TECHNICAL UNIVERSITY OF MUNICH

TUM Data Innovation Lab

"Customer Feedback Analytics Platform"

Authors	Sarah Larbi, Saransh Agarwal, Henrik Sergoyan, Nikolaus Pinger
Mentor(s)	Dr. Matthias Böck, Linda Le, Iwan Uswak, Tim Lorsbach (FELD M
Co-Mentor	M.Sc. Olga Graf
Project Lead	Dr. Ricardo Acevedo Cabra (Department of Mathematics)
Supervisor	Prof. Dr. Massimo Fornasier (Department of Mathematics)

Jul 2021

Abstract

Retail is one of the domains which has been transformed heavily due to digitization. To be aligned with the customer needs and to act upon them, online reviews play a huge role.
Unfortunately many companies underestimate the business value laying in customer feedback. However, the business executives are gradually realising its potential for the development of new products and improvement of the existing ones. In the modern age where huge amounts of data are collected through different streams, there are new possibilities that have opened up, in terms of analyzing this information and drawing maximum business value out of it. Traditionally, the focus was primarily on measures like the Net Promoter Score (NPS). Whereas, textual review analysis was either overlooked or done manually by hand, but with scale, this procedure becomes too cumbersome. Leveraging technology to automate the process of not only the data collection, but also data analysis is a necessity to get real insights about a customer's emotions and feelings rather then just a quantitative rating. Our initial assumption was corroborated by interview partners from the industry who could be potential customers of this solution.
After an extensive market research and competitor analysis, we decided to build an end-to-end application which would automatically analyze the customer feedback and present it in an easy and comprehensible format like graphs and charts. So that the insights can be drawn from it easily without going through an extensive cycle of reading the reviews manually. Our solution is robust, scalable and provides the user with the insights in an easy-to-use dashboard.
The two tasks that we identified which would act as a backbone of this application are: Sentiment Analysis and Topic Classification. Sentiment Analysis is important to filter out negative reviews, which brings the most important insights, as stated by our interview partners. Whereas, Topic Classification would tag the reviews to some pre-defined category (like size, shipping, etc) so the problems can be tackled individually. To provide the user with good insights, we leveraged Natural Language Processing techniques. We developed some domain specific models by training deep learning models on a dataset of reviews.

Contents

1 Introduction

The world is becoming more and more digital and this trend can be seen in various domains. One domain where the path from analog to digital is making a huge impact is retail. The transfer from a purely offline retail market to e-commerce shops is significant. Retail e-commerce sales worldwide have been rising from 1336 billion U.S. dollars in 2014 to 4280 billion U.S. dollars in 2020 and are estimated to rise to 6388 billion U.S. dollars in 2024 [**1**]. Due to the lack of interaction between the seller and customer, online reviews are the most important source of feedback for a brand to improve its products and processes. Since the variety and quantity of customers can increase in the online setting compared to local retail, so do the reviews. Large corporates often get an extremely high amount of reviews which are impossible to track and analyse by hand. Especially challenging in this case is the free text feedback. Our market research showed that there is a clear demand in the market for a useful customer review analytics tool, which is not filled by current solutions.

2 Problem definition and goals of the project

In a study from February 2021, 152 digital marketing or customer experience decision-makers in the US were asked about their priorities focused on customer experience. 81% agreed that improving the ability to use and act on customer feedback and data is among the highest priorities. Further nearly two-thirds of decision-makers see unstructured feedback as the key to richer insights [**2**]. The main problem for people working with customer feedback is that a manual analysis is too time consuming for an uncertain business outcome and an automatic analysis with competitor products doesn't reach the desired quality, especially in free text analysis. That was also displayed in our interview findings. Therefore the goal of this project is to identify the exact underlying problem when dealing with customer feedback and understand the specifications for potential solution. Further building an end-to-end solution that solves this problem and helps companies better analyse their product reviews was the main goal.

2.1 Process and Agile Framework

To achieve a useful result in the end we relied on two frameworks. Firstly we made use of the Data Thinking approach, which is an overarching process for data strategy which is based on Design Thinking principles. Secondly we used the agile development principles (SCRUM).
The Data Thinking process consists, similar to the Design Thinking process, of different stages. Overall it can be divided in the problem area and solution area. In both areas the process follows the idea of first following an exploratory approach by generating multiple different problem definitions in the discover phase or multiple different potential solutions in the development phase. In the second stage of each area the overall goal is to narrow down the findings and agree on a concise problem statement in the problem area and a productive solution in the solution area. The main goals of this process include a deep understanding of the problems and collaborative prioritization of common objectives and an early involvement of relevant stakeholders. Through the short development cycles

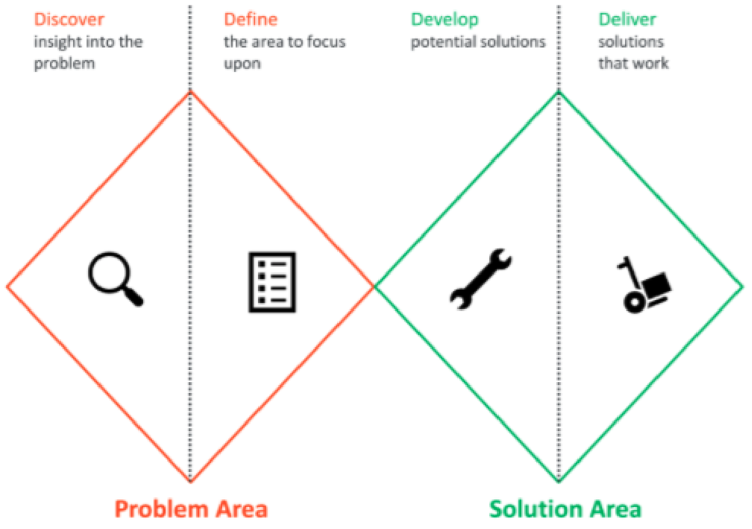


Figure 1: Data Thinking approach

one can create a MVP (minimum viable product) and then iterate quickly and react to unforeseen changes. By having the end goal of developing a fully functional product in mind we followed industry's best practice of agile development. The agile development framework SCRUM [**3**] helped us to manage the end-to-end process by prioritizing different features, iterating over existing solutions and improving the MVP incrementally. Our development phase consisted of 3 two-week sprints to realise the product. Those phases were accompanied by a product backlog, a retrospective meeting and further sync ups within the team.

3 Business Modelling

3.1 Tools

To master the complexity of an end-to-end-product we used various project management tools to help us stay on top of the project. For communication we used an internal Slack group. To keep track of our backlog we used Trello and to brainstorm together and guide us through our meetings we used a Miro board.

3.2 Discovery Phase

The first stage of the Data Thinking process was about exploration. The overarching goal was to identify potential customer needs and problems, to find out which new services, products and business models can be build based on the findings and data. In this stage we conducted three tasks. A quantitative research, which we will present in the dataset chapter, a qualitative part, which we will present in the next section (3.2.1) and a desk research which mainly consisted of a literature review for possible solutions and a competitors analysis.

3.2.1 Interviews / Empathy Map

To identify the problems and pain points of customers when dealing with customer feedback we first developed a questionnaire. The goal of the questionnaire was threefold. First we wanted to understand the status quo of the interview partner. E.g. how are they dealing with customer feedback? Whats the quality and quantity of the feedback? Second of all we wanted to find out what technical tools they already use to analyse the feedback and what characteristics and functionalities make a great and useful product. In a closing question round we wanted to creatively brainstorm what the most pressing problems are and what else has to be considered. To actually conduct the interviews we acquired with the help of FELD M more then 6 Interviews and analysed and documented the results. Our interview-partners came from various industries, various company-sizes and product-categories. To analyse the finding we made use of the methodology of empathy maps (see Appendix), where we captured the pains and gains and analysed what the interview partner said, thought, felt and did. The main findings of our interviews are the following:

- **Time** is crucial. It is the number one reason why companies do not analyze free text feedback manually and therefore the main advantage of an automatized analysis. One of our interview-partners reported that it takes her 4 hours to analyse 300 reviews
- **Quality** of current solutions is too low. Currently the competition doesn't meet industries expectations especially in the domain of free-text analysis. One of our interview partners reported that a competitor shows word clouds which mainly consisted of stopwords (and, the, that, etc.) and are therefore useless
- **Business Value** is everything. To open up resources in a company, both financially and in time and work-force allocation, the business value has to be proven
- **Knowledge** is scarce. Modern Natural Language Processing techniques are needed and marketing departments who mostly work with the feedback neither have the technically skills nor can estimate the feasibility

3.3 Define Phase

After opening up the space of potential problems and customer needs, the Define Phase was mainly about narrowing down to a certain use case we want to tackle. Therefore we identified the relevance and feasibility of the use cases and decided which one should be given the highest priority. From the various interviews that we conducted the interview partners from the fashion retail domain had the most promising use case, since they were very interested in using a similar solution, they felt a huge pain in analysing free text reviews and had a suitable product portfolio. The main reasons why they would be a perfect customer for our tool are that they have many different products and a lot of online reviews and therefore cannot handle it by hand. They have clear expectations of what benefits a solution would bring them, such as addressing the departments with only the necessary reviews that they care about (e.g. logistics, product-quality, etc.). Further things like seasonality or product improvements over different seasons can be analysed.

They want to be quickly informed as soon as a product problem occurs and that would only be possible with an automated analysis.

3.3.1 Feature Prioritization and User Story Mapping

In various iterations and brainstorming sessions we agreed on a certain set of features and defined the way how our user is supposed to interact with our application. First we started with a "Design Studio", in which we drew ideas on paper how our application could look like. Later we gathered feedback in a second round of interviews. After having combined our initial ideas with the feedback we continued with "User Story Mapping", where we visualized the workflows and needs of the potential user. The user story map consists of the main activities on the top and actual tasks/features below that. A detailed overview of the features and the corresponding prioritization into sprints can be found in the appendix . The user story should start with a possibility to login to our tool to account for privacy concerns. Furthermore, some admin setting should be available to the user, which will gain more importance in case a multi-user option is added to the tool later on. That admin settings include functionalities like uploading new reviews and having an overview of the account. The next step in the user story map is the analysis, which is the core of our application. Within this field there are different features integrated, such as sentiment analysis, topic extraction, NPS /rating time line, various filtering options and overviews of the reviews to just mention a few. Further we thought about notification settings, which will become more important as automatic importing functionalities are applied but are not included in the current version. The last step in the user story would be an export functionality, either as CSV, powerpoint or the graphs itself.

3.3.2 Wireframing

The wireframing is part of the ideation process and therefore constitutes a bridge between the end of the define stage and the beginning of the develop phase. This stride engulfs three important steps :

- **Sketching** We developed some hand-drawn sketches that we refined along many stages. We first started with a brainstorming session where everybody first jotted down what he/she imagined the tool to be and what features and functionalities it would have, in a very broad sense. We discussed our sketches and we started to pinpoint what the general architecture of our dashboard is.
  We then focused on more detailed sketches where the features and the user-story-mapping is more precisely defined. Of course, these features were inspired from the discussions that we had with our interview partners. Furthermore, we developed different versions of the sketches to get feedback from our previous interview partners that were gracious enough to grant us some of their time and expertise.

- **Prototyping** When we settled on specific versions of the sketches. We used a prototyping tool, *Figma* [**4**] to transform those hand-drawn sketches into a sleek and professional-looking design.

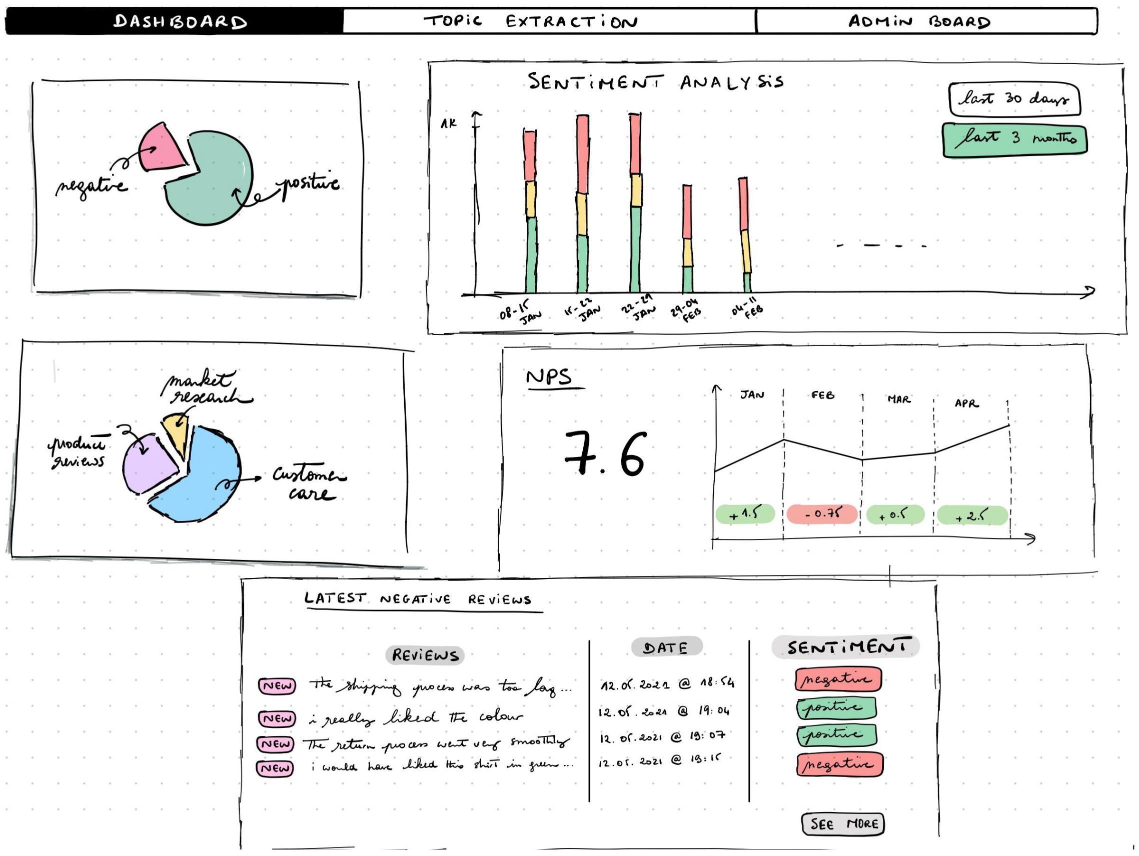


Figure 2: Sketch of the landing page of the dashboard

- **Interview feedback** We conducted meetings with some of our previous interview partners in order to showcase our designs and discuss the features. We focused on pin-pointing their top-priority attributes and narrow down on their ideal tool depending on their use-cases. Although the industries involved were all very different and the company scales varied a lot, we especially kept notice of the points that were frequently brought up and we were able to not only further refine our design because of that but also to lay down all the requirements for the models that will need to be developed.

