	0	16	29
positive	1	5	139

As we can observe, for most of the reviews (159 out of 200), the analysis with the tidy and the udpipe approach led to the same results. It is also significative the fact that, according to the udpipe analysis, 29 reviews have positive sentiment, while, with the tidy approach, the same reviews turn out to have neutral sentiment.

The "*udpipe*" and "*tidy*" approaches are two different methods used in natural language processing (NLP). The key differences between them are the following:

- With the tidy approach, tokenisation is performed using some auxiliary functions from the tidytext package. Udpipe, instead, performs tokenization as part of its pipeline, using built-in models to tokenize text.
- Regarding sentiment analysis, with tidytext the analysis can be performed using lexicons, by applying sentiment score to individual words, without considering the word order or the grammatical word type. Udpipe, on the other hand, provides additional functionalities by which is possible to define some polarity negators, amplifiers and deamplifiers to take into account in the computation of the polarity score, as well as the possibility to indicate the number of words before and after the analysed token that has to be considered into the analysis of each single token.
- Tidy approach can be applied to any language for which appropriate tokenization and sentiment analysis resources are available in R. Udpipe supports multiple languages and provides pre-trained models for various languages.

In summary, the main difference between the two approaches lies in the integration of tokenization, languages supported, sentiment analysis and built-in functionalities.

In our analysis, the udpipe approach was found to be more accurate (as might be expected), having a sentiment polarity distribution more in line with the analysis described above, according to which words with positive sentiment were far greater than words with negative sentiment.

Regarding the impact that the choice of the lexicon has on the result of the tidy dictionary-based sentiment analysis, we have run three separate analyses using three different sentiment lexicons: "bing", "afinn" and "nrc". We have computed the sentiment scores for each review in the dataset, from each of the lexicons, and inserted them in a wide format table that includes all the results. Finally, we have run the correlation matrix between the sentiment scores, as shown in the following table:

	afinn	bing	nrc
afinn	1.0000000	0.6551493	0.5381850
bing	0.6551493	1.0000000	0.4657534
nrc	0.5381850	0.4657534	1.0000000

After that, we have merged the tibble containing the sentiment score computed with all the three lexicons with the initial database, through an inner_join. As shown in the following table, we can see how afinn is the lexicon that, generally, defines a higher sentiment score. Bing and nrc, however, return more similar values, except in some isolated cases.

	doc_id	text	afinn	bing	nrc
1	27	Purchased this for my fiance's youngest son for Christmas a...	12	5	3
2	181	We have always been Nexus Tablet desciples, and in fact hav...	7	5	3
3	30	The images are crisp and text very easy to read. Apps are go...	5	4	0
4	164	This tablet is perfect for the kids. Easy to use, dependable, a...	4	4	1
5	29	This tablet was a gift for my father-in-law and is his first tabl...	9	3	6
6	59	My son has the Hd fire used for last 3 years, I bought this o...	2	3	3
7	70	Tablet is great for kids and adult use. I purchased the tablet ...	2	3	2
8	85	Much lighter than my old kindle. Love it, love the fact that it...	6	3	2
9	95	Upgraded my kindle, easy and fast transfer from old to new	1	3	0
10	107	Decent camera, clarity on the screen, fast downloads.	2	3	1
11	114	Gave this product to my nephews and they love it. Easy to c...	5	3	1
12	117	Got two for my kids for Christmas and they're awesome! An...	10	3	1

Once we finished the dictionary-based analysis, we have proceeded making a per-topic-per-word analysis using the LDA machine learning algorithm. This algorithm allows us to identify the main topics of the analyzed text, in particular, we have chosen to set $k = 2$, so the number of topics searched by the algorithm was 2. To see the probability of each word being generated from each topic, we plotted the results of the implemented LDA method in a bar plot containing a graph for each topic, with the top 15 most probable words per topic (not considering the word "tablet", by far the most common, but unrepresentative word):

From this analysis, we cannot distinctly identify the two arguments, because the words are practically the same in both arguments, even if they are in a different order. We also tried to increase the number of topics, to see if the situation would improve, but, as we might expect, it got worse, creating more topics with the same words. Based on this data, we can conclude that the LDA algorithm is not suitable for current analysis, and this may be due to the way that reviews are written. As many of them, in fact, contain the same words, and considering that the LDA model analyze only the single words and not the context in which they are, it is not able to find arguments.

A further demonstration of the above problems can be obtained by analyzing the topic proportions. As shown in the following table, the calculated gamma value represents the percentage of words for each document that are generated by each topic.

	document	topic	gamma
1	1	1	0.5002150
2	2	1	0.4929359
3	3	1	0.4747359
4	4	1	0.4935086
5	5	1	0.5113562
6	6	1	0.5036335
7	7	1	0.5033199

As can be clearly observed, the two topics always represent about 50% of the words in each document. This shows that the two topics are not clearly distinguishable, and are not able to represent two different aspects of the reality.

In conclusion, we can say that the made analysis shows that the product reviews of the *fire 7 tablet* are mostly positive, and highlight a general satisfaction in relation to the purchase of the product. Some of the most frequently used words in reviews, in fact, are "easy", "love", "perfect", "recommend" and "affordable", and all demonstrate a positive feeling from customers. On the other hand, the search for different topics within the reviews has not been successful, because of the problems of the model in relation to the type of data used.