Capstone Project 2 – Report

Hearing the customer's voice – sentiment classification and summarization of Amazon reviews

- ## Business Problem

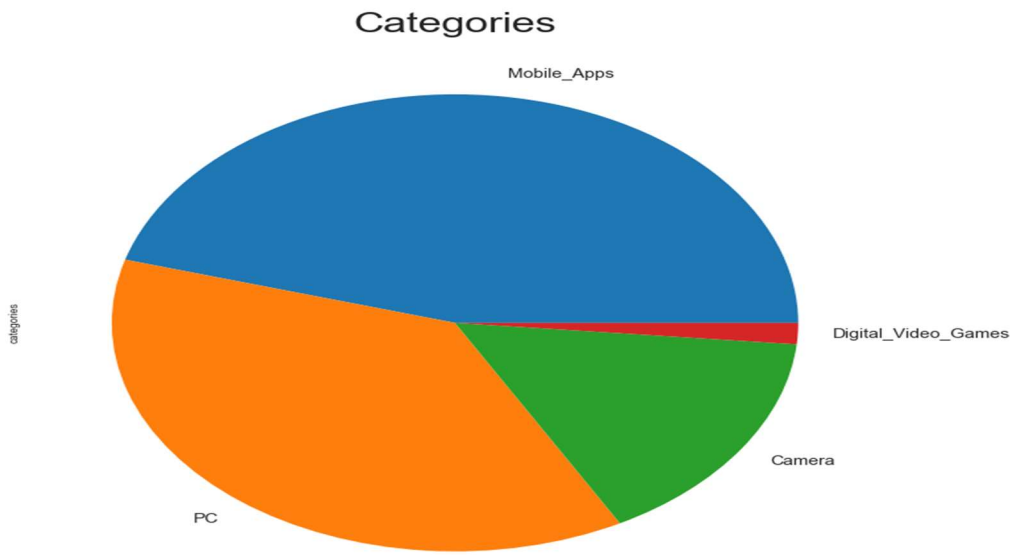
Many enterprises which are challenged to improve their products and services, either if they are sold online or in a more traditional way, need to rely on interaction with and feedback from their customers. Hence, the customer's "voice" about the service or product might be of extensive value in order to re-design the way a company is marketing and servicing its products or it might provide important information for the development of new product series. Therefore, extracting and summarizing information of product reviews is key for many companies to remain competitive. However, many NLP applications focus on classifying reviews to be either positive or negative or do provide another kind of sentiment quantification. In turn, these measures are important to summarize the impression and feelings of customers over a wide range of products. Nevertheless, classifying reviews with regard to sentiment patterns might not provide the sufficient information for further improvements of products, because useful information about details or key problems customers are facing is getting lost. Thus, besides classifying reviews, this project aims to develop a method to summarize product reviews. For instance, it would be quite helpful for a product or marketing manager to get a summary of the 5,000 worst feedbacks of a product. Moreover, from another angle, it could be also quite useful to get a summary of the 5,000 best feedbacks in order to figure out "unique selling propositions (USP)".

- ## Approach

The approach for solving above targets is twofold. First, Deep Learning is applied for the classification task, i.e. to predict the ratings a reviewer gives a product. In a second step, extractive summarization techniques get developed leveraging the encodings provided by the network. Thus, the model of this project should be able to 1) classify a review and 2) to provide a framework to generate summaries of certain reviews which could get selected by a user. Basically, step one is necessary because many companies might not have a clear functionality for customers to rate products on a scale, moreover, in other applications – where the company relies, for example, on datasets like emails or other messages form clients, there are not any possibilities to give categorical ratings without the need of a human. Thus, step one should provide an important output which then helps to classify reviews without the need of labelling the data by humans in a company. Step two is basically leveraging the embeddings of Deep Learning model to generate summaries by means of most representative sentences or reviews.

