**Resolution**

**Dataset**

I started with understanding the data I had - first of all, this dataset is small. After the Language filtering, I found that ~40% of the data was not in English. So, I dropped it... It left me with really small number of apps to train any classifier on.

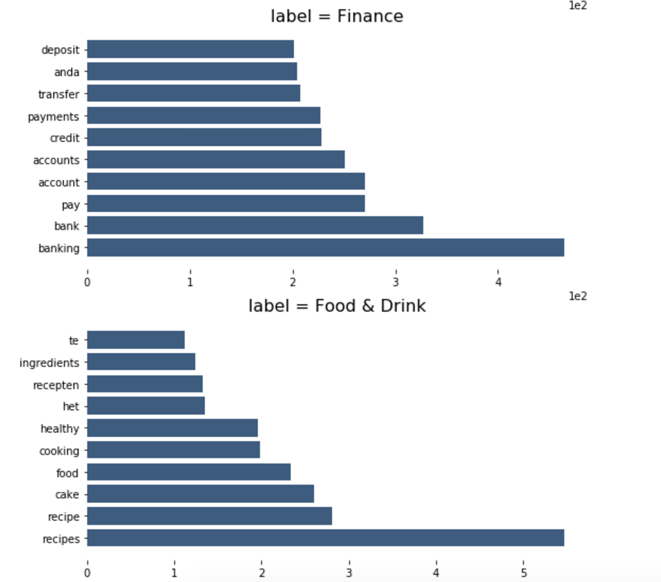


**Strategy**

I saw that there are segments with very smalls amounts of examples - for example: Adult content, Home & Garden etc... I chose to *discard those segments* due to bias it creates.



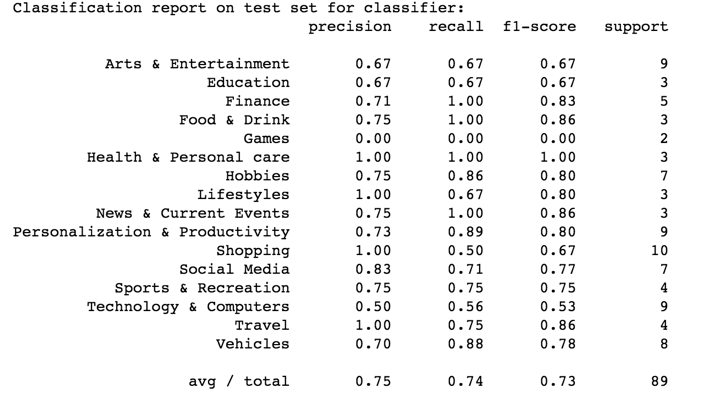
I chose a *bag-of-words approach* with CountVectorizer but than I moved to *TfidfVectorizer* to enhace results. I started using the *stop-word* in the vectorizer when I realized that very common words are used in the classifier (words like 'of', 'a', 'the' etc). So I added the stop-words and the idf threshold to filter very common words.



example of top words to identify two segments

**Performance**

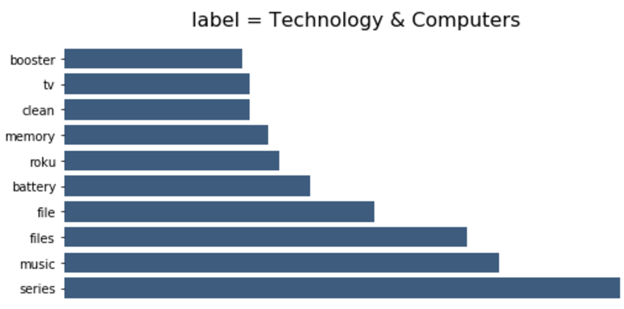
The accuracy is about 74%.



confusion matrix and classifier performance for each segment

Average precision is about 75% - The best precision segments are *Health, Lifestyle, Shopping and Travel* - those segments have distinct words that identify them (unlike *Tech* - that have a high overlap ratios). This is why we see high precision rates with distinct segments like Health and Travel and not in Tech for instance.

Average recall is about 74% - In my opinion, the main cause is the *overlapping* of the labeled data. Because lots of Apps can relate to several segments we see this degraded recall percentage; segments like Tech and Entertainment can related to many other apps and vice versa. Another consideration is that each app was labeled by a different creator, which means that his opinion may be different on how he/she "understands" a segment.



in the “Tech” segment the words will overlap with other segments

**How to make results better**

* Data: Need more quality data! 40% of the dataset was not English and there were segments with really small amount of examples.
* Classifier: Enhance the results with K-Fold approach.
* Different Classifier: I would train a LSTM deep network classifier to include the sequence of words instead of looking at in as a bag-of-words