4 Dataset

Review data is highly confidential and convincing a company to provide us with their personal data for a proof-of-concept is infeasible. Therefore we used the publicly available Amazon Review Data [**13**]. The Amazon data contains 233.1 million reviews across 29 different categories from May 1996 to October 2018. Some of the categories are Amazon Fashion, Software, Books, Automotive, Electronics, Digital Music, etc. For each category, we have the reviews as well as the product metadata. For the purpose of this project, we will focus on the review data from the fashion category, which consists of 883,636 reviews. One good thing about this dataset is that the number of missing values and unexpected

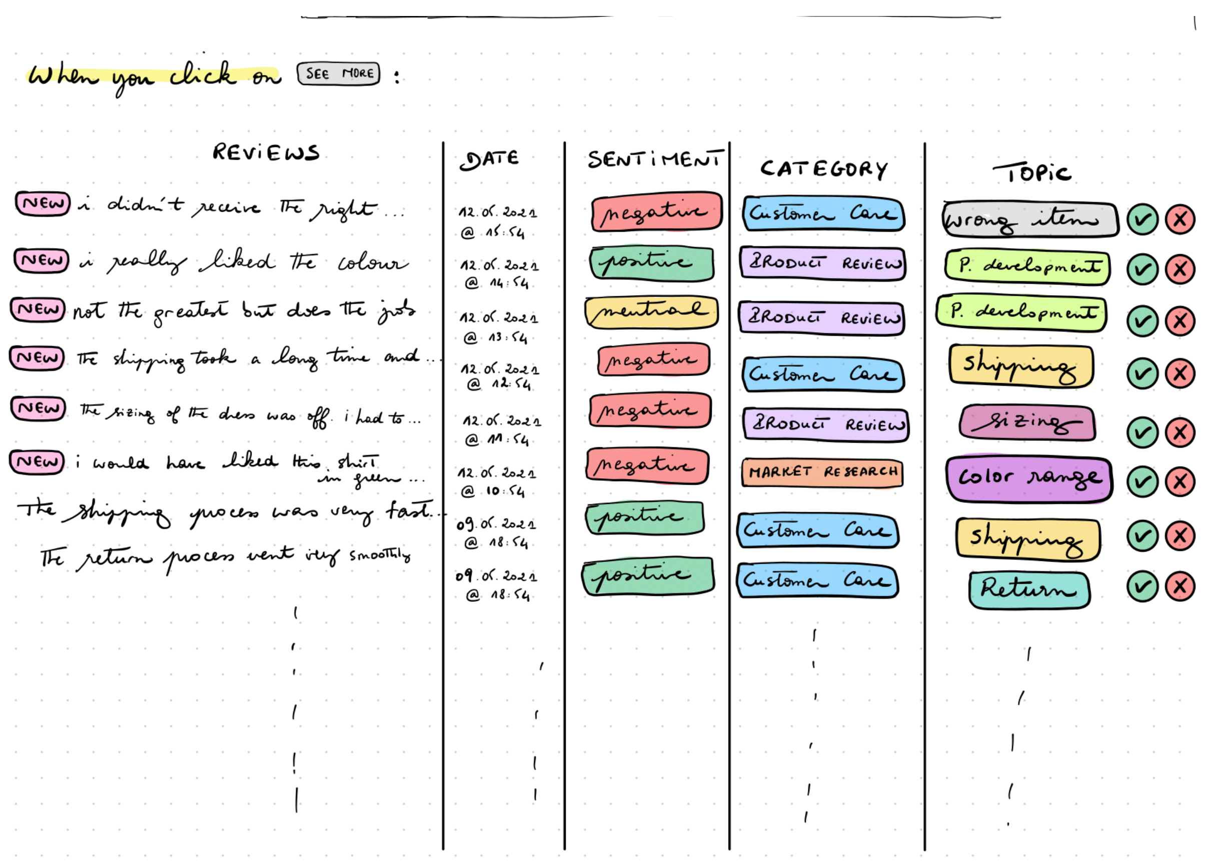


Figure 3: Sketch of a second screen where all the reviews are grouped

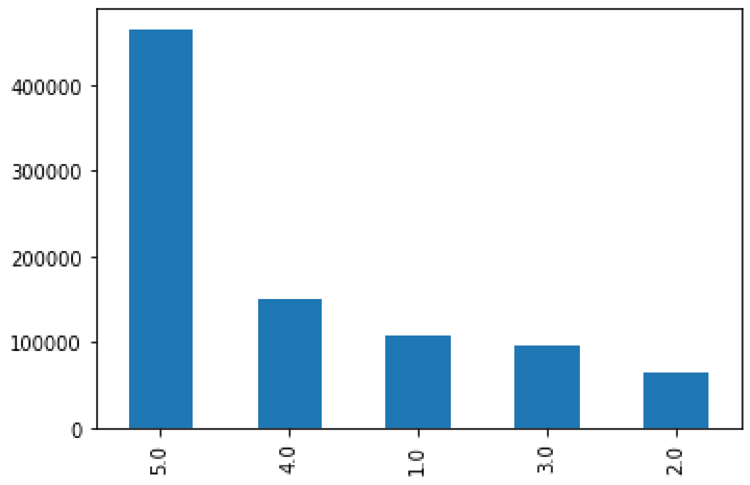
outliers due to data corruption is little to none, as this dataset has been very carefully curated.

4.1 Statistical Data Exploration

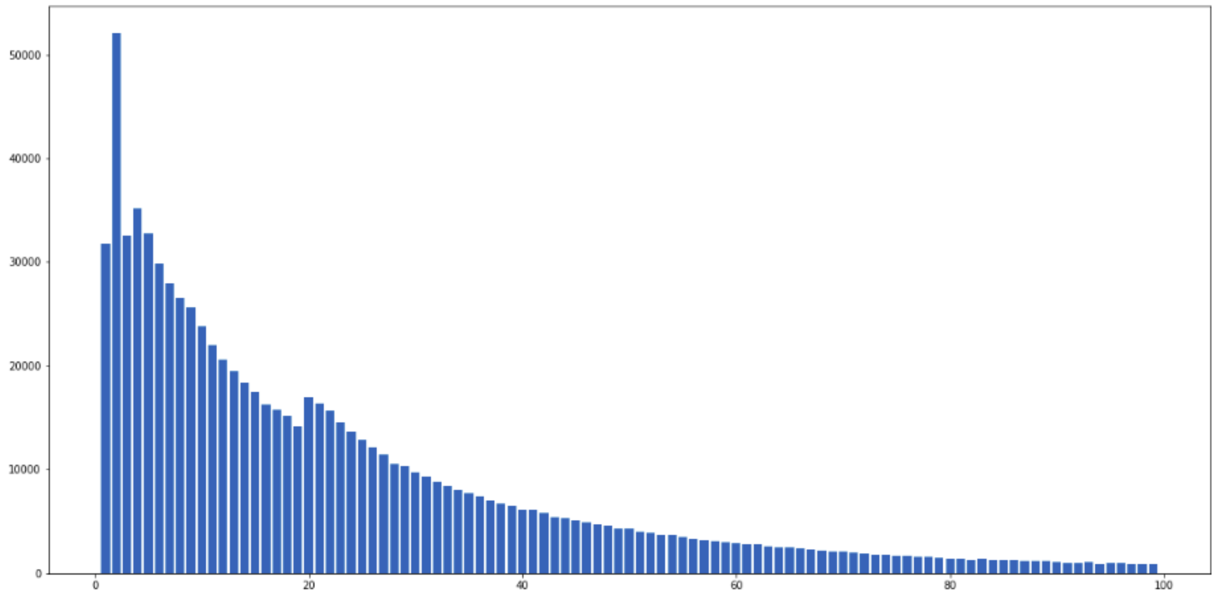
Before starting the project, obviously, we ran some analysis on the dataset so that we know what kind of problem we are dealing with. Note: The analysis was run on the subset of Amazon data (Fashion Dataset). For more information about the dataset, refer to the appendix.
There are certain columns that need an extra attention. Now, let's explore those columns.

4.1.1 Overall Rating

For the sentiment analysis, the overall star ranking(Rating by users from 1-5) is the most important measure for polarity. As can be seen in Figure 2 the classes are imbalanced towards the positive ranking (5star). This is an important thing to note, as for our classification case, getting a decent score on negative reviews is important for us.



(a) Column "overall" distribution [in number of stars from 1 to 5]



(b) Distribution of number of words in the reviews.

4.1.2 Review Text

Similarly, the column Review Text(reviewText) is of particular importance to us, as it holds the actual reviews. We ran some analysis on that column as well. Some of the interesting statistics were :
The **mean word length** in reviews is : **28.617**
The **median word length** in reviews is : **17.0**
Most of the reviews had 2-4 words. Very few reviews had more words present.

4.1.3 Final data taken into consideration

Therefore, we narrowed it down to 5 different data columns that we are incorporating into our solution. **The Review Text, Review Time, ProductID and ReviewerID.**

5 Tech Architecture and Models

The goal of our project was to build a working prototype within the Google Cloud (GCP)[**14**]. For this we had to decide which components needed to be built and which components could be used out of the box (tools offered within the GCP).

Now we briefly describe the tech components used in this project and start with introducing the terms Frontend and Backend.

Frontend : Everything that the end user sees and interacts with falls under the category of Frontend.
Backend : Everything that is abstracted out and runs behind the curtains in an application would fall under the umbrella of Backend. This includes the databases, the computations and any other resource intensive task.

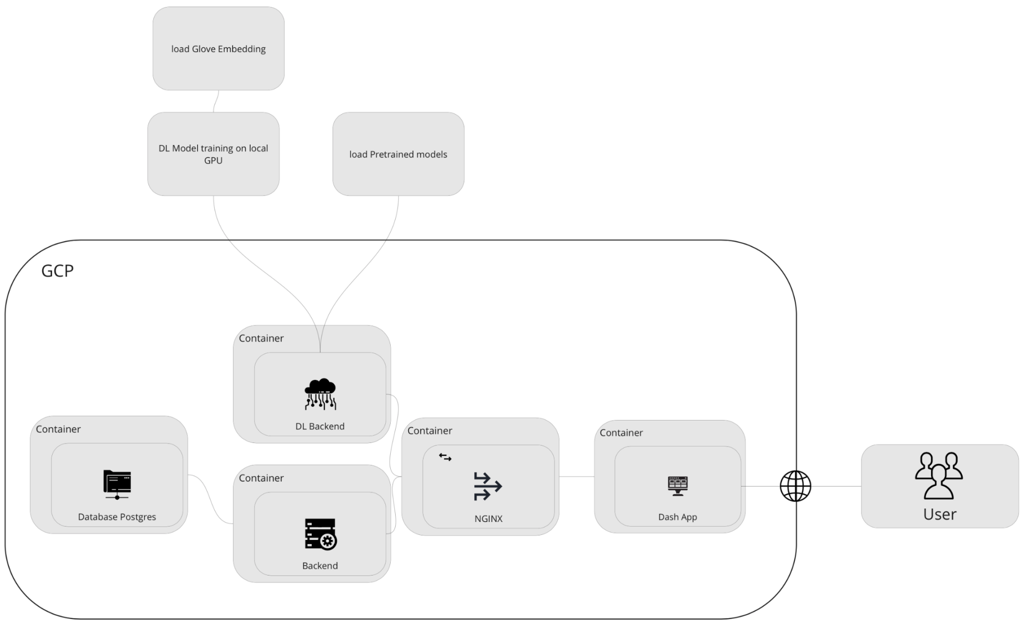


Figure 5: Overview of the System Architecture

5.1 Backend

The Backend of a technology application consists of everything that is taking place behind the scenes. From your data to calculations to services, everything falls under the umbrella of the Backend. To be specific to our application, we have two Backends one to serve the data, the other to run deep learning models. We are storing our data in a PostgreSQL[**15**] database and using NGINX[**16**] as a reverse proxy.

