M2 - STAKEHOLDER

Malika Kuhlman Hansen
Michael Dybdahl

Aalborg Universitet
September 2019

1. Definition of a problem statement and a short outline of the implementation

We found a dataset containing reviews from Amazon on some of their electronic devices like the Kindle, Fire TV Stick etc.

1.1. Problem statement

How are the words in the text-reviews and based on a tidy format of the reviews, can we then compute any topics based on this? Can we compute a model that predict if a review is rated as a good review, based or linked to the NLP results?

2. Description of data acquisition

The data was collected from https://data.world/.
The dataset contains 24 variables and 5.000 observations (reviews).

3. Data preparation

First, we **cleaned** the data for variables that we were not going to use and removed reviews with NA values.
After this we recoded some of the variable's names and the format of "data" and "rating".

4. Natural Language Processing (NLP)

4.1. NLP – preparation (Tidy NLP)

First step we did a tokenization by removing stop words, meaningless words, non-alphanumeric characters etc. This give us a tidy format containing words and their counts from the reviews' text.

In this section we also looked at the topwords from the reviews, where words as "tablet", "love", "kindle" (amazon tablet), was in the top of frequent words.

4.2. Simple vectorization (Tf-idf)

In this section we added the tf, term of frequency, idf, inverse document frequency and tf-idf, term frequency–inverse document frequency to the tidy data.

Here we could see that "tablet" was the words with lowest tf-idf values, which makes sence, because this word was the most frequent and therefor also the most common word in the reviews.

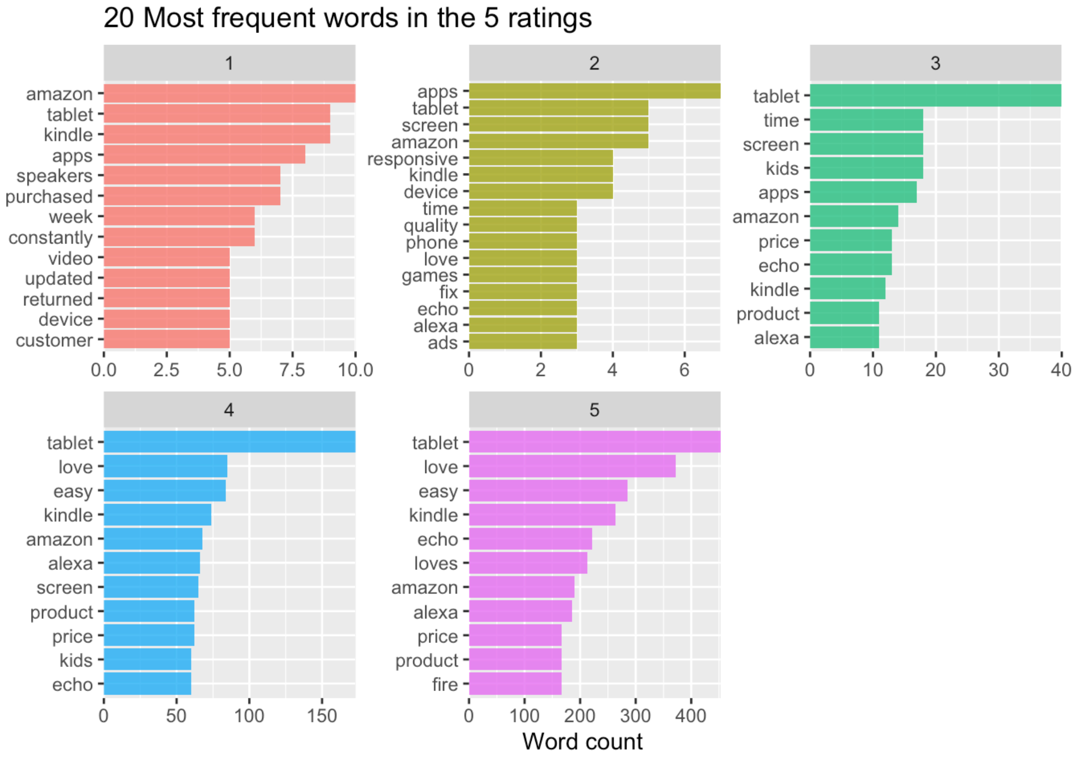
4.3. Topic modelling (LSA)

Here we performed an LSA, which is stable when attempting to do dimensionality reduction as preprocessing for supervised ML workflows, or for visualization, but did not give us any human interpretable topics

Using "Umap", we plotted the LSA findings, which gave us 9 clusters.

4.4. EDA / simple frequency-based analysis

We also did some explanatory data analysis, where we looked at the 20 most frequent words in the five ratings (rating 1-5).



