**Capstone Project 2 – Report**

* **Problem definition – business application**

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 for 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, 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 5000 worst feedbacks of a product. Moreover, from another angle, it could be also quite useful to get a summary of the 5000 best feedbacks in order to figure out “unique selling propositions”.

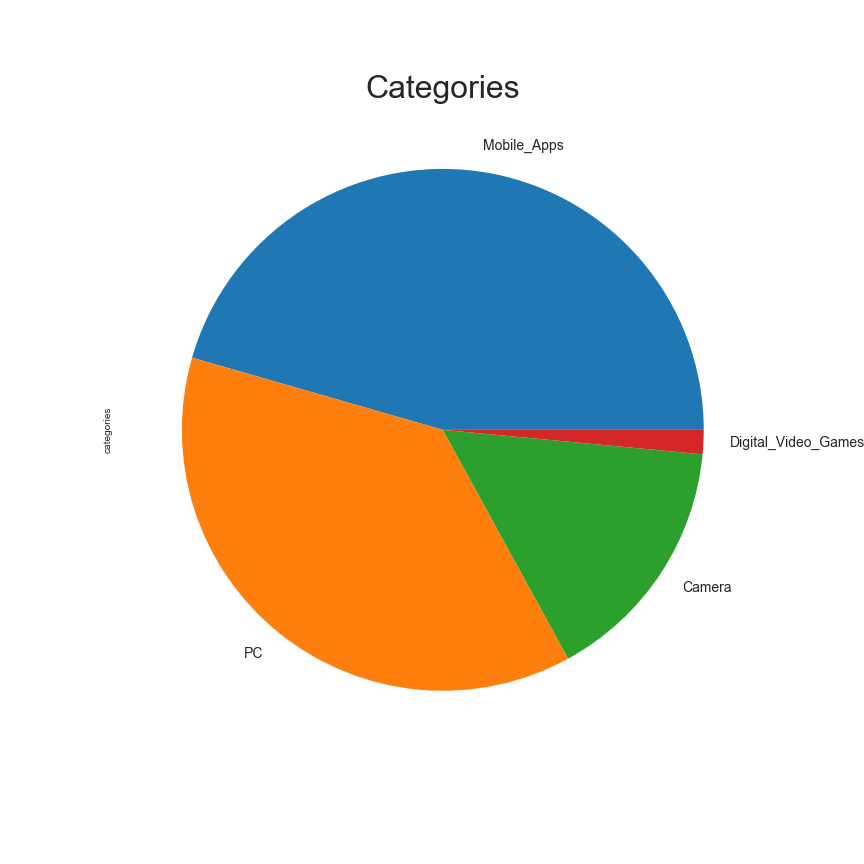
* **Dataset and approach**

Amazon reviews (https://www.kaggle.com/datafiniti/undefined ) of 65 different products are considered as the major dataset. This set contains 28K reviews and provides a labelling by means of “stars rating” and recommendation tagging. The approach to solve the business scope is twofold. First, Deep Learning is applied to the solve the classification task, i.e. to predict the ratings a review 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 1 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 f.i. on datasets like email messaging form clients, there are not any possibilities to give categorical ratings without the need of a human. Thus, step 1 provides an important output which then helps to segment reviews without the need of labelling the data beforehand. Step 2 is basically leveraging the Deep Learning Classifier’s embeddings to generate summaries by means of most representative sentences or reviews.