5.1.1 Docker

As briefly described earlier, our application has many different components. Setting everything up on different machines can be a hard and time-consuming task. This is

where **Docker**[17] comes into the picture. Docker provides an Operating System-Level virtualization in the form of isolated spaces called **containers**.
It helps in packaging our software in an isolated environment that has no dependencies on the host operating system. Making the replication of the application an easy task.
We also use **Docker-Compose**[18] which helps bundle together multiple docker containers. Managing the dependencies of each separate docker container and helps to make the application coherent.
The high level overview of the application is :

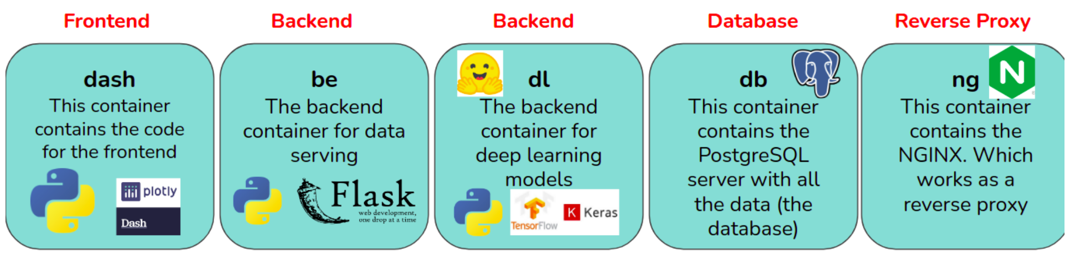

Frontend
dash
This container contains the code for the frontend
plotly
Dash
Backend
be
The backend container for data serving
Flask
web development, one drop at a time
Backend
dl
The backend container for deep learning models
TensorFlow
Keras
Database
db
This container contains the PostgreSQL server with all the data (the database)
Reverse Proxy
ng
This container contains the NGINX. Which works as a reverse proxy

Figure 6: Layout of the application

5.1.2 Flask : Backend #1

The Backend of the application is based on the Python Framework Flask[19]. Flask provides a quick and easy-to-use interface to build medium-sized applications. The purpose of this microservice is to ensure data management and serve any data that the Frontend of the application requires. This service interacts with the Dash Frontend and provides all the APIs which are required for the smooth functioning of the Frontend.

5.1.3 Deep Learning Backbone : Backend #2

The Deep Learning models could have also been implemented in the main Backend, however, there were some reasons, which we'll describe in the following chapter, because of which we decided to have a separate Backend altogether. We used Tensorflow[20] and HuggingFace[21] libraries to implement our Deep Learning models.

5.1.3.1 Why two Backends? MLOps

With the advent of machine learning, a new field has emerged in the world of software engineering. It is called the **MLOps** or (Machine Learning Operations). To make the machine learning model useable in real-time for practical applications, we need to make several optimizations. This is considered among the best practices. The models take much less time to load, as we can pre-load the model weights in our **dl** microservice (where we run our deep learning models) when serving deep learning models as a service, and we decided to go forward with this approach.
Some reasons why taking this approach was helpful in our scenario are :

1. The models take much less time to load, as we can pre-load the model weights in our **dl** microservice.

2. In case there is an error in the deep learning pipeline, it would prevent the whole app from crashing. We will still be able to access the basic data in the application.

3. As deep learning models take some time to run, it would not be a wise idea to block the Backend completely while the deep learning model is running.

5.1.4 Database

The efficient storage of data is very important. As the application scales, the amount of data also increases tremendously. To manage that, we need to have a proper schema for the data to be stored.
This is done through the help of databases. In the modern age, there are infinite options to implement data storage. Many different kinds of databases suit different kinds of applications.
For this application, we have used PostgreSQL as the database. It is among the most common forms of a relational database in the world right now. Our reason to use it is that there is a lot of support available for it, and our relational databases make sense to use in our application.

5.1.4.1 Database Architecture

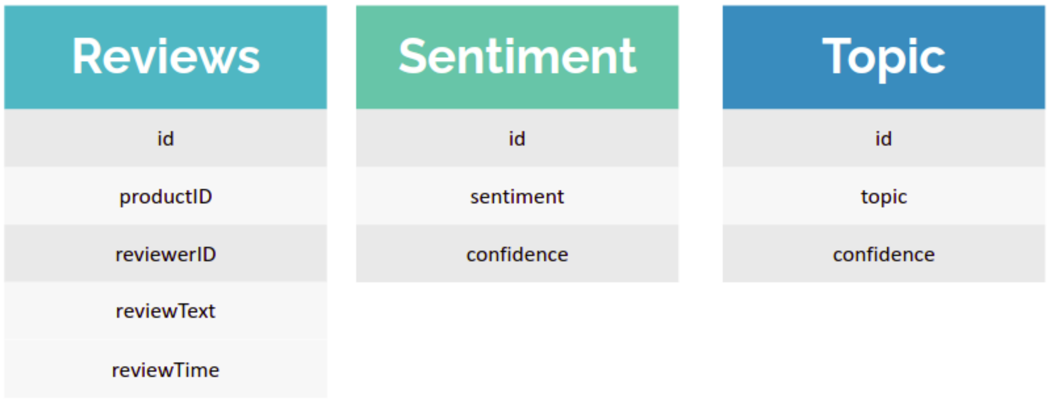


Figure 7: The Database Architecture

Our application has three different components when it comes to data storage.

1. Reviews: Data related to the review (including the review, time of review, and product ID, etc)

2. Sentiment: After the sentiment analysis runs on a review, we want to store the calculated sentiment and the confidence score.

3. Topic: After the topic modeling runs on a review, we want to store the confidences of each class

5.1.5 NGINX : Reverse Proxy

We used a reverse proxy to handle requests on the Backend server. At one time, there can be multiple users accessing the application. Multiple users interacting with the application, in turn, means that there will be a myriad of requests made on the Backend server. This can lead to failures and can potentially cause the application to crash completely.
Therefore, we use NGINX which acts as a reverse proxy or a layer between the requests made to the Backend. It redistributes the requests efficiently and ensures that the application does not crash.

5.2 Dashboard visualization

Very early on during this project, we realized that we needed to develop a dashboard application that would be the graphic interface accessed by our user. This visualization constitutes a very critical element as it engulfs all aspects of statistics and analysis results and would be the means by which the user will be able to digest all the valuable insight and actually be able to derive conclusions and eventually would warrant actions and decisions.

5.2.1 Plotly Dash

For the building of our visualization, we opted to use Dash Plotly [**5**].
Dash Plotly is an open-source Python framework for developing web-analytic applications. It is built on top of Flask.js, React.js and Plotly.js. The most notable advantages that led us to choose this particular framework are :

- It is very well suited for building customized user-interfaces in order to display analytical data
- It allows us to be able to develop our dashboard in python.
- The resulting analytic apps can easily be deployed. It is then run on the web-browser and accessed via a link. Therefore, it allows us a lot of flexibility and usability and our app can be accessed on various platforms.

5.2.1.1 Plotly library The Plotly Python library [**7**] is an open-source library containing a wide variety of plots and graphs that span many applications, whether it is statistical, 3-dimensional, specifically designed for ML and AI applications, etc ... The collection of graphs available is a great asset for the Dash environment as it is written on top of Plotly.js and therefore offers access to those. These graphs would be a great tool to use in order to build our dashboard features.

5.2.1.2 Dash Core Components It is a library [**6**] that offers a set of components that are responsible for the interactive user elements. Whenever, a user has to input a parameter that would affect the displayed data and figures, such a component from the core set has to be implemented. This could range from creating a menu that would roll down with various options, to entering a specific text in an input box (for example, an e-mail), to selecting a specific time frame, etc ... The implementation of such elements needs 2 separate elements :

- **Layout :** This is where all the different aspects pertaining to the visual designs and details need to be developed. We set-up the skeleton of the component with all the details concerning both the element itself and its placement within the screen.
  We also use one of the advantages of Dash which is that it is written on top of javascript which is one of the pillars of Frontend development.
  Furthermore, we need to implement external CSS (Cascading Style Sheets) to further customize the component.

- **App callbacks :** These are functions that need to be implemented in such a way that it takes the user input into account, triggers the live update of the figure and returns the correct updated graph requested by the user.
  In this step, to avoid storing all the data in the Dash app and using that dataset in the app callback functions, we opt for the implementation of API functions.
  These are functions that act as an intermediary agent and bridge between the dataset stored in the Backend and the visualization implemented in the Frontend. It would perform the needed computation in the Backend and only transmit the result so that we optimize the memory storage and the computation operations (i.e. time)

5.2.1.3 Bootstrap and Dash Bootstrap Components : Bootstrap [**8**] is an open-source CSS framework especially used in interactive user interfaces. It has graphic elements written on top of javascript and CSS, such as buttons and cards, that allow for a vast variety of responsive components for the user experience.
Dash Bootstrap Component (i.e. dbc) [**9**] is an open-source Bootstrap components library that is accessible via Dash.
This allows us to increase the complexity of the layout of our dashboard and maintain a high level of responsiveness.

5.3 Deployment

Finally, to deploy our application we need a cloud based service where we could serve our application. Cloud platform is nothing but just a bunch of computers lying around the world in data centers, where users can gain access to resources like compute power, storage and other services. Some of most well known cloud service providers are Amazon's AWS, Microsoft's Azure, and Google's Google Cloud Plaform.
This is the step where using docker to build our app comes in handy. Had we not done it, setting up the whole application on the cloud server could have taken a long time. However, it took very less time in our case.
We used **Google Cloud Platform (GCP)** to deploy our app.
Our application is accessible in the following link

5.4 Application Features

5.4.1 Sentiment Analysis

Sentiment analysis or opinion mining is a very popular task within NLP. It is especially important in the domain of customer feedback to better understand the polarity of the user/customer. We have tried different approaches and evaluated them to identify the best performing model. The most simple implementation of sentiment analysis is by using build in python packages. In the next stage we went on with some baseline neural networks, here we decided on training the algorithms to our use case, since review text differs strongly from normal text data (like wikipedia entries). Due to the time-series characteristic of free text, recurrent neural networks are the most commonly used models. More specifically we used LSTM (Long-Short-Term-Memory) [**23**] Networks and bidirectional LSTM 's and to also try a different kind of algorithm we made an experiment with CNN (Convolutional Neural Networks) [**24**]. In the last iteration of the sentiment analysis we trained our own hierarchical attention model, which was our best performing model on various metrics. For all our sentiment analysis we used supervised learning methods. For that purpose we had to modify our dataset slightly. The input variable will be the [reviewText] column of our dataset and the ground truth label will be the [overall]-rating. To account for the problem that one review could include positive as well as negative points, we only considered reviews with a 5-start rating as positive and reviews with a 1-star rating as negative, assuming that those reviews only contain negative or positive comments, but never both.

5.4.1.1 Python Packages An extensive desk-research in combination with the interviews, who confirmed our choice of packages, came to the result that the two best solutions in this field are VaderSentiment and Textblob. Vador stands for Valence Aware Dictionary and Sentiment Reasoner and 'is a lexicon and rule based sentiment analysis tool, that is specifically attuned to sentiments expressed in social media' [**11**]. TextBlob [**12**] in return is a Python Library for various tasks around text-processing including sentiment analysis. We made experiments in that direction with out-of-the-box solutions, which had the advantage of not needing computing ressources for training .
To evaluate the performance of the two python packages we ran the according package on our dataset and specified some details. For instance the output of VaderSentiment and TextBlob is a polarity score between -1 and 1. To transfer it to a binary count we had to round it to exactly -1 and 1. For the neutral number of 0 we did experiments with classifying them as -1 or 1 and we achieved better results if we classified the neutral reviews as positive (this can also be due to class imbalance). The final results were 85.14% Accuracy for the Vader package and 82.34% for the TextBlob.