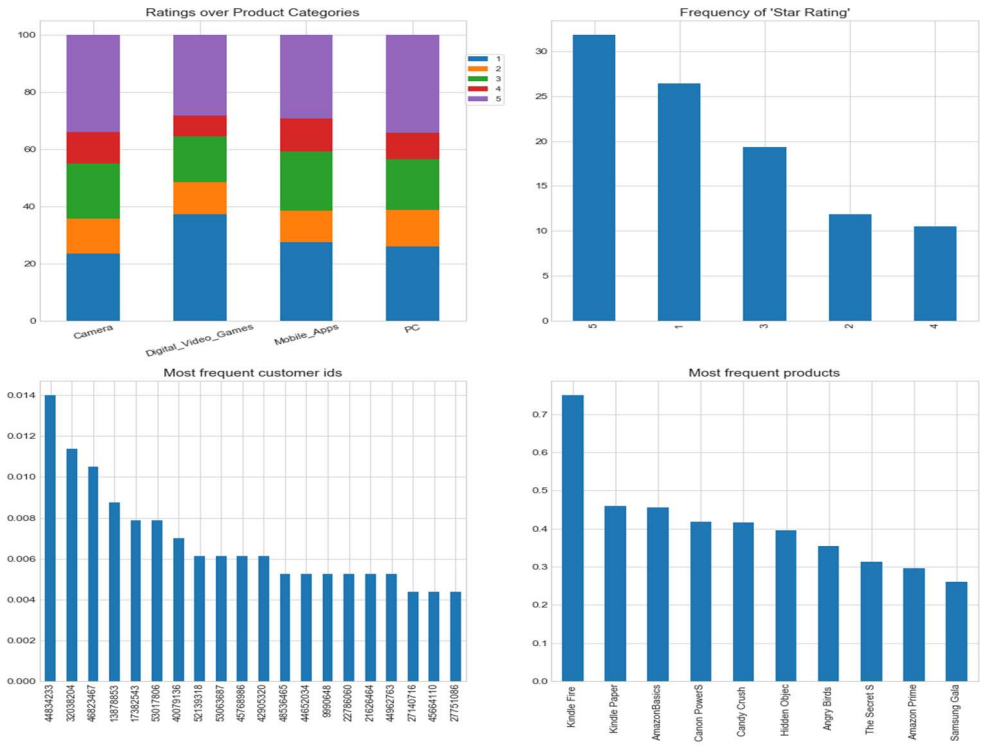
- **Data Query**

The dataset has been created by the author in order to generate a representative subset of Amazon reviews. Following the page https://s3.amazonaws.com/amazon-reviews-pds/tsv/index.txt which provides links to reviews of diverse categories hosted by Amazon, reviews about "Mobile Apps", "PC", "Camera" and "Digital Video Games have been downloaded. These categories could be subsumed to an overall product topic like "Digital Entertainment and Electronic Products". Basically, all four category files had different sizes, ranging form one million to 200,000. To generate a smaller, but still representative dataset, reviews are chosen randomly by a size of five percent. Within these datasets some reviews are duplicated which is a result of Amazon' product classification. After dropping duplicates and a small number of three "NA" values the created dataset contains 114,192 reviews with the proportions shown in the below chart.



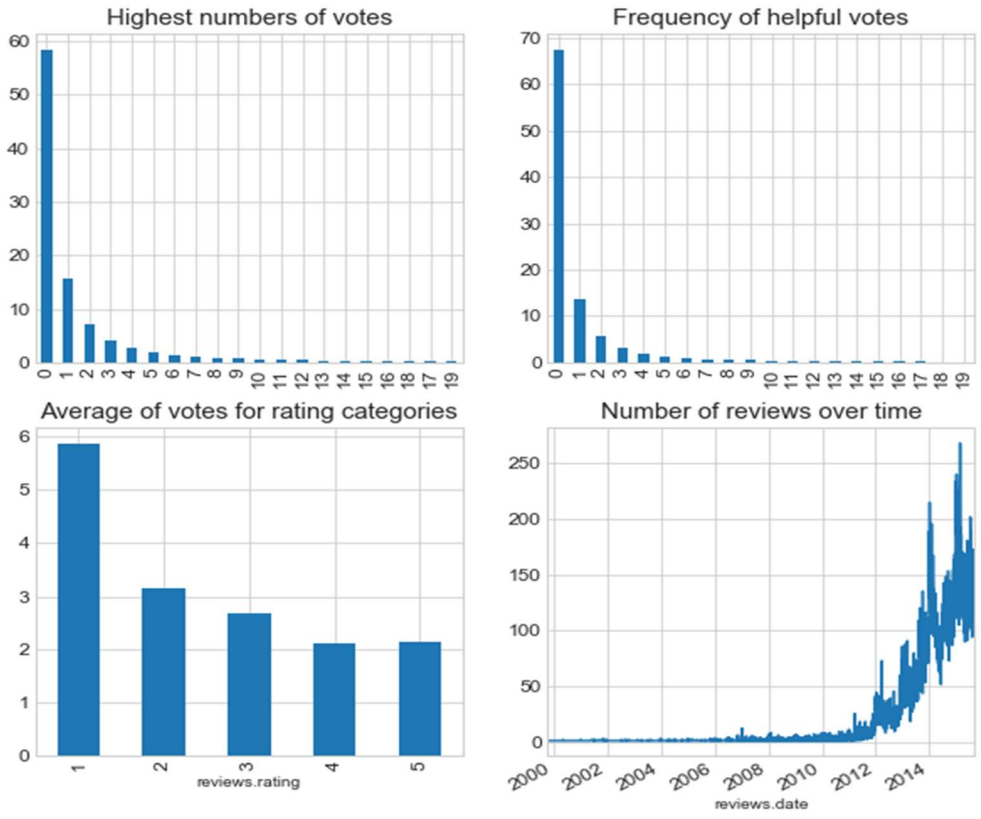
Overall, Amazon provides meta-data to its reviews as well, thus besides "Star rating", review text and title, there is also information about the marketplace, the customers, the name of the product, whether the review is a verified purchase or not, how often the review is considered as helpful, its date and votes of other customers about the review. Thus, the following section aims to describe these features.

- **Data Exploration**



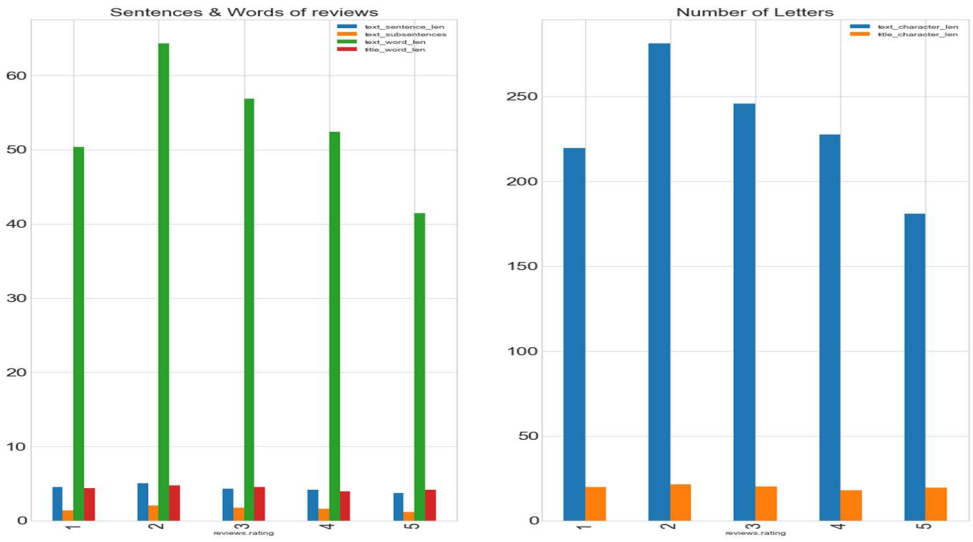
With regard to the distribution of “Star Rating”, there is a considerable bias towards higher five-star ratings which is shown on the top right chart. Moreover, a bias towards polarity is obvious by simply cumulating the lower and upper tails of rating frequencies: five and four stars make up more than 40 percent of all reviews and the lower end – two- and one-star ratings - sum up to approximately 40 percent as well. Instead, the middle – three-star ratings – shows a comparably smaller frequency.