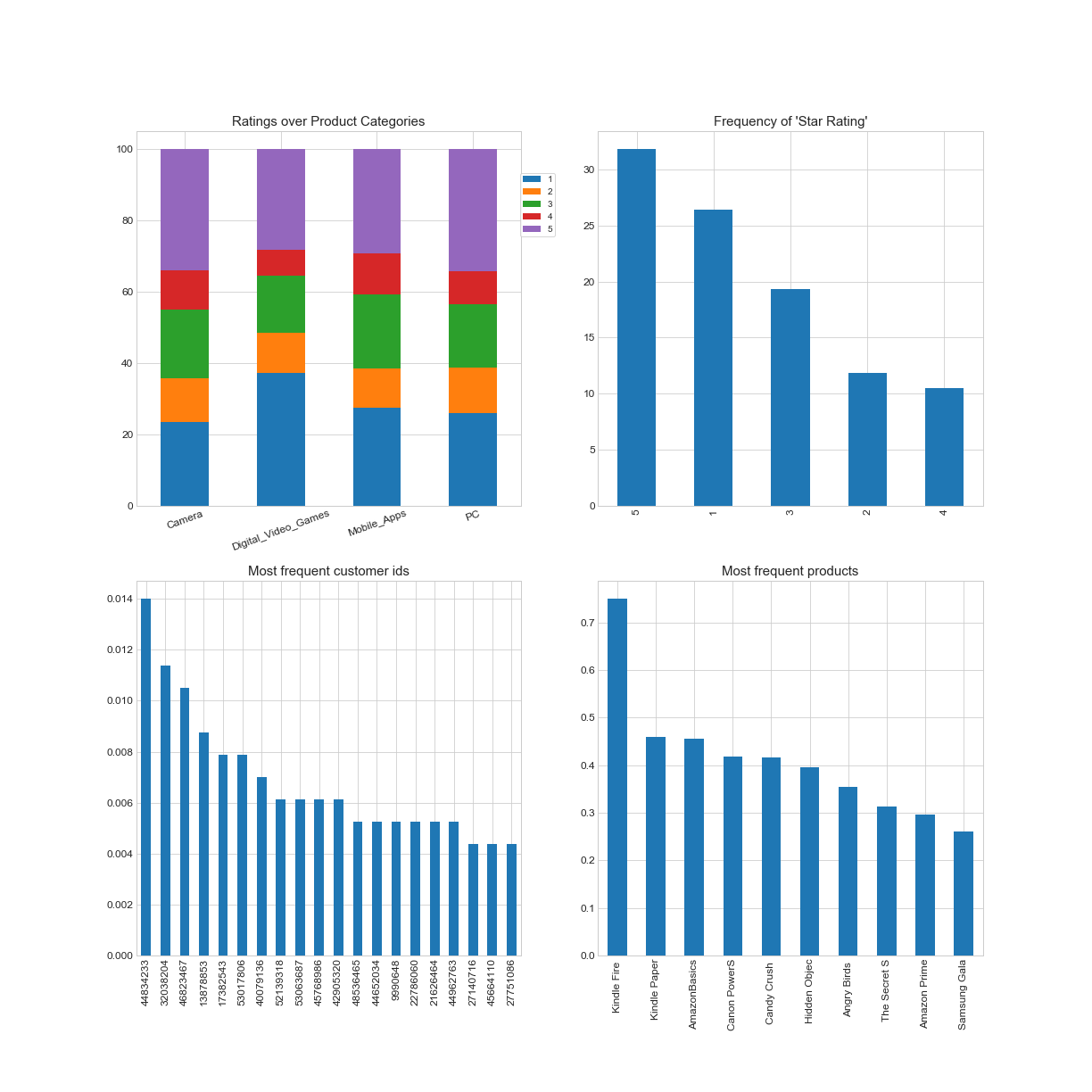
* **Milestones and details**
  + The reviews get described by classical NLP EDA techniques; moreover, other facts of the dataset get analysed, i.e. which products get reviewed the most or is there a trend over time.
  + Deep Learning: a pretrained network gets chosen, i.e. either UlmFit or BERT whose weights get fined tuned on the review corpus by predicting word or sentence sequences.
  + After fine tuning the weights, a final layer is added for the multi class prediction task – i.e. predicting the “Stars Rating”
  + Extractive Summary: after these steps the “encoder” of the network is leveraged by means of “thought” vector representation. Based on this, embeddings for sentences or word phrases are provided. Different techniques of extractive summarization will be applied then, i.e. clustering the embeddings or “Sentence rank algorithm”. So for instance, for each of the 65 products summaries of the most, let’s say 10 % positive or negative “sentiment buckets” shall be generated. Due to the fact that this is an unsupervised learning process, the quality of the summaries will be evaluated following the proposal of this paper <https://www.researchgate.net/publication/220017752_Using_Latent_Semantic_Analysis_in_Text_Summarization_and_Summary_Evaluation>.