5.4.1.2 Baseline Neural Networks Today Deep Learning as a subfield of Machine Learning achieve outstanding results in various tasks in computer vision, speech recognition and natural language processing. To use text data as input for Deep Learning, one has to transform the words into vectors. This is done by a techniques called *word embeddings*, which allows word that have a similar meaning to have a similar representation in vector space. There exists various types of neural networks, but since text data is a form of sequential data, the suitable type is a recurrent neural network (RNN) as it is able to store information from previous steps and therefore accounts for "memorizing" earlier words. To set a baseline we also tried another type of neural network, namely Convolutional Neural Networks (CNN's), which are primarily used in computer vision and image processing.
When working with longer sentences the memorizing ability of classic RNN's is not strong enough, because vanilla RNN's suffer from the vanishing gradient problem (loosing in-

formation of earlier timesteps) [**25**] . To account for this problem the more advanced Long-Short-Term-Memory-Networks (LSTM's) are used. LSTM's are a variation of RNN but are able to remember information for a longer time period due to its gating structure. LSTM's are based on different gates that determine which information from previous time steps will be passed to the next cell or be forgotten. For instance they have a memory cell that can save information over many time steps.
One drawback of classic LSTM's is that it processes sequential data unidirectionally. Therefore the network can only remember the words that appeared before in the text but it cannot make use of information at the end of the sentence. For a better textual understanding and to also take the future into account bidirectional LSTM's show very good results in a lot of NLP tasks, as they can capture the meaning of the text better. The biLSTM is able to take information from the whole text as it processes the sequential text once from beginning to end, but also backwards from end to beginning.
To evaluate our baseline neural networks we first performed a train/test split of our data. For the size of our test set we used the sklearn default value of 0.25 which gave us 401526 training samples and 133842 test samples. The LSTM models consist of three parts, first an Embedding layer, then the actual LSTM layer with integrated dropout to prevent overfitting and then an activation function.
For both the classic LSTM layer, as well as the bidirectional LSTM layer we used rmsprop as an optimizer and a dropout-value of 0.5. For the CNN in turn we also started with an Embedding layer but then turned to a Conv1D layer with Relu activation followed by a maxpooling operation. Then another Conv1D layer (+Relu) with following GlobalMaxpooling1D. Lastly a dense layer was added.
To evaluate the actual performance we plotted the three confusion matrices that can be found in the Appendix.
As can be seen there, the unidirectional LSTM performed slightly better on negative reviews, whereas the other two perform slightly better on positive reviews.
As an overall metric we decided to use the F1 score, as it takes both the precision and recall into account. In this metric the bidirectionalLSTM slightly outperforms the other network but are very close to each other with a score of 0.967, compared to 0.964 for the LSTM and 0.942 for the CNN.
Another measure to take into account is the computation time. In this metric the CNN clearly outperforms the other two with 175s 437us/step. The bidirectional LSTM takes the longest time for training on local CPU with 1957s 5ms/step (single LSTM takes 1509s 4ms/step).

5.4.1.3 Hierarchical Attention Network A HAN neural architecture [**26**] is designed to account for the hierarchical nature of text documents and the fact that word and sentence significance varies according to the contexts in which they appear. As such, a document representation is an aggregation of the representations of the sentences of which it is comprised, just as a sentence representation is an aggregation of the representations of the words in said sentence. A hierarchical attention architecture also enables the model to adjust the relative importance it gives to each word or sentence based on context. The models architecture is presented in Figure 11. The HAN architecture consists of a word sequence encoder, a word-level attention layer, a sentence encoder and a sentence-level attention layer. We used GloVe embeddings [**22**] as network inputs for

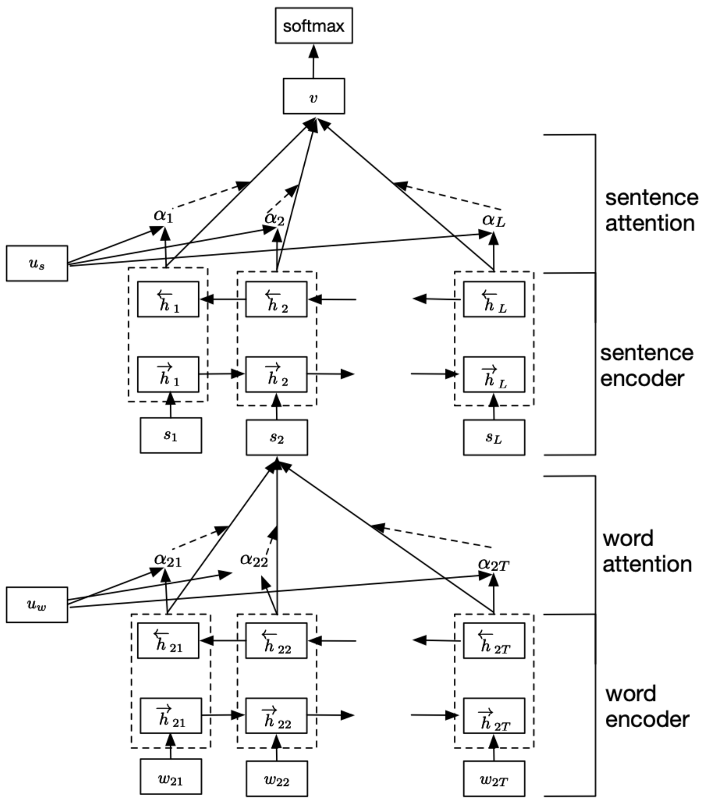


Figure 8: Model Architecture of Hierarchical Attention Network

word representation. Since HAN is designed to handle the hierarchical nature of user reviews and is more robust when dealing with heavily unbalanced data, it outperformed other network architectures that we experimented with.
In Figure 14 we present the results of our state-of the art model - The Hierarchical Attention Model (HAN). For this model evaluation we printed out some more metrics, but still agree that F1 score is the most important metrics to compare it to the other models. The HAN model clearly outperforms the baseline model in all metrics (except training time). Very important for us was especially the performance on negative reviews, since the dataset was imbalanced to positive reviews, but the priority from our customers was on the negative reviews. Apart from the classical HAN implementation, we also achieved some improvements while iterating and improving the algorithm. For instance in an earlier iteration we recognized problems, that the algorithm makes decision based on the length of the review not matter the actual sentiment. This problem could be solved and the algorithm shows stable performance.
Below there can be seen an overview of all the F1 scores which yields us to the decision that the Hierarchical Attention Model is the best performing one and therefore the suitable solution for out sentiment analysis.

	precision	recall	F1-score
negative	0.942	0.935	0.938
positive	0.985	0.987	0.986
accuracy	0.977	0.977	0.977
macro avg	0.963	0.961	0.962
weighted avg	0.977	0.977	0.977

Table 1: results of Hierarchical Attention Network

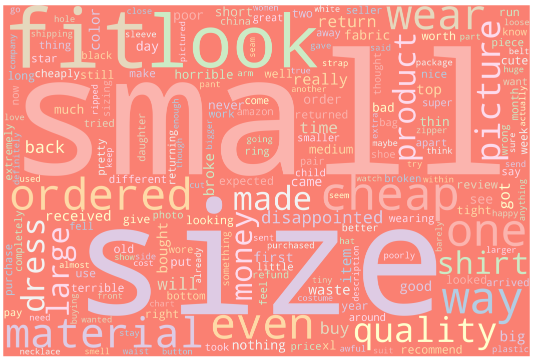
	CNN	LSTM	biLSTM	HAN
F1-score	0.942	0.964	0.967	0.977

Table 2: Comparison of F1 scores between the Neural Networks

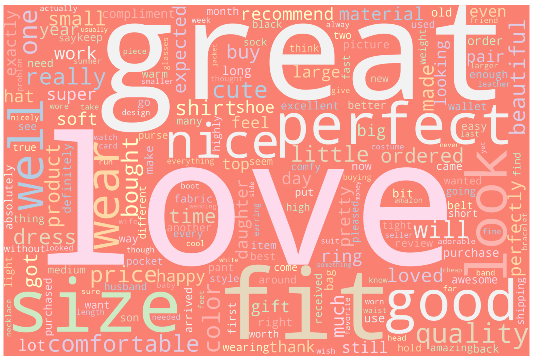
5.4.2 Topic Modeling

Apart from Sentiment Analysis, Topic Modeling is a very common task within the field of Natural Language Processing. It was also the main finding from our interviews, that topic modeling is very hard to perform automatically and time-intensive when performed manually. The main hurdle in this tasks lies in the fact, that topic modeling is compared to sentiment analysis an unsupervised learning task, as we do not have labelled training data. For this purpose we have experimented with three different algorithms in the following sections. Before starting the algorithms we printed word-clouds to get a feeling for the content of the positive and negative reviews. One of the shortcomings of using unsupervised models for text classification is the lack of an accuracy metrics that can be used to measure and benchmark the performance of the various models. Nevertheless, well-managed manual testing of the respective models has been a practical alternative to accuracy metrics that would otherwise be available in the case of supervised learning models. Analysing the results of the topic modeling cannot be as easily quantified with metrics as in the sentiment analysis since it is an unsupervised tasks without ground truth labels. Therefore the evaluation is more qualitative.
The first insights we gained from the topic modeling was by printing the wordclouds for the positive and negative category as can the seen in the Figure 10. To make them useful we first removed the stopwords.

5.4.2.1 LDA Latent Dirichlet Allocation [**27**] is an unsupervised learning technique, that tries to discover the hidden themes in a collection. It takes as input the data and the number of classes it should categorize the data into. It only takes the bag-of-word into account with some pre-processing like lemmatization to improve the results. Extracted topics are sets of words that represent topics discovered in the document. In the representation of documents, latent topics are characterized by distributions over words within each topic. The same term can be present in several topic sets at once, but with different probabilities representing the degree to which the word fits in each set. Topic probabilities provide a way to explicitly represent a document.



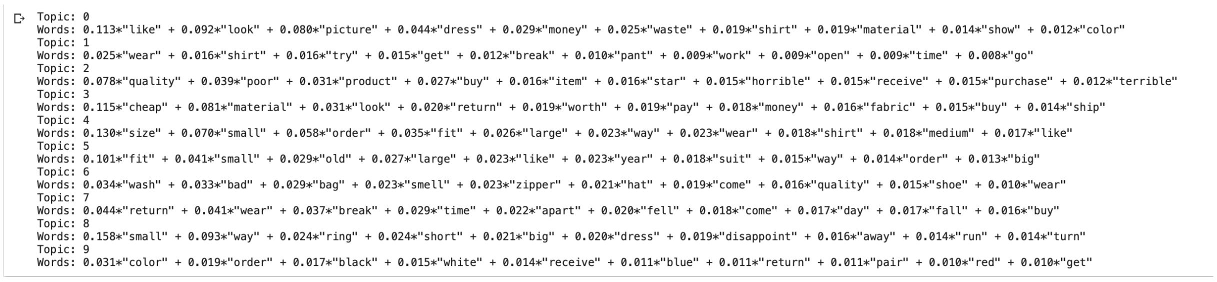
(a) negative



(b) positive

Figure 9: Wordclouds

For our first approach, the Latent Dirichlet Allocation we achieved the following results: Figure 10 shows the categories as outlined by the LDA with the best describing words for that category. One can see clear tendencies in the categories for instance towards quality in topic 2 or towards size in topic 4. Drawbacks of this solution include that the characteristics are not exclusive, therefore it is sometimes hard to differentiate the categories. Further one has to predefine the number of classes the LDA is supposed to look out for, therefore a certain knowledge about the content is prerequisite.

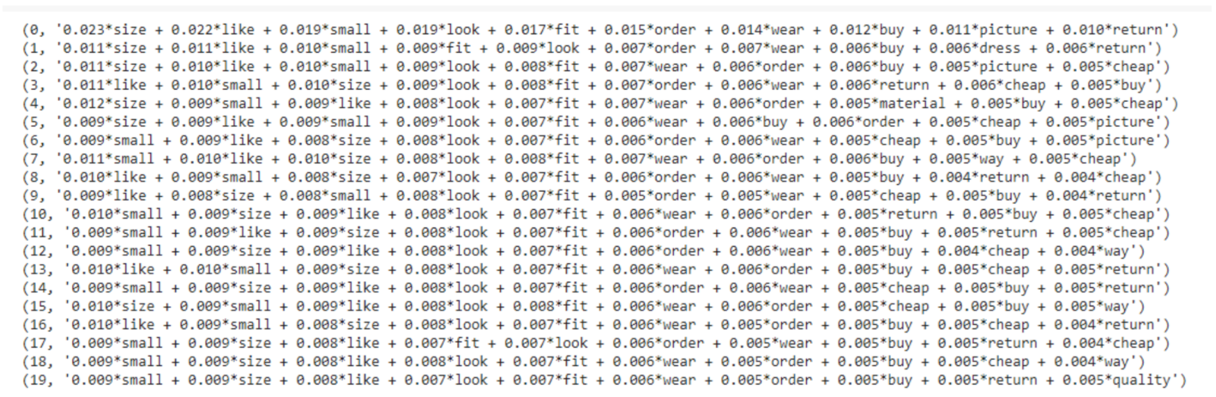


```
Topic: 0
Words: 0.113*"like" + 0.092*"look" + 0.080*"picture" + 0.044*"dress" + 0.029*"money" + 0.025*"waste" + 0.019*"shirt" + 0.019*"material" + 0.014*"show" + 0.012*"color"
Topic: 1
Words: 0.025*"wear" + 0.016*"shirt" + 0.016*"try" + 0.015*"get" + 0.012*"break" + 0.010*"pant" + 0.009*"work" + 0.009*"open" + 0.009*"time" + 0.008*"go"
Topic: 2
Words: 0.078*"quality" + 0.039*"poor" + 0.031*"product" + 0.027*"buy" + 0.016*"item" + 0.016*"star" + 0.015*"horrible" + 0.015*"receive" + 0.015*"purchase" + 0.012*"terrible"
Topic: 3
Words: 0.115*"cheap" + 0.081*"material" + 0.031*"look" + 0.020*"return" + 0.019*"worth" + 0.019*"pay" + 0.018*"money" + 0.016*"fabric" + 0.015*"buy" + 0.014*"ship"
Topic: 4
Words: 0.130*"size" + 0.070*"small" + 0.058*"order" + 0.035*"fit" + 0.026*"large" + 0.023*"way" + 0.023*"wear" + 0.018*"shirt" + 0.018*"medium" + 0.017*"like"
Topic: 5
Words: 0.101*"fit" + 0.041*"small" + 0.029*"old" + 0.027*"large" + 0.023*"like" + 0.023*"year" + 0.018*"suit" + 0.015*"way" + 0.014*"order" + 0.013*"big"
Topic: 6
Words: 0.034*"wash" + 0.033*"bad" + 0.029*"bag" + 0.023*"smell" + 0.023*"zipper" + 0.021*"hat" + 0.019*"come" + 0.016*"quality" + 0.015*"shoe" + 0.010*"wear"
Topic: 7
Words: 0.044*"return" + 0.041*"wear" + 0.037*"break" + 0.029*"time" + 0.022*"apart" + 0.020*"fell" + 0.018*"come" + 0.017*"day" + 0.017*"fall" + 0.016*"buy"
Topic: 8
Words: 0.158*"small" + 0.093*"way" + 0.024*"ring" + 0.024*"short" + 0.021*"big" + 0.020*"dress" + 0.019*"disappoint" + 0.016*"away" + 0.014*"run" + 0.014*"turn"
Topic: 9
Words: 0.031*"color" + 0.019*"order" + 0.017*"black" + 0.015*"white" + 0.014*"receive" + 0.011*"blue" + 0.011*"return" + 0.011*"pair" + 0.010*"red" + 0.010*"get"
```