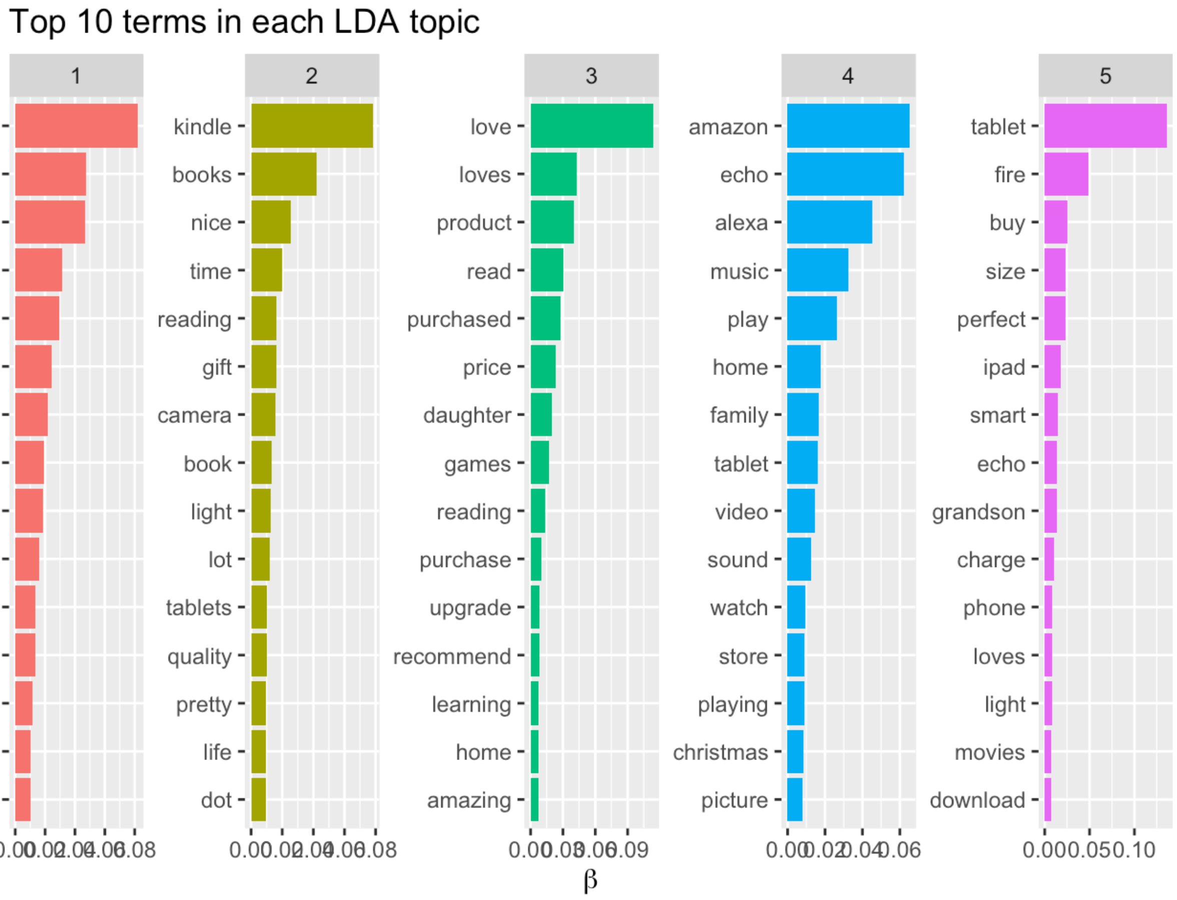
Figur 4.1: 20 Most frequent words in the 5 ratings

We see that:

- ***Rating 1:*** *contains words like constantly, returned, purchased which could have a negative meaning and then there are a lot of words about the electronics.*
- ***Rating 2:*** *contains words like responsive, time, fix which also could have a negative meaning, and here we also see words describing the electronics*
- ***Rating 3:*** *contains words like the two other, but here the amount each words are used are almost the same for them all, except tablet. Therefore this looks loke the ones that don*t take this review seriously.*
- ***Rating 4:*** *containd words like love, easy, price and product this would usually have positive meaning in a text together with the describtion of the electronic.*
- ***Rating 5:*** *contain words like love(s), easy, price, wich are way more positive words than the first two ratings. We also see that the amount each words are used are way higher here.*

4.5. Topic modelling / Clustering (LDA)

Using LDA, we computed five topics based on a document-term matrix (dtm) and we created plots for the five topics containing the 10 most frequent words.