* **Data Query**

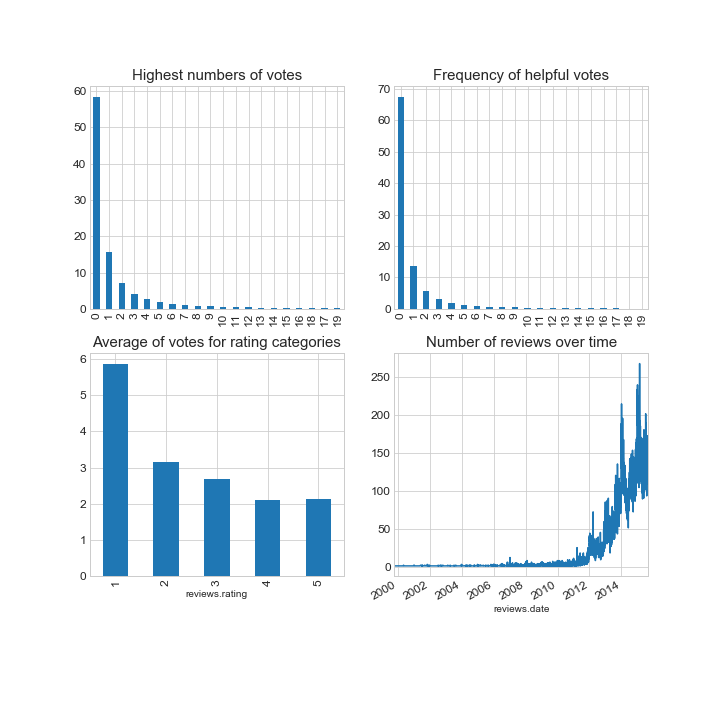
The dataset has been created by the author to generate a representative subset of Amazon reviews. Following the page <https://s3.amazonaws.com/amazon-reviews-pds/tsv/index.txt> which provides links to reviews of diverse categories hosted by Amazon, reviews about “Mobile Apps”, “PC”, “Camera” and “Digital 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 K. To generate a smaller, but still representative dataset, reviews are chosen randomly by a size of five percent. Within these category 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 114192 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 of the review, 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.