Figure 10: LDA results

5.4.2.2 HDP

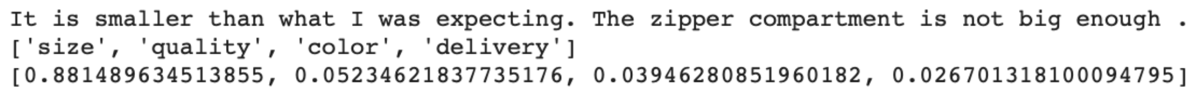
Hierarchical Dirichlet Process (HDP) [28] is a nonparametric Bayesian approach to clustering. It acts as an extension to LDA with the main advantage of not having to predefine the number of topics. This holds the advantage that no prior knowledge about possible classes must be available and the maximum number of topics is unbound. Like LDA, HDP models topics as mixtures of words. However, instead of documents being composed of a set of topics, the number of topics is decided by a Dirichlet process. Since the number of topics is a random variable as well, HDP is often unstable and generates topics that are not inline with the human interpretation of the documents in question. This is a shortcoming that is typical of most non-parametric unsupervised clustering models.
The results of the HDP algorithm are shown in the figure 11. One can see that despite our high expectations the results are less meaningful compared to the LDA. Therefore the advantage of not having to outline the number of topics in advance, looses its benefits, because there is no clear inter-class differentiation.



```
(0, '0.023*size + 0.022*like + 0.019*small + 0.019*look + 0.017*fit + 0.015*order + 0.014*wear + 0.012*buy + 0.011*picture + 0.010*return')
(1, '0.011*size + 0.011*like + 0.010*small + 0.009*fit + 0.009*look + 0.007*order + 0.007*wear + 0.006*buy + 0.006*dress + 0.006*return')
(2, '0.011*size + 0.010*like + 0.010*small + 0.009*look + 0.008*fit + 0.007*wear + 0.006*order + 0.006*buy + 0.005*picture + 0.005*cheap')
(3, '0.011*like + 0.010*small + 0.010*size + 0.009*look + 0.008*fit + 0.007*order + 0.006*wear + 0.006*return + 0.006*cheap + 0.005*buy')
(4, '0.012*size + 0.009*small + 0.009*like + 0.008*look + 0.007*fit + 0.007*wear + 0.006*order + 0.005*material + 0.005*buy + 0.005*cheap')
(5, '0.009*size + 0.009*like + 0.009*small + 0.009*look + 0.007*fit + 0.006*wear + 0.006*buy + 0.006*order + 0.005*cheap + 0.005*picture')
(6, '0.009*small + 0.009*like + 0.008*size + 0.008*look + 0.007*fit + 0.006*order + 0.006*wear + 0.005*cheap + 0.005*buy + 0.005*picture')
(7, '0.011*small + 0.010*like + 0.010*size + 0.008*look + 0.008*fit + 0.007*wear + 0.006*order + 0.006*buy + 0.005*way + 0.005*cheap')
(8, '0.010*like + 0.009*small + 0.008*size + 0.007*look + 0.007*fit + 0.006*order + 0.006*wear + 0.005*buy + 0.004*return + 0.004*cheap')
(9, '0.009*like + 0.008*size + 0.008*small + 0.008*look + 0.007*fit + 0.005*order + 0.005*wear + 0.005*cheap + 0.005*buy + 0.004*return')
(10, '0.010*small + 0.009*size + 0.009*like + 0.008*look + 0.007*fit + 0.006*wear + 0.006*order + 0.005*return + 0.005*buy + 0.005*cheap')
(11, '0.009*small + 0.009*like + 0.009*size + 0.008*look + 0.007*fit + 0.006*order + 0.006*wear + 0.005*buy + 0.005*return + 0.005*cheap')
(12, '0.009*small + 0.009*size + 0.009*like + 0.008*look + 0.007*fit + 0.006*order + 0.006*wear + 0.005*buy + 0.004*cheap + 0.004*way')
(13, '0.010*like + 0.010*small + 0.009*size + 0.008*look + 0.007*fit + 0.006*wear + 0.006*order + 0.005*buy + 0.005*cheap + 0.005*return')
(14, '0.009*small + 0.009*size + 0.009*like + 0.008*look + 0.007*fit + 0.006*order + 0.006*wear + 0.005*cheap + 0.005*buy + 0.005*return')
(15, '0.010*size + 0.009*small + 0.009*like + 0.008*look + 0.008*fit + 0.006*wear + 0.006*order + 0.005*cheap + 0.005*buy + 0.005*way')
(16, '0.010*like + 0.009*small + 0.008*size + 0.008*look + 0.007*fit + 0.006*wear + 0.005*order + 0.005*buy + 0.005*cheap + 0.004*return')
(17, '0.009*small + 0.009*size + 0.008*like + 0.007*fit + 0.007*look + 0.006*order + 0.005*wear + 0.005*buy + 0.005*return + 0.004*cheap')
(18, '0.009*small + 0.009*size + 0.008*like + 0.008*look + 0.007*fit + 0.006*wear + 0.005*order + 0.005*buy + 0.005*cheap + 0.004*way')
(19, '0.009*small + 0.009*size + 0.008*like + 0.007*look + 0.007*fit + 0.006*wear + 0.005*order + 0.005*buy + 0.005*return + 0.005*quality')
```

Figure 11: HDP results

5.4.2.3 Zero - Shot learning Having tried the most used solution with LDA and an improved version in the HDP, our research guided us in another promising field - Zero-shot-learning. This tendency was then also confirmed by an interview with an experienced team, that has also worked on a similar problem in a large company before. Zero-shot learning is about getting the model to do text classification without explicit training. The difference to other unsupervised techniques is that instead of just inputting the data, in zero shot learning the algorithm gets as input the data and the labels. Those candidate labels are then transferred to a semantic embedding space in the same dimension as the text input to later identify similarities. To understand the meaning of the labels, the textual descriptions of the word are taken into account, for instance by augmenting the single word with a wikipedia description or other definition about it. When running the algorithm, there is the option to choose between a single label classification or a multi-label classification. In our case we started of with the single label classification to be able to match each review with exactly one topic, but a multi-label classification is definitely to be considered for future iterations.
In Figure 18 the result of an example review with zero shot learning are shown. In this case the predefined candidate labels were: "size", "quality", "color", "delivery" and below that there can be seen the probabilities that the review belongs to the topic. Clearly in this



```
It is smaller than what I was expecting. The zipper compartment is not big enough .
['size', 'quality', 'color', 'delivery']
[0.881489634513855, 0.05234621837735176, 0.03946280851960182, 0.026701318100094795]
```

Figure 12: Example of a zero shot learning output

6 Results

6.1 Dashboard Application

In the context of our project, we developed a Dashboard web-app that displays all the analysis performed on the review data. It constitutes the visualization interface that the user interacts with.

6.1.0.1 Functionalities

We embedded our app with certain functionalities that we believe contribute to creating a fluid and comprehensive user experience. These functionalities are as follows :

- **Authentication Process :** The data we are dealing with is very sensitive and confidential.
  One of the advantages of Dash is that it is run on web browsers. However, the downside of it is that if the link would be leaked or inadvertently sent to the wrong person, this individual now has unlimited access to the data and the analysis resulting from it.
  We therefore implemented an authentication process that would require a username and password to login and gate-keep the valuable data from third parties. The user would then automatically logout when exiting the web browser.

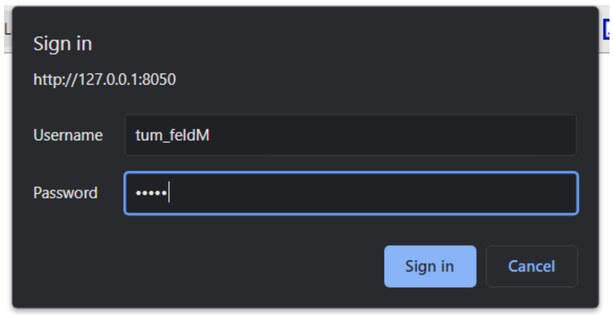
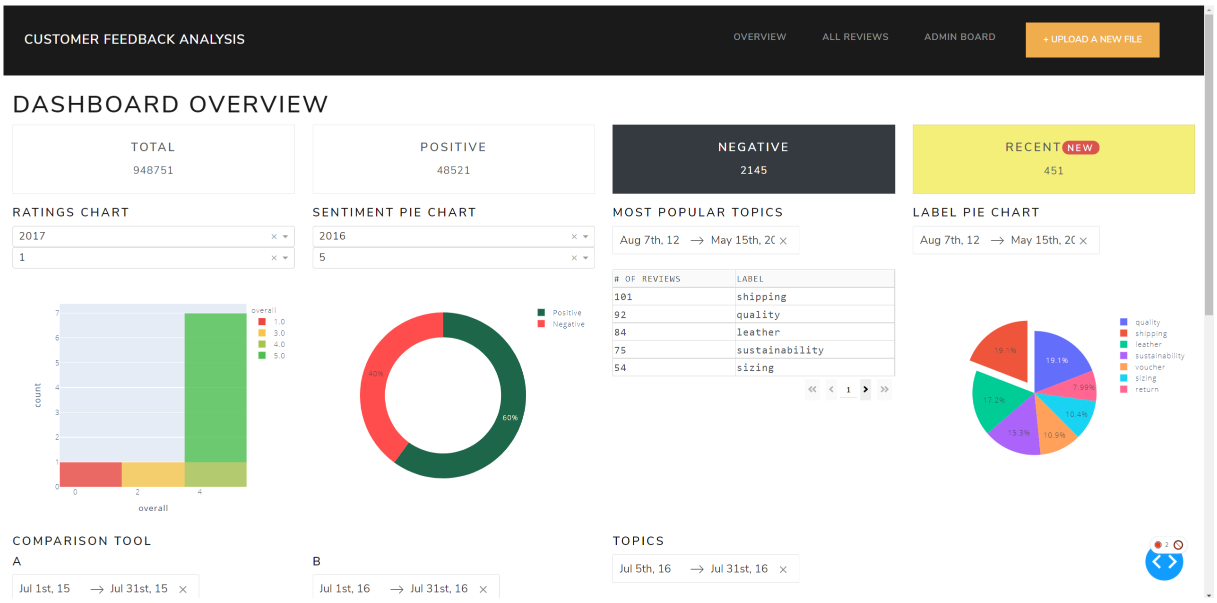

Sign in
http://127.0.0.1:8050
Username tum_feldM
Password •••••

Figure 13: Login authentication process

- **Multi-page app :** Forcing all the different displays on the same screen would overcrowd our dashboard app and actually render it very difficult and confusing to use. We therefore embedded our Dash app into a multi-page architecture. We are offering three different screens that cater to different roles :
  1. **Overview :** This is the landing page of our Dash app. It encompasses statistical graphs that give an overview of the current dataset, allows to pinpoint the timeframes of interest and get the corresponding graphs and charts.
  2. **All reviews :** This page allows an overview of all the reviews within certain filters that the user inputs. So that the user could be able to only extract the reviews of interest and get that insight without it being drowned in the other reviews that could be skewing results.
  3. **Admin Board :** In here, there is valuable information stored that pertains to the user/company and their specific data (in regards to the list of topics).
- **Upload a new file :** New reviews come in daily/weekly/... so we need to be able to process new incoming reviews as needed.
  We therefore offer this functionality where the user is able to upload a new csv that