Figur 4.3: Top 10 terms in each LDA topic

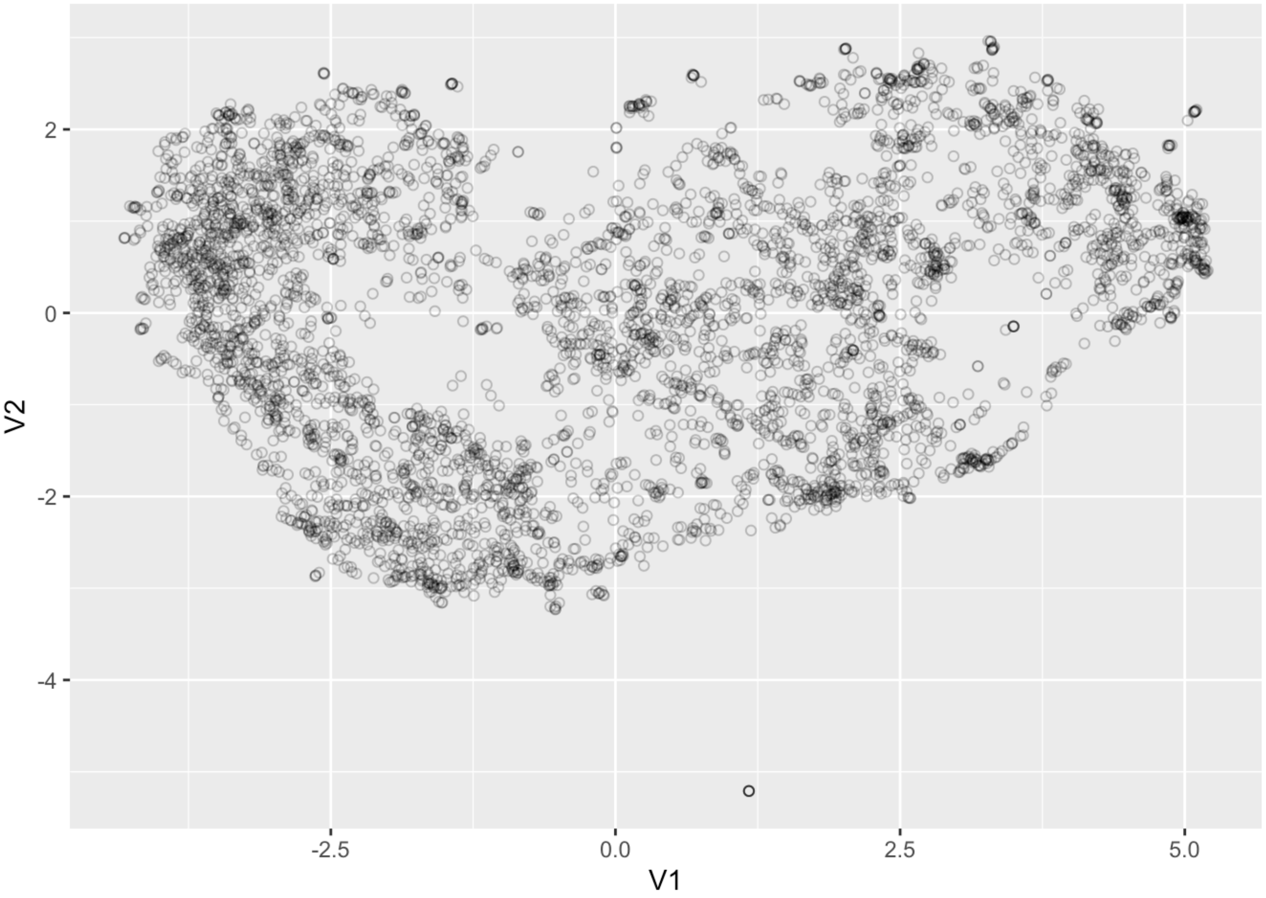
- **Topic 1:** We see that topic one is about quality for HIFI stuff.
- **Topic 2:** We see that topic two is mainly about reading. The word Kindle are top 1, but words like books, read(ing), light, are also in here.
- **Topic 3:** We see that topic three is re mess of different types of words.
- **Topic 4:** We see that topic four is about HIFI, since words like echo, alexa and music are highly rated here.
- **Topic 5:** We see that topic five is mainly about tablets, since the words like tablet, fire and ipad are in this topic. together with some other tablet related words.

Overall, we can say that the topics are mainly split into categories and not rating like we did earlier

4.6. Embedding-model based vectorization (GloVe)

Here we computed a GloVe Embedding-model.