This could be the result of diverse behavioural biases on part of people who are writing reviews. For instance, there might be a rather strong motivation to post reviews if people are rather very satisfied or disappointed. Considering the percentage proportions of “ratings” over the diverse product categories, reviewers have been quite content with “Cameras” while on the other side “Digital Video Games” show the least frequent five and four star ratings, instead, negative reviews (one- and two star) are comparably high as it gets depicted by the top left chart. Breaking reviews down to products, “Kindle Fire” has been most often reviewed, followed by “Kindle Paper” and “Amazon Basics” as it is shown by the lower right plot. However, there is only a slight concentration of customers writing reviews. Taking into account the top left plot there is one customer which has given reviews quite often. Though, breaking the number down, this customer has written approximately 1,300 reviews which seems to be rather a high number. This could be a possible outlier, though, it might be the case that with this specific customer’s id more people are acting, i.e. in a school teachers are using the same account.



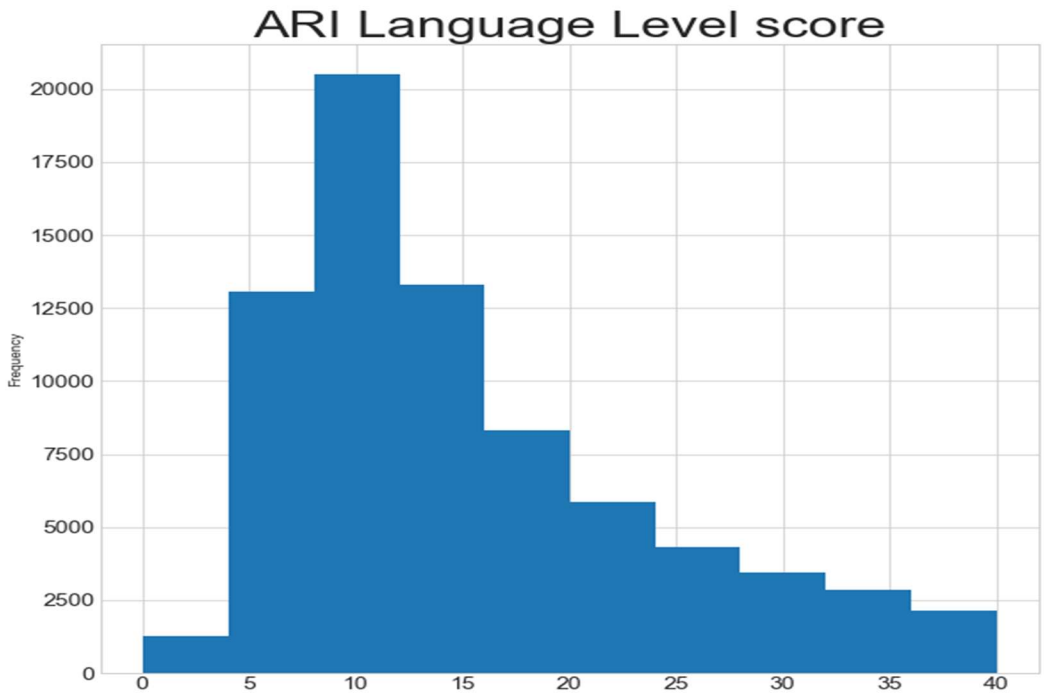
Taking a look at the other features provided by Amazon, most reviews do not get any appreciation by other customers or Amazon users. As it can be seen by the two plots in the upper panel of the figure above. Most reviews do not get any vote at all, while only more than ten percent of the reviews got one vote. Nevertheless, looking at the average number of votes with respect to the ratings it is remarkable that the worst rated reviews have been voted quite often. Possibly, this reflects situations when other customers could have been grateful for being warned about purchasing a bad product. A larger part of reviews has been written between 2011 and 2015. The fact that till 2011 there haven not been many reviews might be due to the fact how online retail business has developed; basically, before 2010, people have not used the internet for shopping that often than nowadays.

In order to get an impression of the linguistic properties the reviews' texts have been processed to extract further features. So, the number of words, characters, sentences and sub-sentences are assessed by very basic tokenization. Thus, the charts below show the average number of words, sentences and characters for each of the categories.