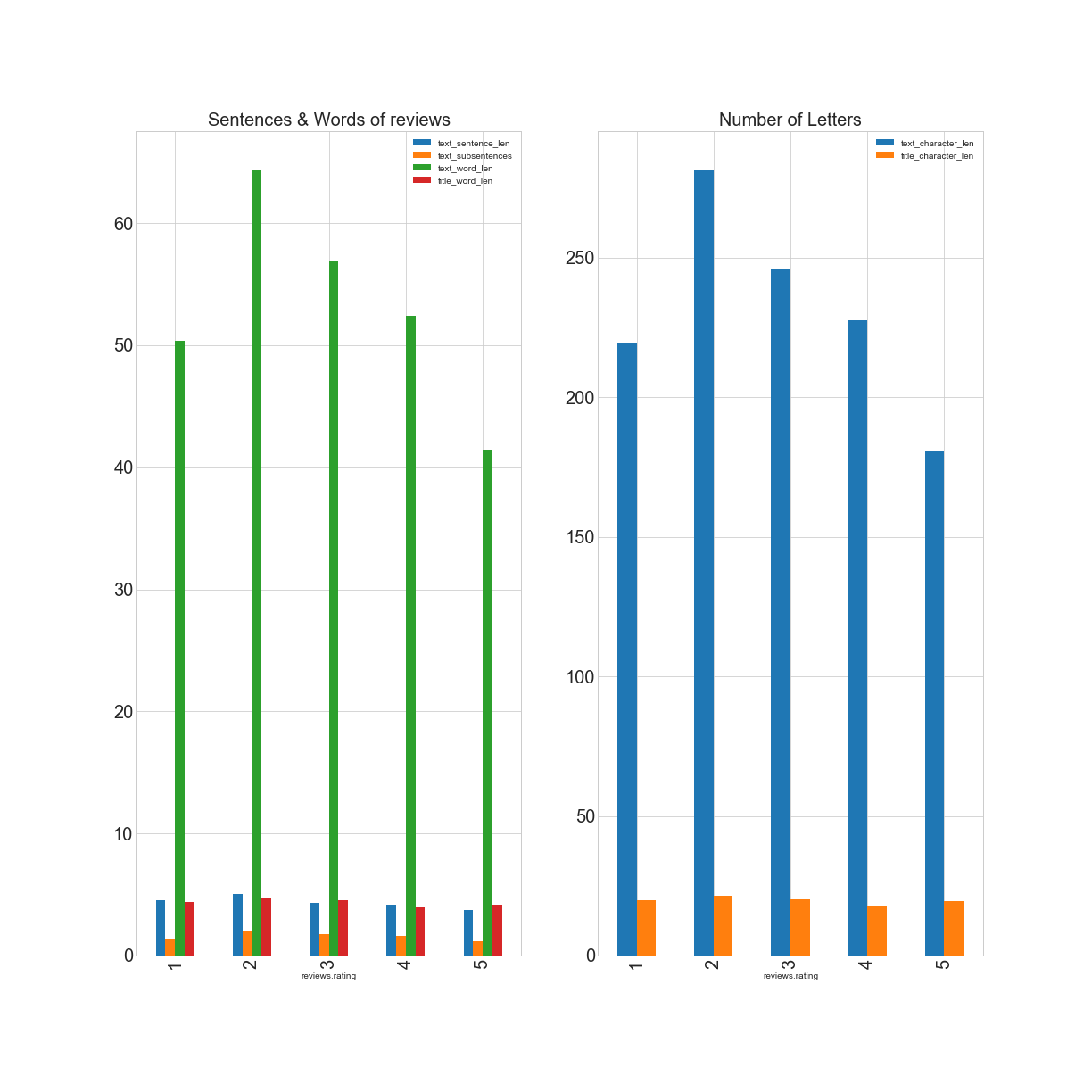


With regard to the distribution of “Star Rating”, there is a considerable bias towards higher 5 Star ratings which is shown on the top right charts. Moreover, a bias towards polarity is obvious simply cumulating the lower and upper tail of rating frequencies: 5 and 4 stars make up more than 40 percent of all reviews and the lower end – 2 and 1 star ratings - sum up to close to 40 percent as well. Instead, the middle – 3 star ratings – shows a comparably small 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 review if people are rather very satisfied or disappointed. Considering the distribution 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 5 and 4 star ratings, but negative reviews (1 & 2 star) are comparably high as it gets depicted by the top right 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 ~ 1.3 K reviews which might seem to be a high number which could be an outlier.