Using the GloVe model, we looked at the clostest words to “kindle” and “tablet”. As seen earlier the data contains mostly of high rated reviews. Therefore, it makes sense that the words like tablet and kindle follows by positive words like great, love and good.



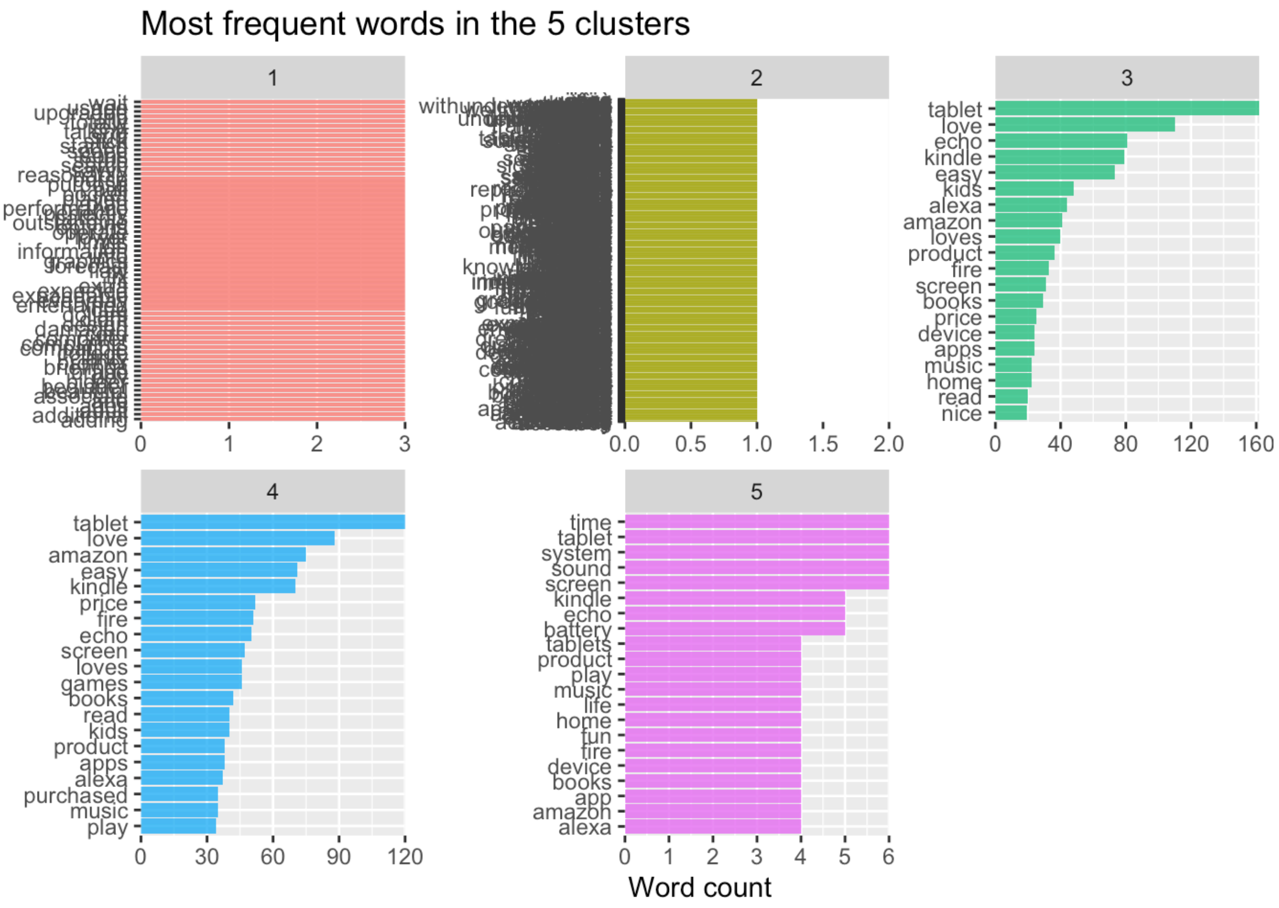
Figur 4.4: GloVe model by Umap.

Looking at above plot, then it does not look like the data have any clusters right now. If we look closely, we could argue about some few clusters at the plot, which might be tablet and/or kindle stuff.

5. Supervised / Unsupervised ML

We did kmeans at the tf_idf values, and computed 5 clusters since we had 5 different ratings in the data.