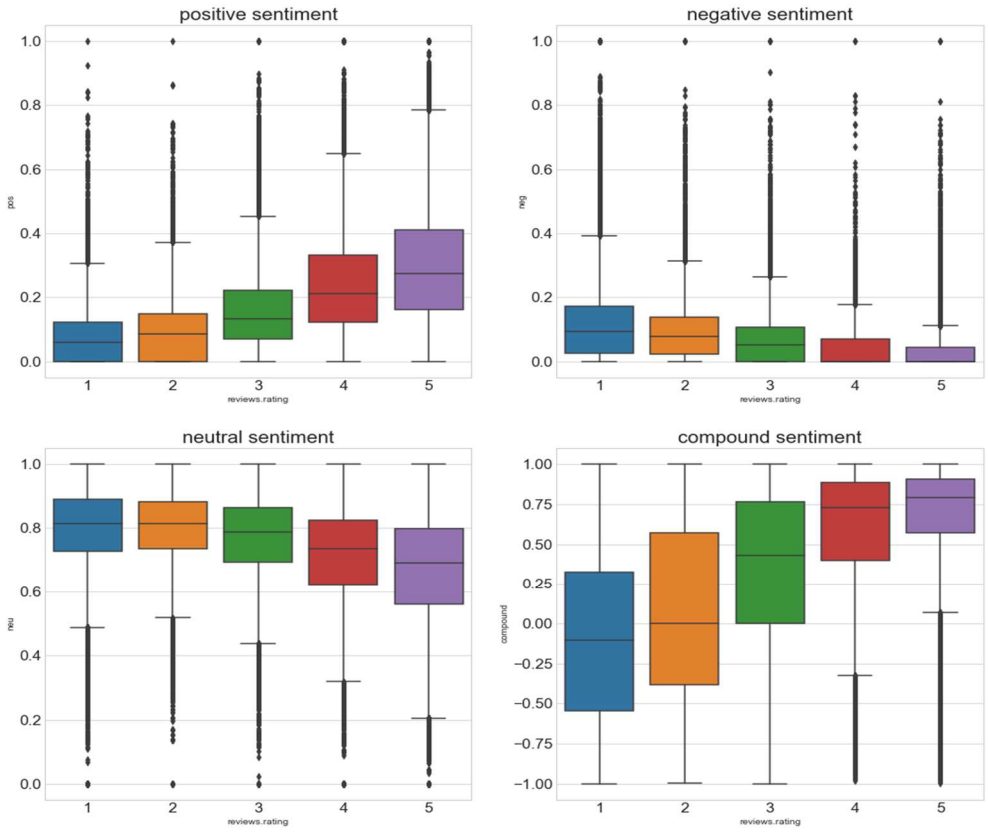
Overall, medium to slightly badly rated reviews (one- and two-star rating on the first left chart) are on average longer than those rated on the more positive end. Though, there are not any indications for patterns with respect to different rating categories.

In order to assess the level of the understandability of reviews the Automated Readability Index (ARI) has been calculated which basically forms a ratio between word and sentence difficulty. ARI levels of 17.00 correspond to text quality written by pre-college students. The histogram below shows these levels for aggregated from all reviews with an average of 15.00 and a lower quartile at 8.8. Considering that reviews are not bounded to any formal requirements and that writers are rather likely to express themselves in an ordinary language, this ARI statistic reflects an overall sufficient level of understandability of the dataset's reviews.



The topic of language modelling and processing is often confronted by behavioural biases the way people tend to express their opinions and sentiment. Hence, to get an impression about the divergence between assigned rating and actual sentiment patterns within the reviews a lexical approach is leveraged. Accordingly, VADER's sentiment lexicon provides a scoring scheme for texts, especially for those in social media. This rule-based approach is able to measure prevailing sentiment patterns on the dimensions of "positive", "negative"," neutral" and "compound". While the former dimensions are quite clear to understand, the compound score resembles an aggregate of all rating scores that appear within a text. This score takes values between minus and plus one, hence, it could be considered as an overall polarity score.

The charts below show boxplots of VADER's sentiment score with respect to the rating of the reviews. Actually, ignoring outliers the scores' medians seem to be well aligned with star ratings. However, neutral scores are comparably higher for lower rated reviews which might hint at some kind of behavioural bias, i.e. people could be quite dissatisfied with a product but might express their view with more neutral wordings. Moreover, for prediction algorithms that aim to classify sentiment or star ratings this ambiguity of writings might be the reason for biases and remarkable performance problems.



- **Cluster Analysis and Topic extraction**

Besides lexical approaches to gain some insight about the reviews' texts, methods of topic extraction might provide further insights. Thus, the texts have been processed by removing stop words, i.e. words which do not add exploratory value, by lowercasing and by adjusting for n-grams (one-gram). In addition, only words are considered which appear more than three times in all reviews. As a result, features get generated for the reviews by counting the occurrences of approximately 20,000 tokens. The word cloud below shows the most frequent words of the reviews.