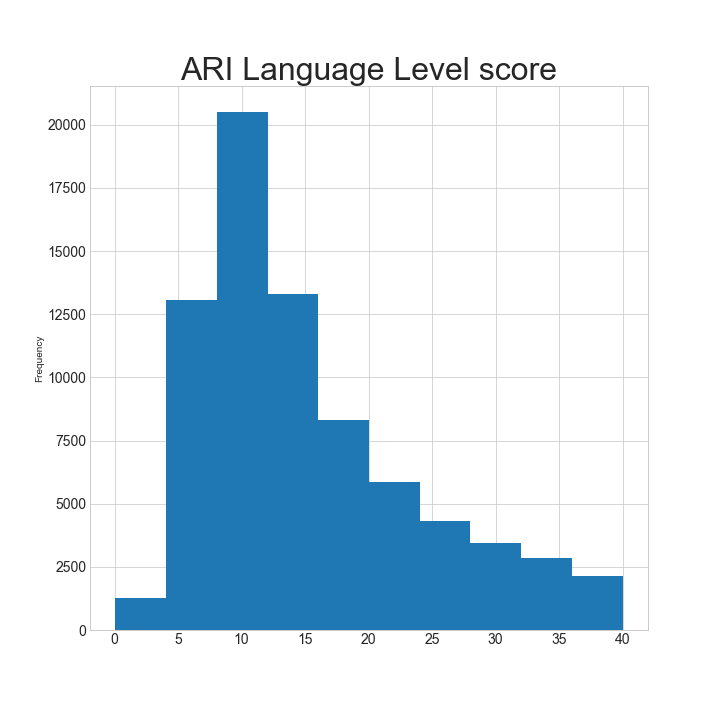


By taking a fast look at the other features provided by Amazon, most reviews did not get any appreciation as it can be seen by the two plots in the upper panel of the figure above. Most reviews did not get any vote at all, while only something more than percent of the reviews got one vote. Nevertheless, 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 purchasing a bad product. A larger part of reviews has been written between 2011 and 2015. The fact that till 2010 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.

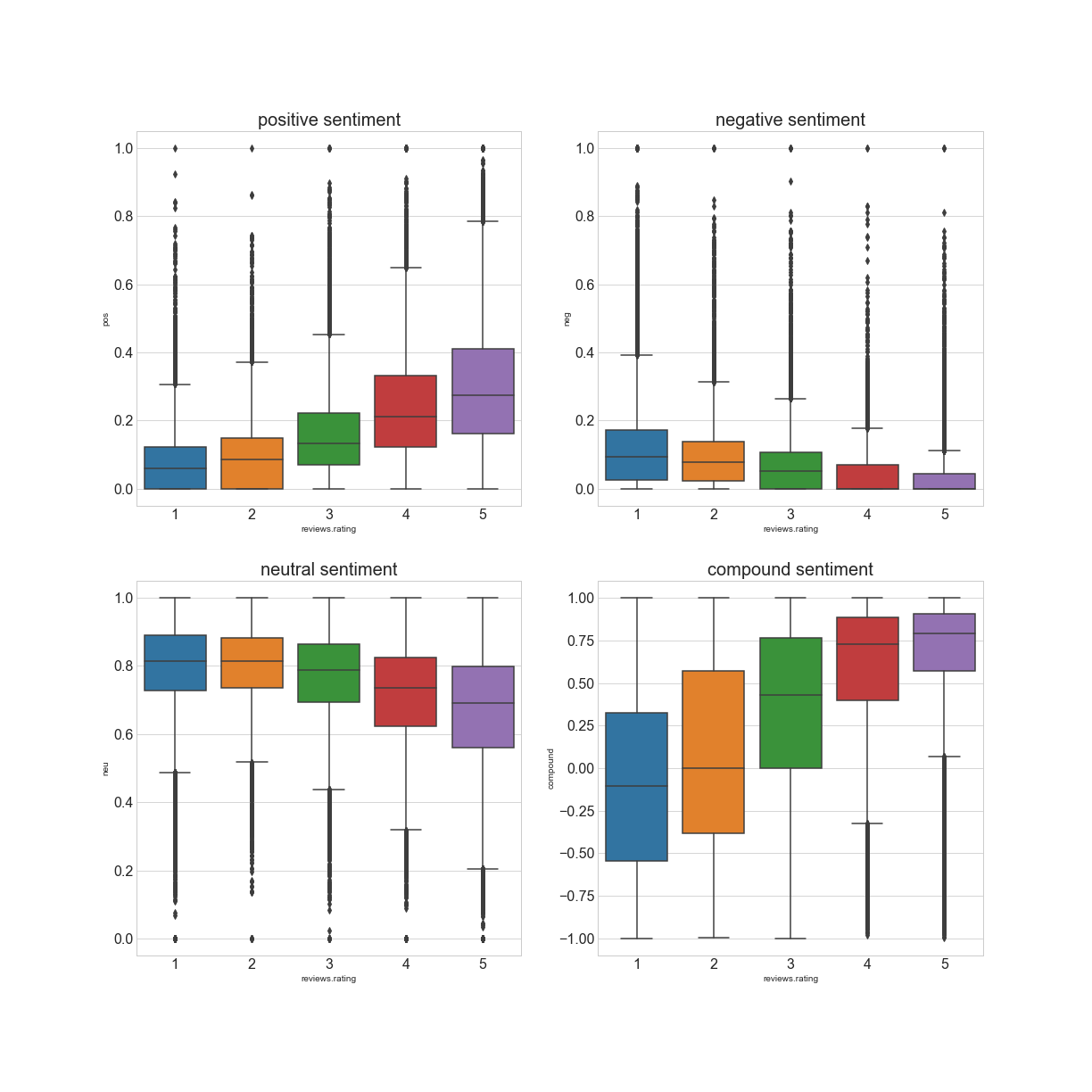
In order to get an impression of linguistic properties the review texts have been processed to extract further features. First, the number of words, characters, sentences and sub-sentences are assessed by very basic tokenization. Thus, the charts below show the average number of words, sentences and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