5.1. Unsupervised ML



Figur 5.1: Most frequent words in the 5 clusters

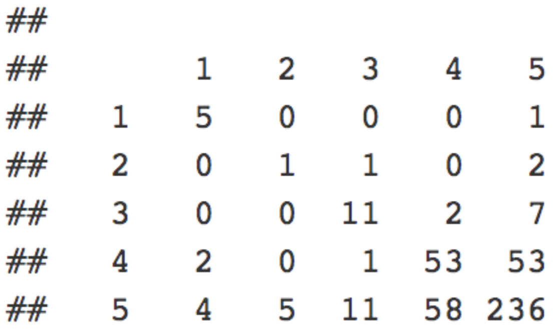
Unfortunately the plots are messy. since some of the clusters (cluster 1 and 2) contain words represented the same amount of times.

The plots contain almost the same words, it is hard to find differences in these three plots. So apparently clustering based on the tf-idf values does not makes sense.

5.2. Supervised ML – Method 1

We wanted to classify the five ratings based on a tidy format, classifying all five ratings at once, using a logistic model, using this approach: https://juliasilge.com/blog/tidy-text-classification/

This gave us following confusionmatrix:



```
##
##          1    2    3    4    5
##    1     5    0    0    0    1
##    2     0    1    1    0    2
##    3     0    0   11    2    7
##    4     2    0    1   53   53
##    5     4    5   11   58  236
```

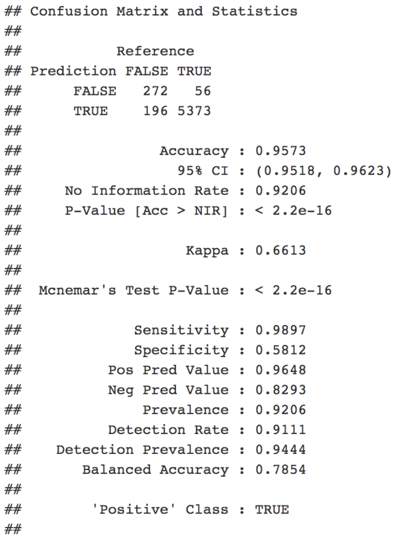
Figur 5.2: Confusionmatrix for method 1.

We see that the model is not that good, since it only predicts ~50% correct.

5.3. Supervised ML – Method 2

We also used our approach from M1, to classify whether a reviews was rated good (4 or 5) or not (1-3).

Here we tested this on three models: Logistic, random forest and decision tree. The random forest gave us the best model with a Specificity at 58,12%.



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE   272   56
##      TRUE    196 5373
##
##                Accuracy : 0.9573
##                  95% CI : (0.9518, 0.9623)
##     No Information Rate : 0.9206
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6613
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9897
##             Specificity : 0.5812
##          Pos Pred Value : 0.9648
##          Neg Pred Value : 0.8293
##              Prevalence : 0.9206
##          Detection Rate : 0.9111
##    Detection Prevalence : 0.9444
##       Balanced Accuracy : 0.7854
##
##        'Positive' Class : TRUE
##
```

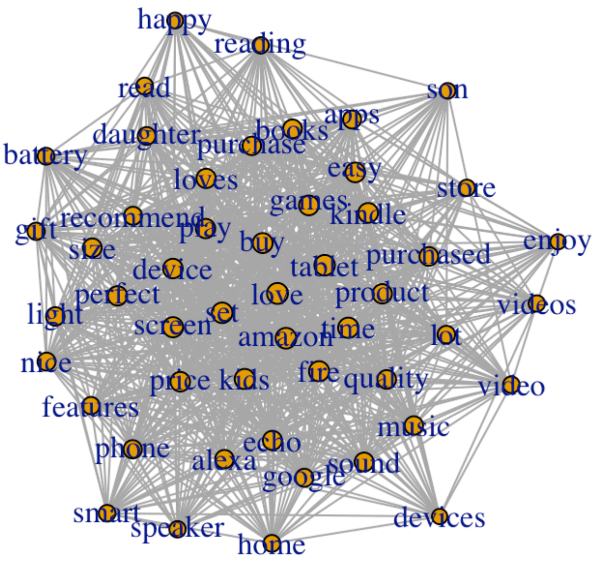
Figur 2: Confusion matrix of RF-model - method 2.

6. Network analysis

We also did some networking based on the reviews data, although not with the best results.

We did several constructions and tried interpret different attributes to the network, but for some reason we could not include any attributes. For example we wanted to try include rating and primaryCategories and use them for coloring or shaping the nodes

This gave us a network containing the top 50 words, where the edges told if a word was connected (used) in the same review.



Figur 6.1: Network of words

The nodes are sized by the degree, which can be difficult to see, because most of the words have around the same degree, although we could see that “tap” (degree = 25) was smaller than fx “love” (degree = 49).

This makes sense, that not that many reviews have tap in the text, but rather more has love inside (probably because the many high rated reviews).