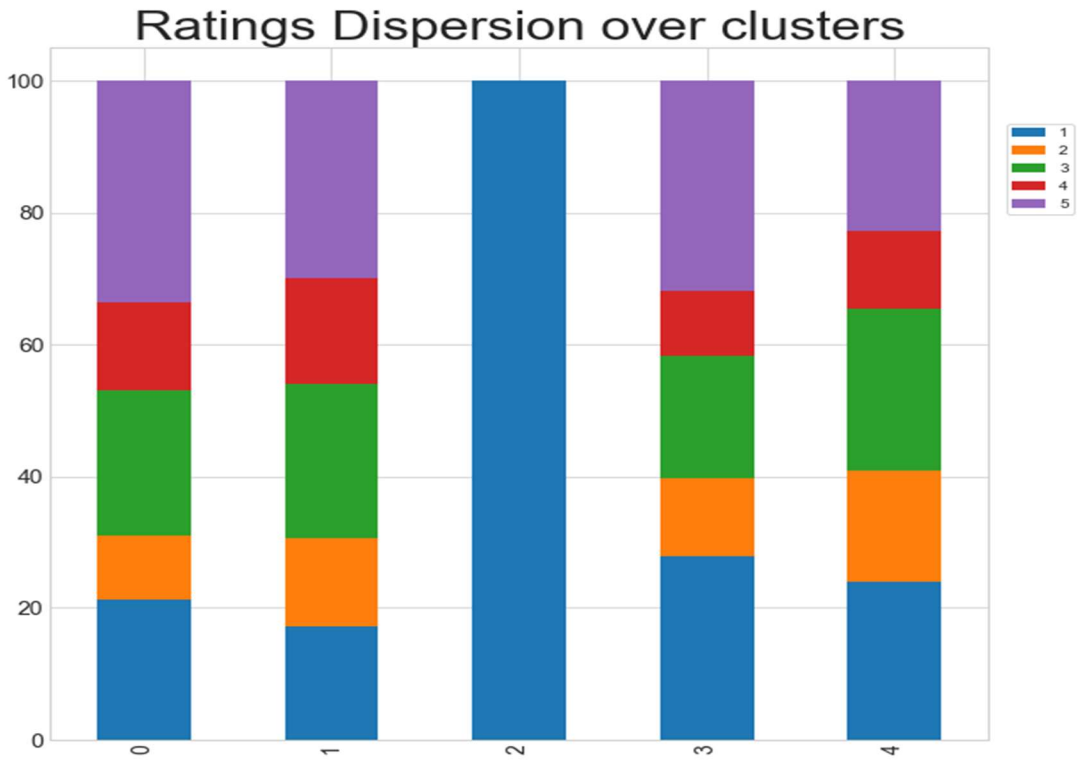
Extending this approach of the document term frequency matrix, dimensionality reduction techniques are applied to extract topics and to perform cluster analysis. Thus, the term count matrix is being decomposed by Non-Negative Matrix Factorization (NMF) which represents a suitable technique for sparse matrices. In this regard, text topics are extracted by means of high vector loadings of the resulting feature matrix. So, the text below shows the 30 most relevant words/one-grams of the Amazon reviews.

"br router review cons device mode pros ve update issues netgear usb 10 windows using need issue software screen button want support ll video settings power bit high look live camera pictures video good battery quality cameras use canon great photos card zoom flash picture shots time sony digital batteries mode shoot light small took bought easy taking features better game play fun time games good playing love great free played money level graphics levels really buy kindle enjoy don way make stars recommend awesome coins people best new download use work time product good drive battery great computer laptop did mouse usb works using does problem device new bought router used buy screen power amazon price windows card support case ipad keyboard cover tablet screen fit kindle great use does nice little good stand protection fits hold cases love plastic doesn keys air leather mini key using looks feel 34 amp keyboard tablet just amazon ssid key new did screen time mouse button device way access called problem t1 support people double ve computer customer said kindle purchase say lens canon focus lenses use great nikon good zoom quality light macro image kit sharp update 7d mode using ring better hood photos ef low shots 50mm sigma ve field app kindle free amazon apps love time great play use phone download device work don version android tablet does downloaded store fun access want google works able hd quot update 62 br warranty year product amazon return thanks llc don drive seller new technical hard item refund disk listing 2014 wrote deceptive details years original message pm business receipt know like just really don think want lot little know better ve way thing make things nice doesn right need going look ll screen good feel pretty looks say didn got"

Actually, the tokens "br", "router", "review"," device"," update" seem to be quite important. The token "br" indicates a line break in HTML text, whereas the other ones are easy to understand. Basically, these words do not have much meaning by itself, though they show that the reviews might get abstracted by a couple of tokens. In addition, taking into account the "loadings" of single text documents it is possible to reduce the overall dimensionality among reviews. Hence, in order to avoid bias caused by unambiguous loadings, KMeans cluster analysis has been conducted for a grouping to

five clusters. Thus, the plot below shows on the x-axis the respective clusters and on the y-axis the frequencies of the different “star ratings”. The first cluster is obviously dominated by reviews with quite good ratings, similar to the second one.

However, the third cluster seems to be entirely dominated by very bad reviews, those with a one-star rating. Moreover, this middle cluster represents somehow a break-point considering the last two clusters. In these clusters the proportion of five- and four-star ratings is still high, but the number of middle and bad ratings increases in comparison to the first two clusters.



Hence, relying solely on a term frequency matrix it is possible to separate reviews to distinct groups which should show at least different word frequencies.

- **Further cleaning steps and preparation of the dataset**

The first target of this project is the development of a sentiment classifier; therefore, further cleaning steps are necessary. Thus, it is necessary to remove reviews which do not have much content or which are primarily reflections of the emotional state of writers. Simply, to avoid training the classification algorithm on possible reviews like this one: “It is such a cool product, I like it!!!!!!!!!!”. Consequently, reviews are scanned for non-alpha-numeric or non-English expressions or other special characters. Then, reviews which have more than 80 percent of non-English or non-alpha-numeric expressions are removed. Furthermore, also reviews that show more than 30 percent of special characters in their text are dropped, but also very short reviews which have less than ten words are deleted. As a result, the dataset is being reduced to 98,918 reviews.