Overall, medium to slightly badly rated reviews 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. To check the level of the understandability of reviews the Automated Readability Index (ARI) has been applied 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 all reviews with an average of 15.00 and the 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 a day to day style, this ARI statistic reflects an overall sufficient level of understandability of the datasets’ 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 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 with 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 the distributions 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, prediction algorithms that aim to classify sentiment or star ratings this ambiguity of writings might be the reason for biases and remarkable performance problems.

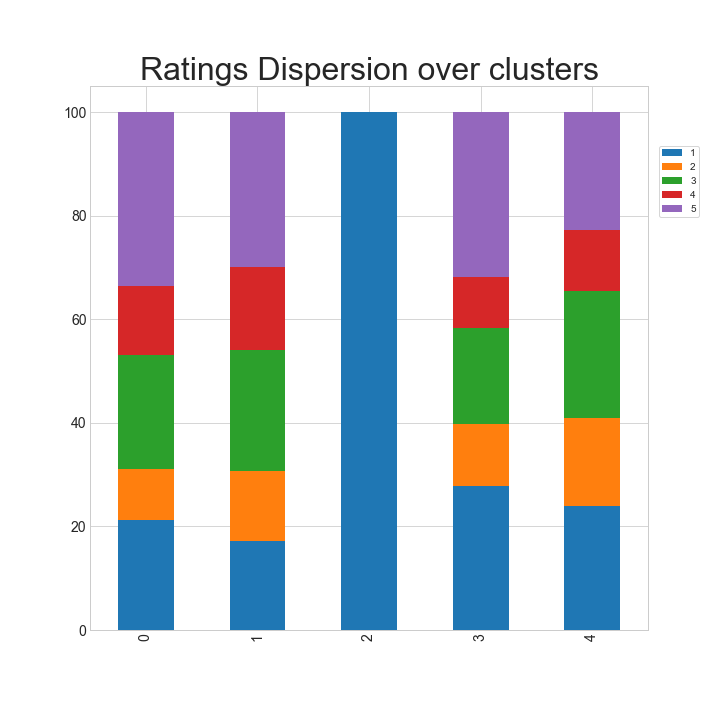


* **Cluster Analysis and Topic extraction**

Besides lexical approaches reviews have been processed by removing stop words – words which do not add explanatory value -, stemming words and by dropping out some special characters. 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.



“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” , “camera”, “game”, ”good”, ”love” ,great appear quite frequent. The token “br” indicates a line break in HTML text, whereas the other ones are easy to understand. Extending this approach of the 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 which represents a suitable technique for sparse matrices. In this regard, text topics are extracted. So, the text below shows the 50 most relevant words of the Amazon reviews. Basically, these words do not have much meaning by itself, though they show that the reviews might get summarized by a couple of tokens. In addition, taking into account the “loadings” of single text documents to these topics 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 to shrink dimensionality further 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, quite similar to the second cluster. 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 in distinct groups which in turn indicates at least to different occurrences of words between these groups.