CUSTOMER FEEDBACK ANALYSIS

OVERVIEW ALL REVIEWS ADMIN BOARD + UPLOAD A NEW FILE

DASHBOARD OVERVIEW

TOTAL	POSITIVE	NEGATIVE	RECENT NEW
948751	48521	2145	451

RATINGS CHART: 2017, 1

SENTIMENT PIE CHART: 2016, 5

MOST POPULAR TOPICS: Aug 7th, 12 → May 15th, 20

# OF REVIEWS	LABEL
101	shipping
92	quality
84	leather
75	sustainability
54	sizing

LABEL PIE CHART: Aug 7th, 12 → May 15th, 20

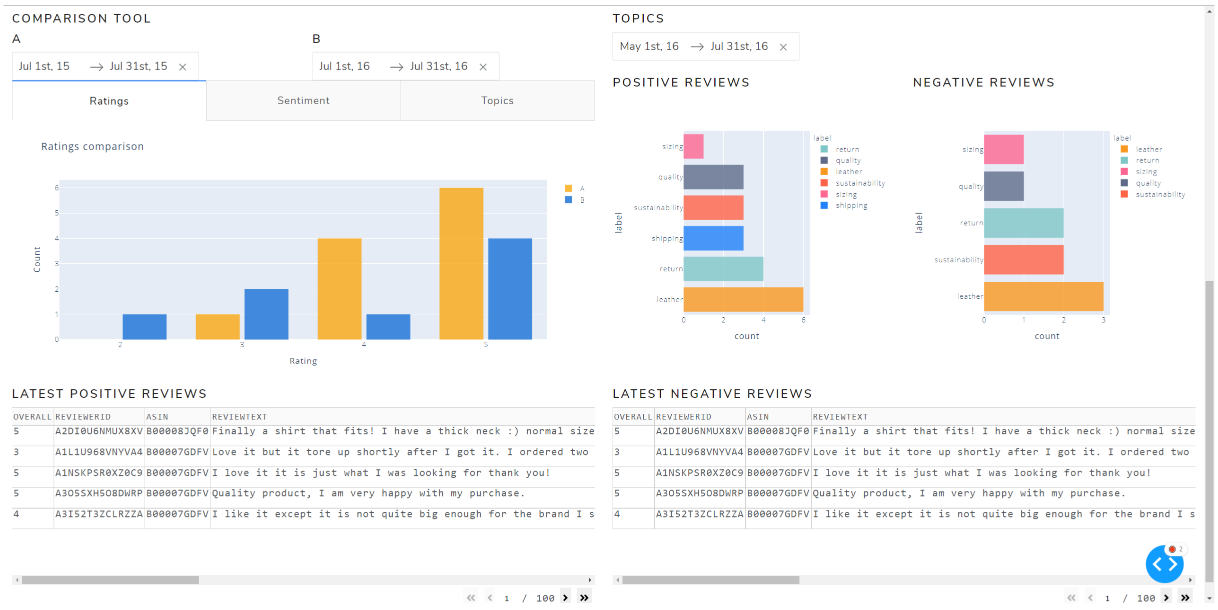
COMPARISON TOOL

A: Jul 1st, 15 → Jul 31st, 15

B: Jul 1st, 16 → Jul 31st, 16

TOPICS: Jul 5th, 16 → Jul 31st, 16

Figure 14: Screenshot of the landing page "Overview" - part 1


COMPARISON TOOL

A: Jul 1st, 15 → Jul 31st, 15

B: Jul 1st, 16 → Jul 31st, 16

TOPICS: May 1st, 16 → Jul 31st, 16

POSITIVE REVIEWS

NEGATIVE REVIEWS

LATEST POSITIVE REVIEWS

OVERALL	REVIEWERID	ASIN	REVIEWTEXT
5	A2DI0U6NMUX8XV	B00008JQF0	Finally a shirt that fits! I have a thick neck :) normal size
3	A1L1U968VNYVA4	B00007GDFV	Love it but it tore up shortly after I got it. I ordered two
5	A1NSKPSR0XZ0C9	B00007GDFV	I love it it is just what I was looking for thank you!
5	A3O5SXH5O8DWRP	B00007GDFV	Quality product, I am very happy with my purchase.
4	A3I52T3ZCLRZZA	B00007GDFV	I like it except it is not quite big enough for the brand I s

1 / 100

LATEST NEGATIVE REVIEWS

OVERALL	REVIEWERID	ASIN	REVIEWTEXT
5	A2DI0U6NMUX8XV	B00008JQF0	Finally a shirt that fits! I have a thick neck :) normal size
3	A1L1U968VNYVA4	B00007GDFV	Love it but it tore up shortly after I got it. I ordered two
5	A1NSKPSR0XZ0C9	B00007GDFV	I love it it is just what I was looking for thank you!
5	A3O5SXH5O8DWRP	B00007GDFV	Quality product, I am very happy with my purchase.
4	A3I52T3ZCLRZZA	B00007GDFV	I like it except it is not quite big enough for the brand I s

1 / 100

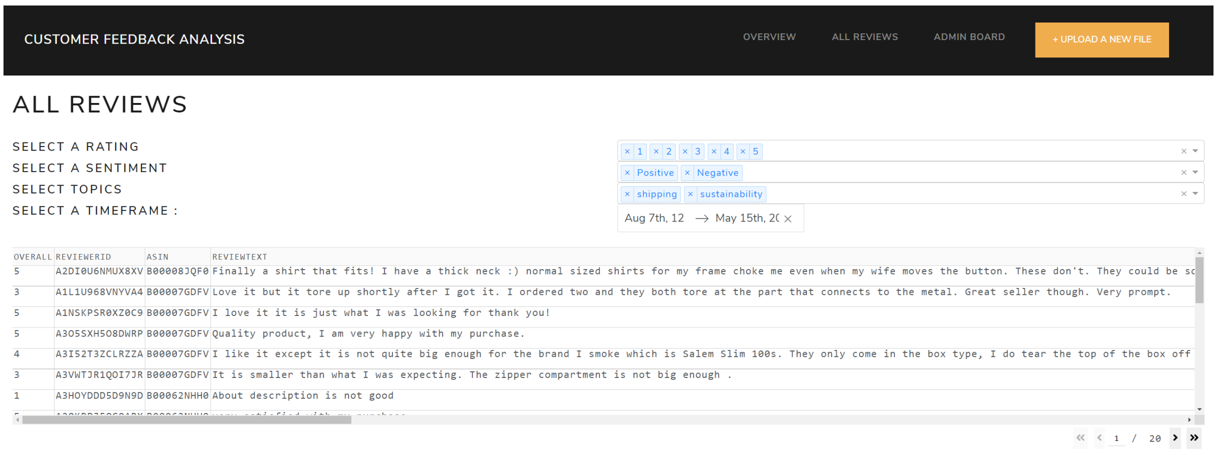
Figure 15: Screenshot of the landing page "Overview" - part 2

would be added to his pre-existing dataset. This operation would trigger the process of these new reviews via the sentiment analysis models and topic extracting models and render the updated results.

We also populated our visualization architecture with graphs and charts designed to convey the insight derived from the analysis in the most thorough fashion.

6.1.0.2 Implemented features

As a result from the define step pipeline, we established a list of top-priority features that our interview partners would be seeking in such a product. We need now to implement



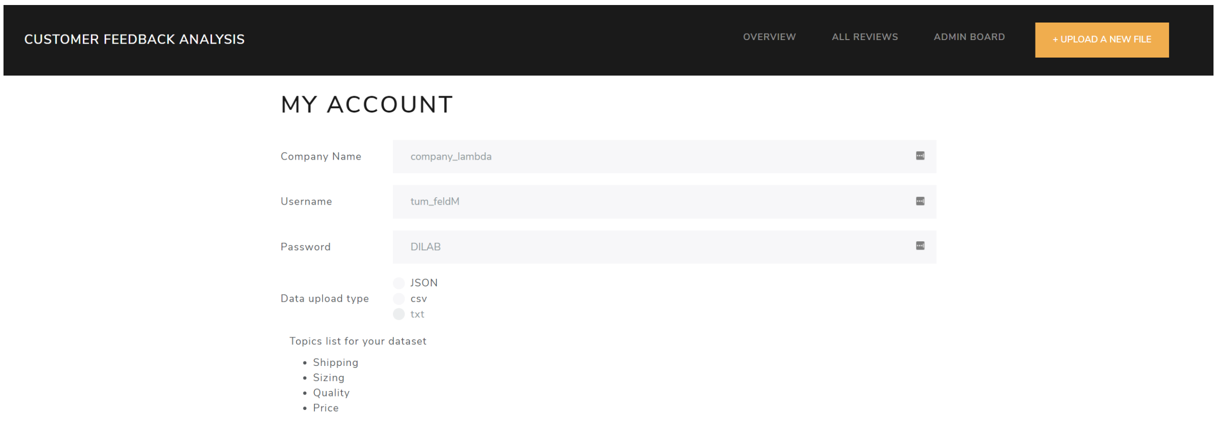
ALL REVIEWS

SELECT A RATING	× 1 × 2 × 3 × 4 × 5
SELECT A SENTIMENT	× Positive × Negative
SELECT TOPICS	× shipping × sustainability
SELECT A TIMEFRAME :	Aug 7th, 12 → May 15th, 2(

OVERALL	REVIEWERID	ASIN	REVIEWTEXT
5	A2DI0U6NMUX8XV	B00008JQF0	Finally a shirt that fits! I have a thick neck :) normal sized shirts for my frame choke me even when my wife moves the button. These don't. They could be sc
3	A1L1U968VNYVA4	B00007GDFV	Love it but it tore up shortly after I got it. I ordered two and they both tore at the part that connects to the metal. Great seller though. Very prompt.
5	A1NSKPSR0XZ0C9	B00007GDFV	I love it it is just what I was looking for thank you!
5	A3O5SXH5O8DWRP	B00007GDFV	Quality product, I am very happy with my purchase.
4	A3I52T3ZCLRZZA	B00007GDFV	I like it except it is not quite big enough for the brand I smoke which is Salem Slim 100s. They only come in the box type, I do tear the top of the box off
3	A3VWTJR1QOI7JR	B00007GDFV	It is smaller than what I was expecting. The zipper compartment is not big enough .
1	A3HOYDDD5D9N9D	B00062NHH0	About description is not good

1 / 20

Figure 16: Screenshot of the "All reviews" page



MY ACCOUNT

Company Name	company_lambda
Username	tum_feldM
Password	DILAB
Data upload type	JSON csv txt

Topics list for your dataset

- Shipping
- Sizing
- Quality
- Price

Figure 17: Screenshot of the Admin board

it in our dashboard. This was done by the use of various graphs and components that we found best conveyed the insight drawn from the data.

- **Comparison tool :** This tool is implemented as a mean to compare the data between two different timeframes. This features was a very sought-after attribute that was discussed with our interview partners.
  This tool is actually used to detect if there is a surge in a certain topic that they need to watch out for and seek the resolution of the issue before it extends in time, or an increase in positive/negative reviews and investigate the reason behind that. This is also useful to track if there is a positive feedback from customers due to changes implemented in services and such.
  The user is therefore able to choose the two different timeframes that he/she wishes to compare. Once the dates are selected, the three graphs regarding the score, the sentiments, and the topics are updated automatically. It is then possible to toggle between the different tabs and try for different dates as needed.

- **Topic analysis :** This tool seeks out to give more insight using the topic extracting data. This allows to see the most popular topics within the positive reviews and

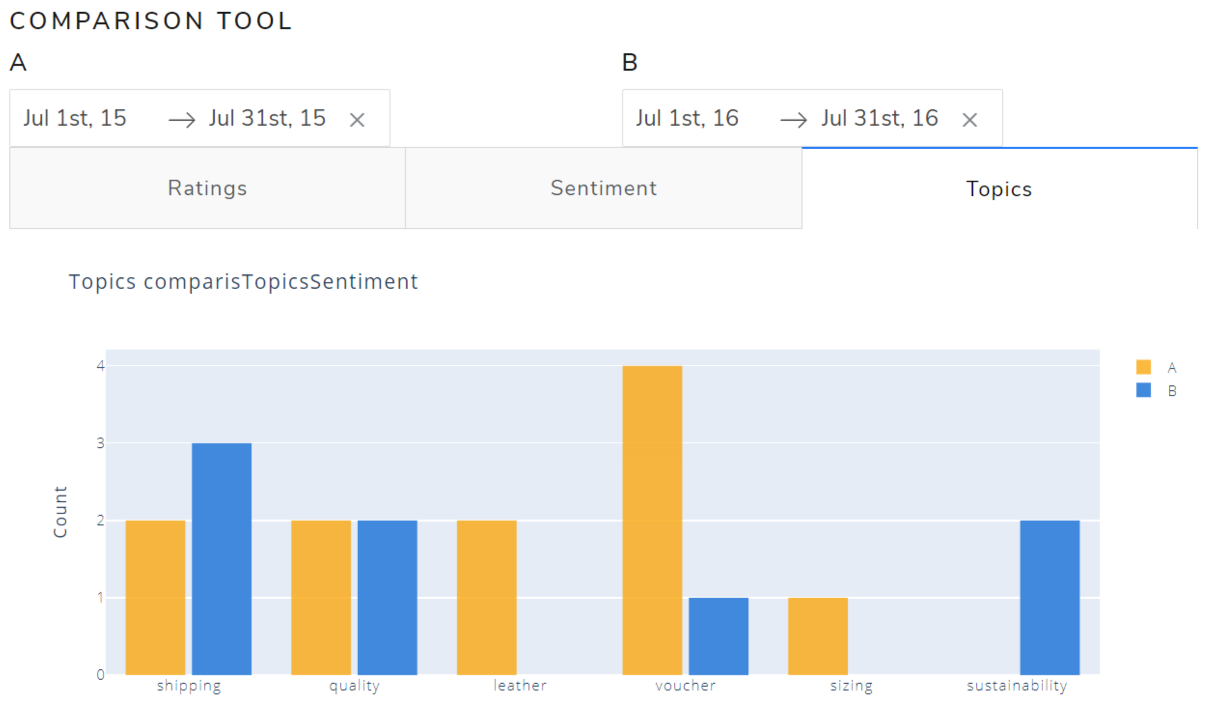


Figure 18: Comparison tool that displays graphs comparing the data from the months of July of 2015 and 2016

the negative reviews separately, within a desired timeframe.
So it's possible to discern between topics that seem to be successful among the customers and the topics that are gaining popularity for the wrong reasons and that need to be addressed by the company to improve their services and better cater to their customers. This offers a better insight than to lump everything together and not be able to act on the data.