• Deep Learning for Text Classification

The design of the sentiment classification algorithm is built upon Deep Learning approaches for Natural Language processing. In this regard, the further process is split in two parts: the first one's target lies in the generation of a Language model which aims to provide so called encodings for the vocabulary of the reviews. The second part leverages these encodings to develop a classification model. The decision to rely on a Deep Learning framework might get explained by the following advantages over more conventional models used in natural language processing:

- Language models provide encodings for words or sentences. These encodings are representations which might capture different semantic meanings and relations among different words, phrases or sentences. The very basic approach of using lexicons/dictionaries to decode the sentiment of text does not account for acronyms or special slangs. Moreover, feature generation by means of document term matrices, as it has been done above, has its limitations. Of course, it is possible to try various tokenization techniques or to extend the vocabulary to n-grams. Though, human language is very fine grained, and thus, such feature generation techniques might not be able to cover the diverse context in which certain words do appear.
- Deep Learning approaches are known to offer larger flexibility in classification tasks, consequently, they are able to capitalize on feature rich representations provided by word encodings.
- Transfer Learning is becoming popular for NLP as well. Thus, pre-trained language models offer encodings developed from large text datasets that capture context of different domains, moreover, the requirement for large datasets and long training times is being reduced substantially.
- The final reason for using Deep Learning in this project is based on the capability to generate text summarizations. Thus, different encodings throughout a neural network could be used to cluster reviews in a more meaningful way.

Ahead of the first step the dataset has to be split in test and training set, additionally a separate set for the language model is necessary to generate word encodings. After randomly splitting the dataset the size of the test set is approximately 12,000 whereas the training set contains 33,000 reviews. The rest of the dataset is dedicated to the language learning model.

• Language Learning Model

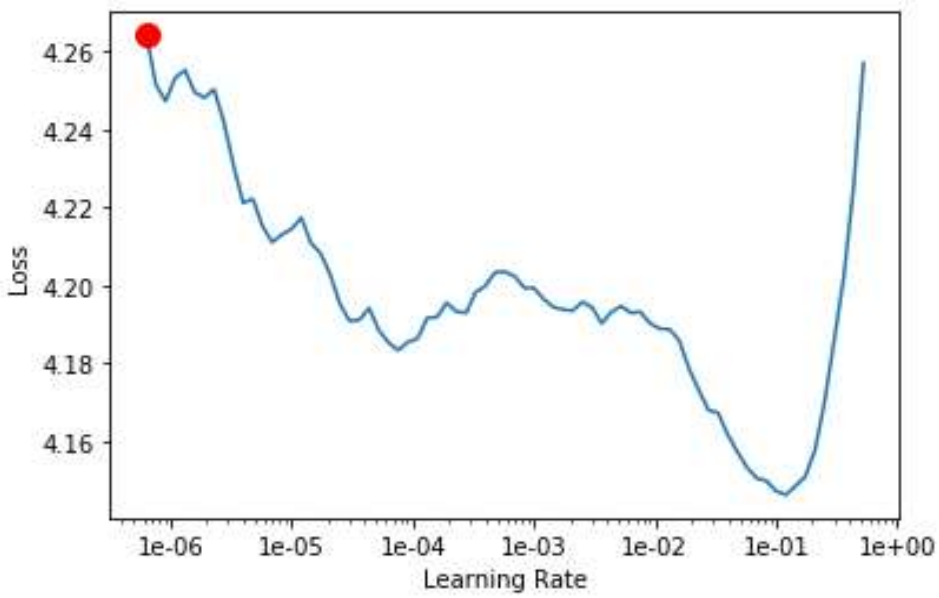
In this step, the pretrained "ULMFit" model featured by "fastai" is being deployed. First of all, a couple of pre-processing steps are required to generate data batches.

- The text is tokenized by removing special characters or HTML snippets, while word contractions or punctuations are being kept.
- Numericalization of tokens to build a dictionary of vocabular and numbers. Actually, the numbers represent the words/tokens which are then fed to the neural network.
- The whole text of the dataset is separated to data batches necessary for processing the neural network.

Thus, for the dataset dedicated to the language model a vocabulary of approximately 33,000 entries is constructed, in addition, the set gets subdivided into batches of size 48 to deploy the ULMFit language model.

This model's task is actually the prediction of word sequences, i.e. to predict the next word with the previous one as input. Technically, the architecture of this network is based upon a Long/Short Term Memory network in which encoding layers are passed forward. However, ULMFit comes with pre-trained word embeddings which have been trained on a large dataset of Wikipedia articles. In order to obtain more specific embeddings for Amazon reviews the model is trained further on the dataset which is dedicated for the language model as well.

For training this model two rather new developments in the field of Deep Learning have been applied: cyclical learning rate scheduling and the learning rate finder. Hence, the plot below shows the learning rate finder and suggested rates highlighted by the red dot.



epoch	train_loss	valid_loss	accuracy	time
0	3.782862	3.689017	0.304888	00:52
1	3.622302	3.583864	0.312590	00:52
2	3.498814	3.540415	0.316205	00:52
3	3.382692	3.530978	0.317090	00:52
4	3.203698	3.547635	0.315948	00:51
5	3.043352	3.583271	0.313255	00:52
6	2.908148	3.624678	0.311251	00:52
7	2.754258	3.670273	0.307939	00:52
8	2.663227	3.702912	0.306286	00:52
9	2.609444	3.718244	0.305493	00:52

After training the network for 10 epochs (upper table) 30 percent of the adjacent words have been predicted correctly. This might not look impressive firstly, though, on the one hand, predicting 30 percent of the time the right word out of a huge vocabulary given the preceding word as input sounds more compelling. On the other hand, the network is not intended to provide solutions for text completion, but to learn more about the semantics and the contextual meaning of words. Thus, the network's first embedding layer provides now review domain specific representations of the dataset's vocabulary.