* **Further cleaning steps and splitting the dataset**

The primary target of this project is the development of a sentiment classifier; therefore, further cleaning steps are necessary. In order to remove reviews which do not have much content or which are primarily reflections of the emotional state of the writers. Simply, to avoid giving the classification algorithm possible reviews like this: “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 % of non-English or non -alpha-numeric expressions are removed. Furthermore, also reviews that show more than 30 % of special characters in their text are dropped, but also very short reviews which have less than ten words are being deleted. As a result, the dataset is being reduced to 98918 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 representation which might capture different semantic meanings and relations among different word, 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 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 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.

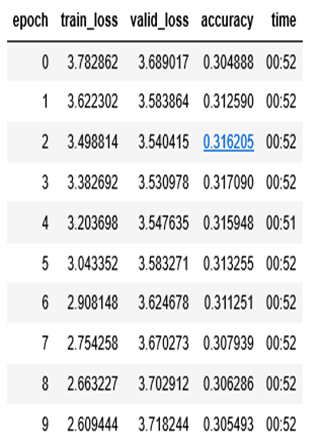
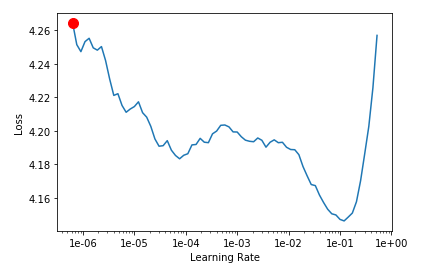
* **Language Learning Model**

In this step, the pretrained model ULMFit model from the fastai api gets 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 taken care of
* Numericalization of tokens to a 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, with the dataset dedicated to the language model a vocabulary of approximately 70 K entries subdivided into batches of size 48 are being constructed to deploy 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 decoding layers that are passed forward. However, ULMFit comes with pre-trained word embeddings which have been trained on a large dataset of Wikipedia articles. However, in order to obtain embeddings for Amazon reviews the model is trained further on this dataset as well.

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

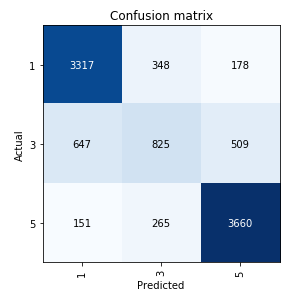


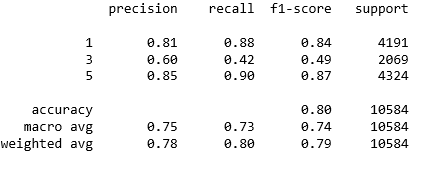
Hence, after training the network for 10 epochs 30 % percent of the adjacent words have been predicted correctly. This might not look impressive on first place, though, on the one hand, predicting 30 % of the times the right word out of 70,000 vocabularies given the preceding word as input sounds more impressive. On the other hand, the network is not thought 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**

These 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 is not the exact start rating. Instead, as a target 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, remained. 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 for a human 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 states of the 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 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:





Actually, an overall accuracy of 80 % 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. Moreover, by subsuming the polarity classes and 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 encoding**

Another important aspect of Natural Language Processing lies in the generation of summaries. Hence, in this application it might be of uttermost importance to generate summaries of let’s say all badly rated reviews of Amazon Kindle product. Overall, it might be helpful for product managers to have some kind of numbers, f.i. 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 what are key aspects of a product customers appreciate very much. Hence, text summarization might provide a product or marketing managers more insight about what customers precisely think. So, as an additional by-product of our language model the word embedding layers are used to generate summaries about the reviews. Specifically, it an extractive text summarization technique gets deployed. As a target, 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.

* **Review Extraction**

Leveraging the word embedding trained by the language model which have 400-dimensional shape, it is necessary to aggregate these embeddings for the review. Accordingly, the embedding vectors of all the words in 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 the usual approach in Latent Semantic Analysis. For this project, the KMeans clustering is applied to reduce the space of the reviews, subsequently, for each cluster a review is select which is closest to the cluster’s centre. In plain words, let’s say out of a subset of 5,000 reviews ten get selected which are most representative for different clusters.