- **KPI cards :** We implemented 4 KPI (Key Performance Indicator) cards that hold important numbers that the user would monitor such as total number of reviews, divided into both positive and negative ones and the number of recently uploaded reviews.

- **Summary graphs :** We also implemented features that would offer a quick summary of the state of the feedback (in accord with what our interview partners looked for). We therefore offer a barplot of the review ratings, this also can be used as a visual monitor of the NPS (Net Promoter Score) which is a metric that describes how likely would a customer recommend the company to others.
We also offer a pie chart of the positive/negative reviews within a specified timeframe. This offers an idea about the customer response at a first glance.
For the topics, we display a table of the most popular topics (again, within a specified timeframe) but also a pie chart about the distribution of these topics (by percentage).

- **Glimpse of the latest reviews :** Another sought after feature would be to have

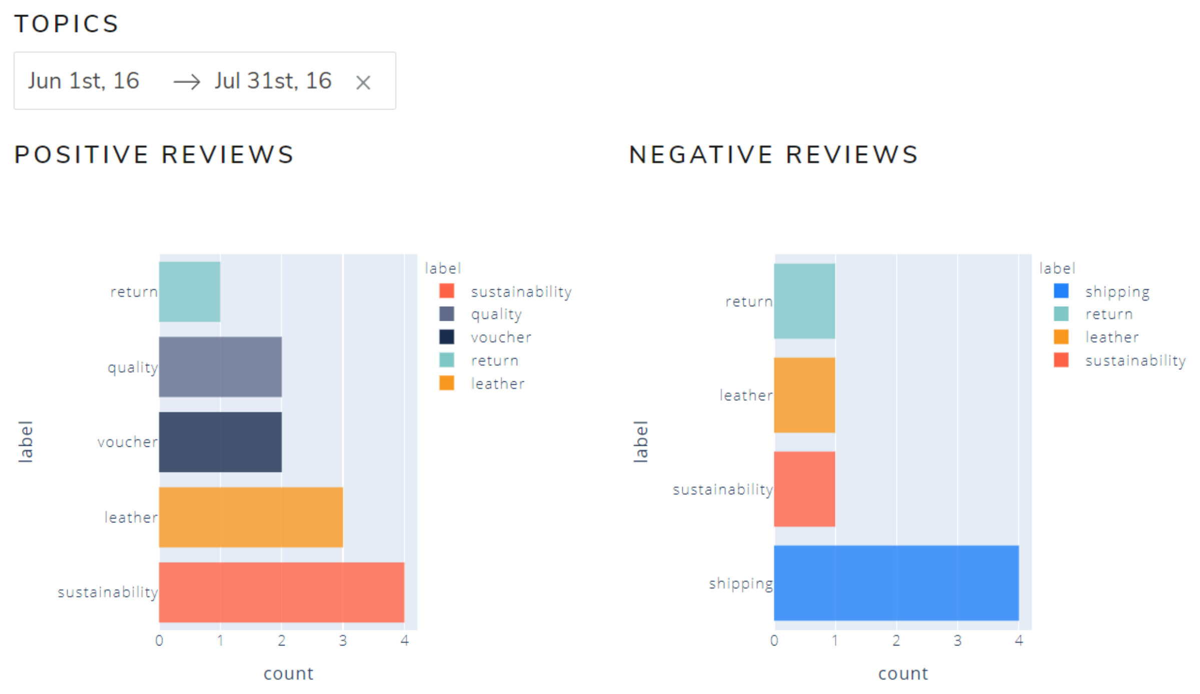
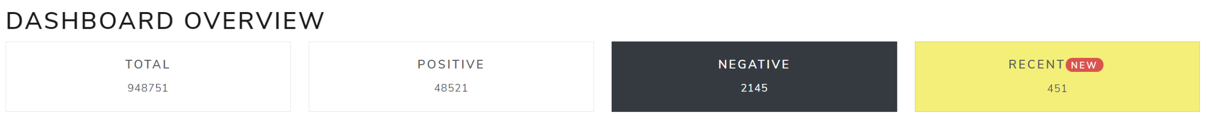


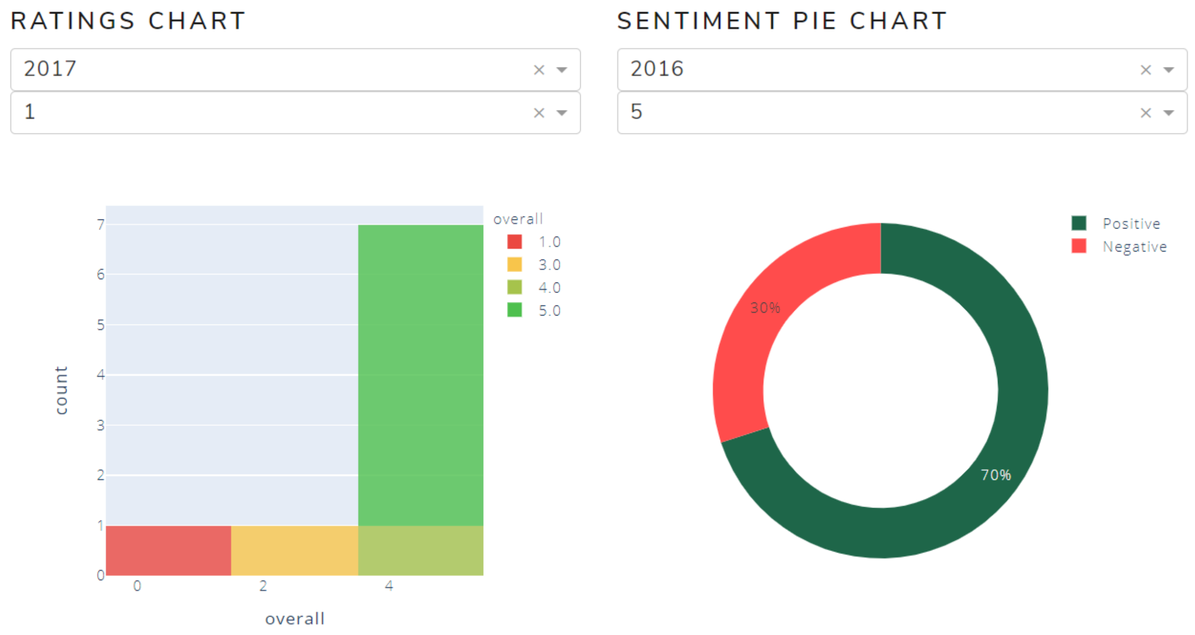
Figure 19: Topic analysis tool that discerns between positive and negative reviews in the months of June July of 2016


DASHBOARD OVERVIEW

TOTAL	POSITIVE	NEGATIVE	RECENT NEW
948751	48521	2145	451

Figure 20: Four KPI cards displayed in the dashboard landing page

a very quick overview of the latest reviews and have a quick feeling about what is said both for positive reviews and for negative reviews.


RATINGS CHART

2017

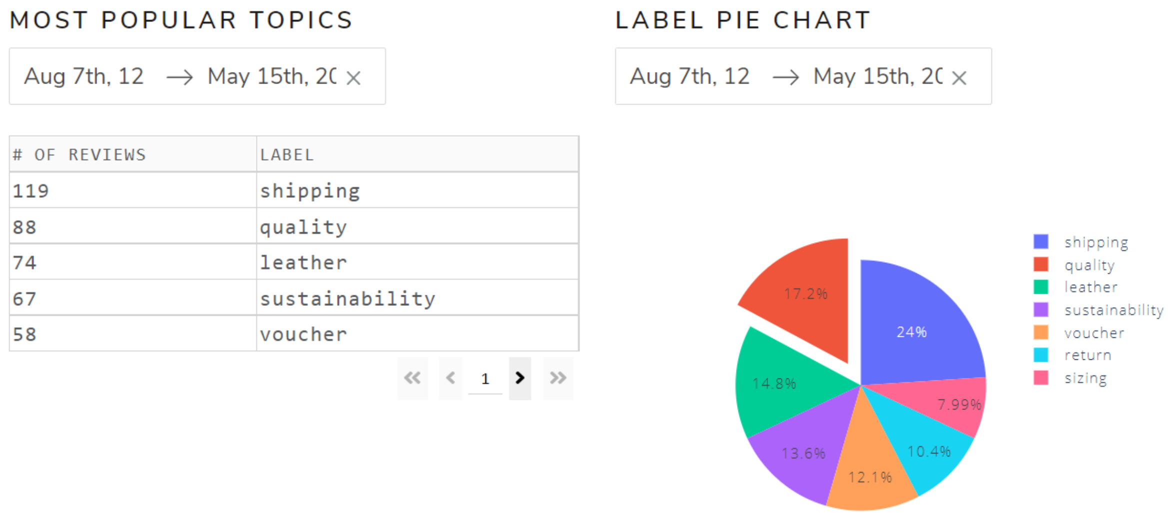
1

SENTIMENT PIE CHART

2016

5

Figure 21: Review ratings barplot chart & Review sentiment pie chart


MOST POPULAR TOPICS

Aug 7th, 12 → May 15th, 2(

# OF REVIEWS	LABEL
119	shipping
88	quality
74	leather
67	sustainability
58	voucher

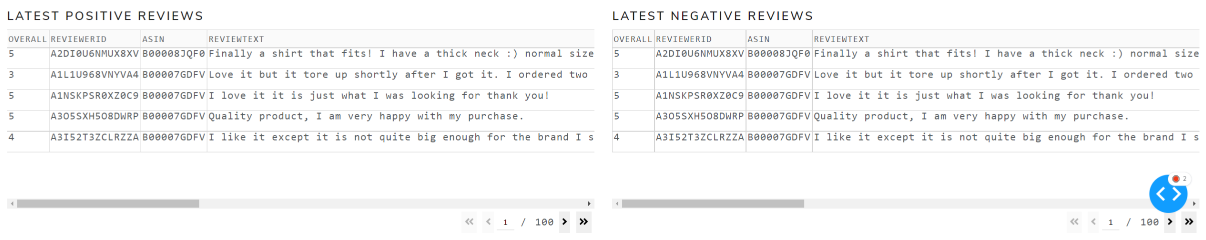
LABEL PIE CHART

Aug 7th, 12 → May 15th, 2(

Figure 22: Most popular topics and their distribution

7 further work

In the aforementioned business modelling section we brainstormed various features that are important for a useful customer feedback analytics platform. We started of building an MVP and iteratively added features. There were some aspects to the product that we think could improve this product even further, which we were not able to tackle due to



LATEST POSITIVE REVIEWS

OVERALL	REVIEWERID	ASIN	REVIEWTEXT
5	A2DI0U6NMUX8XV	B00008JQF0	Finally a shirt that fits! I have a thick neck :) normal size
3	A1L1U968VNYVA4	B00007GDFV	Love it but it tore up shortly after I got it. I ordered two
5	A1NSKPSR0XZ0C9	B00007GDFV	I love it it is just what I was looking for thank you!
5	A3O5SXH5O8DWRP	B00007GDFV	Quality product, I am very happy with my purchase.
4	A3I52T3ZCLRZZA	B00007GDFV	I like it except it is not quite big enough for the brand I s

1 / 100

LATEST NEGATIVE REVIEWS

OVERALL	REVIEWERID	ASIN	REVIEWTEXT
5	A2DI0U6NMUX8XV	B00008JQF0	Finally a shirt that fits! I have a thick neck :) normal size
3	A1L1U968VNYVA4	B00007GDFV	Love it but it tore up shortly after I got it. I ordered two
5	A1NSKPSR0XZ0C9	B00007GDFV	I love it it is just what I was looking for thank you!
5	A3O5SXH5O8DWRP	B00007GDFV	Quality product, I am very happy with my purchase.
4	A3I52T3ZCLRZZA	B00007GDFV	I like it except it is not quite big enough for the brand I s

1 / 100

Figure 23: Glimpse of the latest positive and negative reviews

time constraints.

- Multi-user login support: With the multiple user login support, we could have maintained different data tables for different users and can have multiple customers (from different companies) using this product.

- BERT based approach for sentiment model: We trained several models for Sentiment Analysis, however, keeping in mind the timeline of the project, we decided to use the best model we had at the end of the sprint in which we planned to work on sentiment analysis. However, had we trained a BERT based model, we could have potentially gotten a little more robust model because it would also work on Out-of-vocabulary(OOV) words.

- Pre-defined topics : In our current version of the product, the different topics or categories in which the reviews would be classified are pre-defined. However, we could have provided users with the functionality to dynamically create or delete some of the categories in which the reviews would be classified.