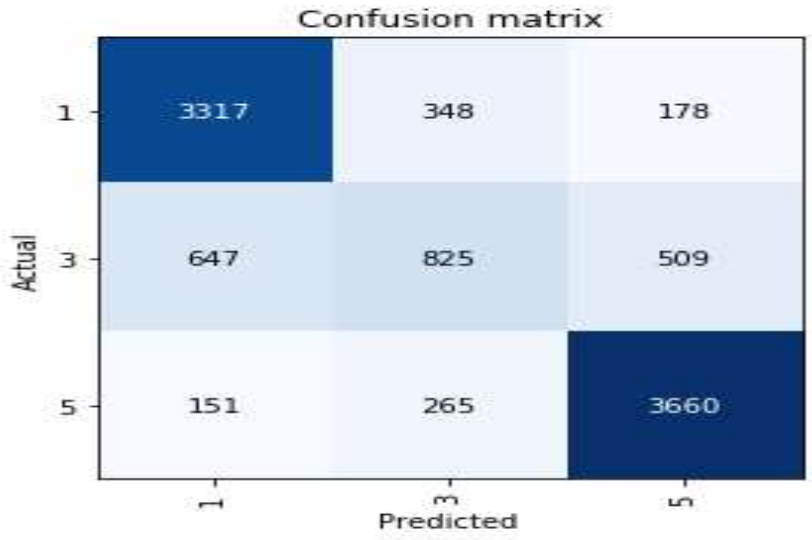
• Classification Model

The encodings obtained from the language model are now used for the embeddings of the tokens in the training and test set. However, the focus of the classification model is rather on predicting the polarity of the reviews than on exact categorization. Thus, the targets which have to be learned by the model are not the exact start rating. Instead, as targets for the model star ratings at the extreme are subsumed to one category, i.e. five- and four-star rating to category five, one- and two-star rating to category one. Reviews with neutral, three-star rating, are being kept. Actually, as noted previously there are likely some behavioural biases which lead review writers to give star ratings. In turn, it might not be able even for humans to guess the rating correctly for some reviews. In addition, the problem of predicting ratings might be considered as being a problem on a continuous scale than on a strict categorical scale.

The network for this task contains the similar long short-term memory encoding as the language model previously as its first module. The second module consists of linear layers to which the hidden layers of the first module are transferred via max-pooling.

In a first stance, only the last layers are trained and then previous layers are being additionally trained in a step-wise fashion. Basically, this routine of gradually unfreezing layers when training NLP models has been tested successfully a couple of times for diverse classification problems. The learning rate finder is employed again to assess the learning rate to start with, moreover, various batch sizes have been tested, but with indifferent results.

After gradually unfreezing the layers and running the learner for a couple of epochs results on the test set are obtained, shown by the following confusion matrix:



	precision	recall	f1-score	support
1	0.81	0.88	0.84	4191
3	0.60	0.42	0.49	2069
5	0.85	0.90	0.87	4324
accuracy			0.80	10584
macro avg	0.75	0.73	0.74	10584
weighted avg	0.78	0.80	0.79	10584

Actually, an overall accuracy of 80 percent in predicting the three classes has been scored, while f1-scores are quite reasonable for the polarity classes, i.e. “5” – containing five- and four- star ratings – and “1”— containing two- and three- star ratings. Thus, it is remarkable that most miss-classifications occur in predicting class three as can be seen on the confusion matrix. The reasons for this, are obviously on the one hand, based on biases of review writers, and in this regard, on the fact, that the problem might be better stated as being continuous than categorical. On the other hand, by subsuming the polarity classes an imbalance for the middle class emerged, i.e. only a portion of about 20 percent.

However, for possible business applications, if for instance, emails or other unlabelled reviews from clients about products have to be classified, the achieved results of the classifier seem to be reasonable.

- **Review Summarization – leveraging the encodings**

Another important aspect of Natural Language Processing lies in the generation of summaries. Hence, in this application it might be of importance to generate summaries of let’s say all badly rated reviews of the Amazon Kindle product. Overall, it might be helpful for product managers to have some kind of numbers, for instance, how is the sentiment polarity scored for reviews about products in question. Though, this represents rather a numerical description, but it won’t be possible to get more insight about what people actually dislike or about key aspects of a product customers appreciate very much.

Hence, text summarization might provide product or marketing managers more insight about what customers precisely think.

So, as an additional by-product of the language model the word embedding layer is used to generate summaries about the reviews. Specifically, an extractive text summarization technique gets deployed. Finally, reviews or a subset of reviews should get summarized by a couple of most representative reviews. These ones, should nevertheless be different to each other in order to represent distinct informational content.

Basically, the embeddings of words contain important information about the context in which the word is used, therefore, it is possible to model the relation of different words. In this regard, by "measuring" the closeness between embeddings renders word clusters. So, for example, searching the ten closest embeddings to the embedding of the word laptop gives back: keyboard, backpack, netbook, motherboard, macbook, laptops, notebook, desktop, tablet, computer. This shows, how powerful the concept of embeddings is working, and thus, it might be useful to cluster reviews as well.

- **Review Extraction**

Leveraging the word embeddings trained by the language model which have a 400-dimensional space, it is necessary to aggregate them on a "review" level. Basic approaches for aggregation, would either be averaging all the word embeddings of a review or finding the maxima of all embeddings or doing both. In this project, the embedding vectors of words within a review get averaged. As a result, every review is represented by a 400-dimensional vector.

Now, it is possible to deploy dimensionality reduction techniques as it is often done in Latent Semantic Analysis. So, KMeans clustering is applied not only to reduce the space of the reviews' aggregated embeddings, but also to find clusters with distinct informational content. Subsequently, for each cluster the review which is closest to the cluster's centre is being selected as the most representative one. In plain words, let's say out of a subset of 5,000 reviews ten get selected which are most representative for each of a different cluster.

For example, about the game "candy saga" 166 reviews have been written, hence, to extract the ten most representative reviews, a cluster analysis is run with aggregated embeddings as input. As a result, the ten reviews below are being selected as most representative due their vicinity to the respective cluster centres:

At first it's fun and pretty mindless, but as you progress, you have to buy the powerups to beat the levels. I got stuck on level 29 for days. It took 15-20 tries. There ARE ways to beat them, but it's mostly just how lucky you get that determines whether or not you win that round. Pair all of this with only 5 "lives" a day and it's incredibly challenging. But, of course, you can buy more lives at .99 cents a pop. It's just another game trying to get your money. But, if you like a challenge and vow to spend no money, it's a fun game to pass the time or wind down with at night. The feature of connecting your Facebook and being able to see how far your friends progress is a fun addition. And doing this lets you switch back and forth between devices without losing your progress. I'd give it 5 stars if it weren't tailored to force you to buy in-game powerups and if it didn't cut you off after 5 tries.

##

Fun, but be careful, it'll get you and keep you awake for days! Could lose lower lip from concentrating too hard. Have fun!

##

This has to be the most addictive game I've ever played. I thought it was a kid's game but I just can't put it down.

##

This game is highly addictive. Lots of fun and can be very competitive to see what all your friends are doing as well. Very happy that it became available on the Kindle.

##

I LOVE PLAYING BUT SOME LEVELS IT TAKES DAYSTO PASS. I WILL NOT SPEND MONEEY TO PLAY A GAME TO BUY POWER-UPS

##

Simple mindless game to pass time. The higher levels are challenging enough to keep you hooked. Minimal crashes and app issues, game runs smooth.

##

At first it's fun and pretty mindless, but as you progress, you have to buy the powerups to beat the levels. I got stuck on level 29 for days. It took 15-20 tries. There ARE ways to beat them, but it's mostly just how lucky you get that determines whether or not you win that round. Pair all of this with only 5 "lives" a day and it's incredibly challenging. But, of course, you can buy more lives at .99 cents a pop. It's just another game trying to get your money. But, if you like a challenge and vow to spend no money, it's a fun game to pass the time or wind down with at night. The feature of connecting your Facebook and being able to see how far your friends progress is a fun addition. And doing this lets you switch back and forth between devices without losing your progress. I'd give it 5 stars if it weren't tailored to force you to buy in-game powerups and if it didn't cut you off after 5 tries.

##

I am a die hard bejeweled player. My sister challenged me to play this game. It is very addictive. Awesome.

##

Can't stop playing

To much fun I love this application really love to play it

This is a blast but pissed me off at same time

##

my older children got me hooked. Even though I find it frustrating at times, it is such a fun and relaxing game. I also installed it on my Kindle Fire and take it everywhere with me. It would be nice to let us send more than 50 gifts as I play with 450 other people on Facebook, It would be "really" nice to have a few more moves and extra lives. Otherwise it is an-awesome game.

Based on these reviews, some information about customers' product experiences can be gained. Aside sentiment expressions like "awesome" or "waste of money", some other information about product features like "Facebook" connection or "using it on more devices without losing progress" or

information why people used the game “my sister brought me”/”my children..” are extractable. Hence, based on word embeddings by means of cluster techniques it is possible to extract information which could provide relevant summaries product or marketing managers.

• Conclusion

In this project modern NLP techniques have been conducted exploring solutions for a) sentiment classification and b) text summarization. Although, the dataset contains only a rather small subset of Amazon reviews it is possible to apply “data hungry” Deep Learning approaches which is feasible due to the concept of transfer learning. Hence, the presented approach does not only provide suitable solutions for producers when selling their products via Amazon or Amazon retail managers, instead, this approach might get deployed for many other business applications as well. For example, an insurance company which crawls emails from their customers or Q&A queries from their homepage, might be able to generate sentiment scoring (negative – neutral – positive) by training a Deep Neural Network on only a small subset of labelled data. Moreover, it could then easily be used to summarize customer statements about specific products.
Thus, the presented Deep Learning approach has been able to solve both problems. In addition, the high flexibility of Deep Learning algorithms seems to provide solutions to handle these “fine grained nuances” of human language as opposed to traditional feature engineering approaches in NLP. Thus, the deployment of pretrained language models to train domain specific text, i.e. the language used in the context of financial products is likely to be different to the jargon used in discussions about fashion; can provide a strong advantage to modelling.
Nevertheless, in this project different and maybe more advanced classification models could have been tested on the dataset to obtain higher accuracy. Basically, the architecture of the “ULMFit” model is well suited to text with shorter length, whereas other pretrained models like “BERT” are more powerful for tasks with longer text like classifying newspaper articles or longer legal documents. Though, to improve performance of the classifier the last layer module of ULMFit could have been extended by more linear layers to add more flexibility. In addition, one-dimensional convolutional networks could have been tested as well to provide a better solution to the problem regarding the misclassifications of the imbalanced neutral sentiment class.
The approach to generate summaries by highlighting the most representative reviews is extractive and unsupervised. Thus, next to K-Means cluster analysis other techniques like agglomerative clustering or the Sentence Rank algorithm could have been tested.
Overall, the project has demonstrated a comparably straightforward approach which is capable to classify sentiment contained in reviews and which extracts summaries at one stroke.

-