- Inclusion of online training: We have a fixed trained model that we use to make predictions on the new reviews. However, we would like to incorporate some of the feedback from the users into the app and then use this information to train the model again for a few epochs (Online training).

- NPS data : There was no publicly available NPS data for us to incorporate intelligence from that data into our dashboard. However, if we get some real-world data from the customers, we can also provide some customization and analysis tools specific to the NPS score.

- Notifications : We would also want to incorporate push-notification or email-notification system in which the users are notified if there are some new reviews.

- Automated Data Collection : Right now the user uploads the data in the form of CSVs. However, as a product this process would be automated with the help of APIs from Amazon or something else.

8 conclusion

The goal of this project was to solve the problem of analysing textual customer feedback, which is one of the biggest challenges of customer insight managers. Current competitors

are not able to solve this problem to an extent that satisfies our interview partners, who were working with customer feedback on a daily basis.
The main aim of this project was to get a working product which had significant business value. This meant that only working on the algorithms was not sufficient, rather we had to allocate significant amount of time to the business modelling and application development process as well. Integrating all three aspects together to make a coherent solution was among the greatest challenges in this project.
We went through three development sprints which resulted in a working solution to the problem which we deployed on Google Cloud Platform. The solution was modular as it has different components. We tried to stick to the best practices in software development. Finally, we have a product that does exactly what we were aiming for, which includes efficiently storing the reviews, calculation the sentiment, the topic of the reviews and displaying it in a visually appealing application to customers for analysis.
This project showed that it is possible to gain interesting insights from textual feedback and perform the two main tasks sentiment analysis and topic modeling on review data. Although due to time limitations the algorithms still have potential to improve, the results of this project can be seen as a proof of concept and together with the included recommendation for further work a real product that generates business value can be build and the interviews indicated that there is a demand for such a product.
Last but not least we want to thank our company partner FELD M and specifically Matthias for his time, help and guidance along the way.

Bibliography

References

[1] Stephanie Chevalier, *Global retail e-commerce sales 2014-2024*, (https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/, Accessed: 03.07.2021

[2] Forrester consulting, *Unlock the Power of Unstructured Feedback To Drive A Better Customer Experience*, https://reputation.com/forrester-opportunity-snapshot/, Accessed: 10.05.2021

[3] Ken Schwaber and Jeff Sutherland *The scrum guide*, Scrum Alliance21 (2011), p. 19.

[4] https://www.figma.com/

[5] https://dash.plotly.com/

[6] https://dash.plotly.com/dash-core-components

[7] https://plotly.com/python/

[8] https://getbootstrap.com/

[9] https://dash-bootstrap-components.opensource.faculty.ai/

[10] Jianmo Ni, Jiacheng Li, Julian McAuley, *Justifying recommendations using distantly-labeled reviews and fined-grained aspects*,
Empirical Methods in Natural Language Processing (EMNLP), 2019

[11] *VaderSentiment Documentation* https://pypi.org/project/vaderSentiment/

[12] *Textblob Documentation* https://textblob.readthedocs.io/en/dev/quickstart.html

[13] Justifying recommendations using distantly-labeled reviews and fined-grained aspects
Jianmo Ni, Jiacheng Li, Julian McAuley
Empirical Methods in Natural Language Processing (EMNLP), 2019

[14] https://cloud.google.com/docs

[15] https://www.postgresql.org/

[16] https://www.nginx.com/

[17] https://www.docker.com/

[18] https://docs.docker.com/compose/

[19] https://flask.palletsprojects.com/en/2.0.x/

[20] https://www.tensorflow.org/

[21] https://huggingface.co/

[22] Pennington, J., Socher, R. and Manning, C., *Glove: Global vectors for word representation*,
Empirical Methods in Natural Language Processing (EMNLP), In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) 2014

[23] Sepp Hochreiter, Jurgen Schmidhuber, *LONG SHORT-TERM MEMORY*,
Neural Computation, 1997

[24] Kim, Y. *Convolutional neural networks for sentence classification*,
arXiv preprint arXiv:1408.5882 (2014)

[25] Yoav Goldberg *A Primer on Neural Network Models for Natural Language Processing*
https://arxiv.org/abs/1510.00726v1 (URL, May 2018)

[26] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, Eduard Hovy *Hierarchical Attention Networks for Document Classification*
2016

[27] David M. Blei, Andrew Y. Ng, Michael I. Jordan
Latent Dirichlet Allocation , 2013

[28] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal and David M. Blei, *Hierarchical Dirichlet Processes* 2005

Appendix

Overview of the dataset

The columns present in the dataset were :
reviewerID - unique ID of the reviewer, e.g. A2SUAM1J3GNN3B
asin - unique ID of the product, e.g. 0000013714
reviewerName - name of the reviewer
vote - helpful votes of the review
style - a dictionary of the product metadata, e.g., "Format" is "Hardcover"
reviewText - text of the review
overall - rating of the product
summary - summary of the review
unixReviewTime - time of the review (unix time)
reviewTime - time of the review (raw)
image - images that users post after they have received the product

However, not all the columns were of relevance to us. As we were building a product that could be generically used by customers. Therefore, **columns like image, style, vote could not be used.**

Results from trained deep learning models

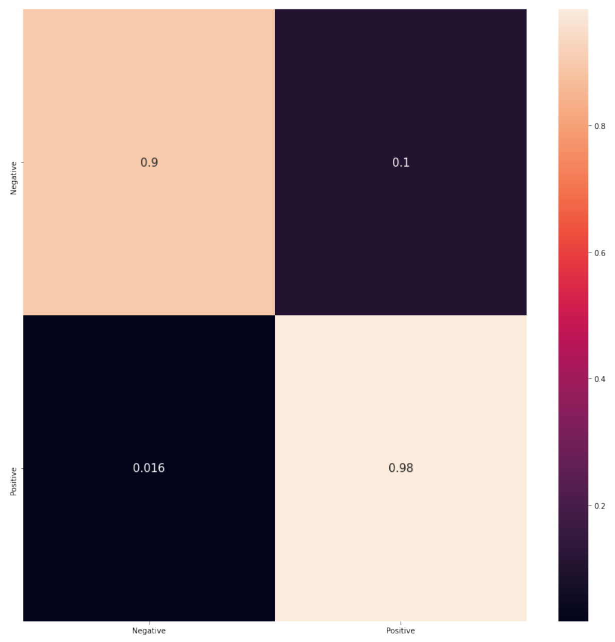


Figure 24: Confusion Matrix for bidirectional LSTM (best performing baseline model)

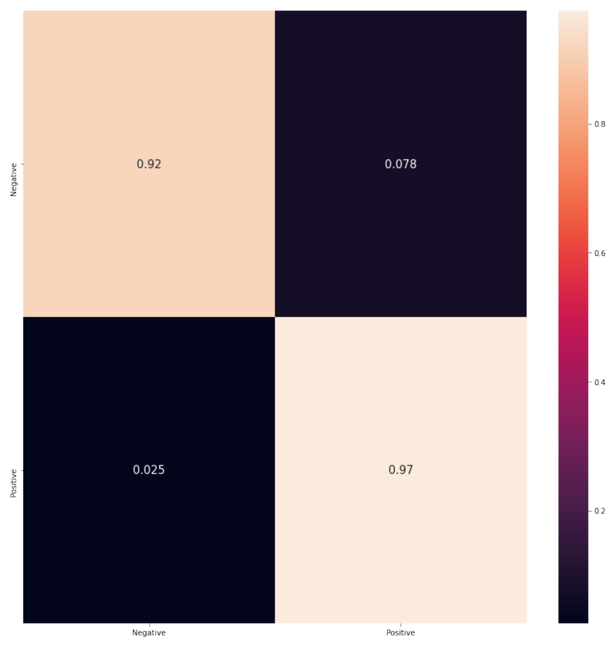


Figure 25: Confusion Matrix for unidirectional LSTM

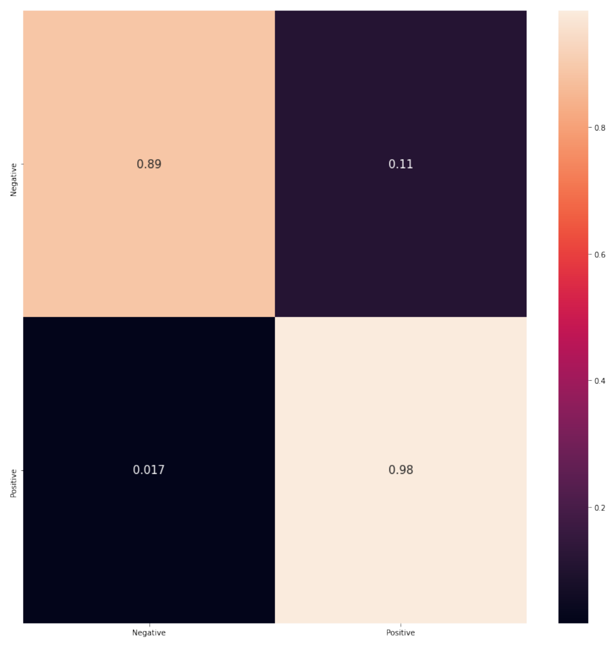
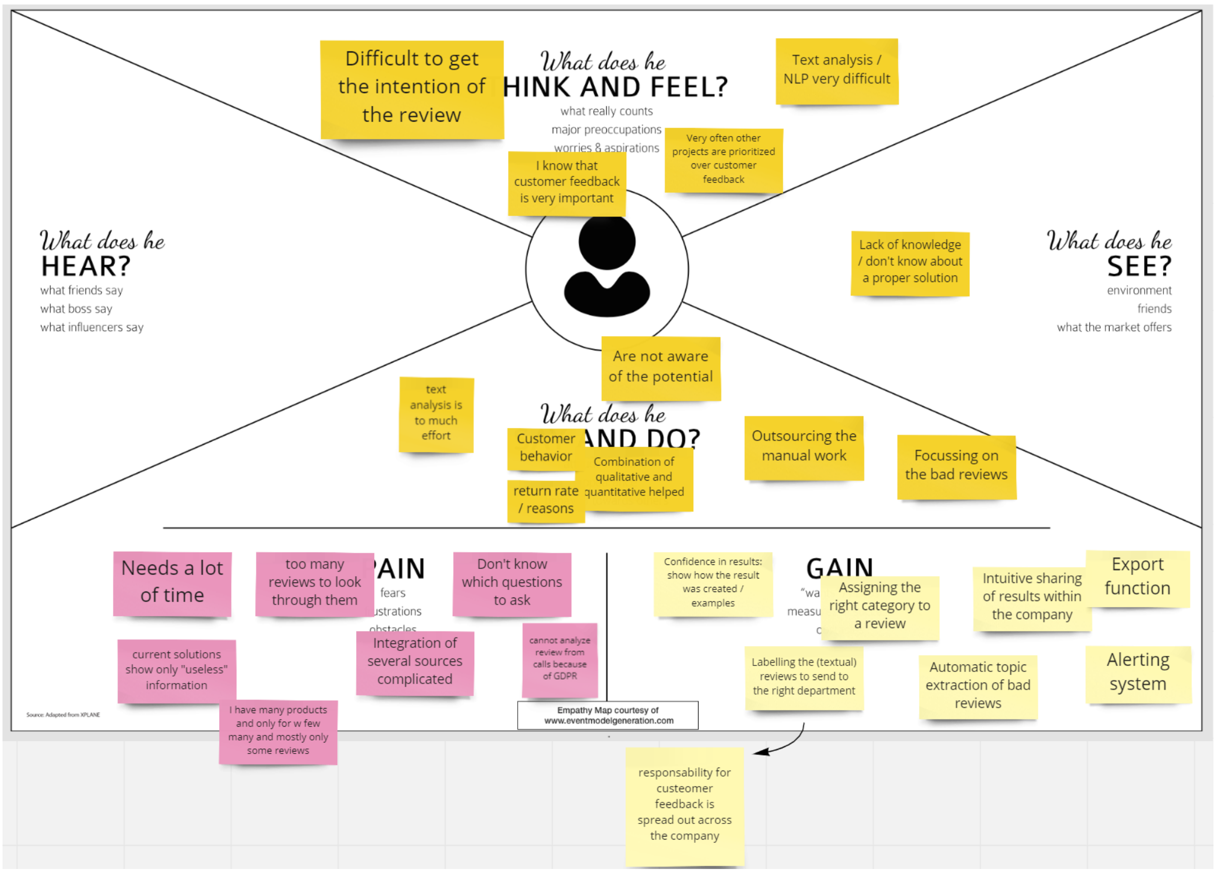


Figure 26: Confusion Matrix for CNN

Empathy Map

An empathy map is a collaborative tool used to determine what are the wants and needs of the customer, to better envision the solution that we are providing. After every interview, we filled out, as a team, the corresponding empathy map. We then, created a combined one in order to gather the recurrent themes and the shared experiences between the different interview partners.



What does he THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Difficult to get the intention of the review
Text analysis / NLP very difficult
Very often other projects are prioritized over customer feedback
I know that customer feedback is very important
What does he HEAR?
what friends say
what boss say
what influencers say
What does he SEE?
environment
friends
what the market offers
Lack of knowledge / don't know about a proper solution
Are not aware of the potential
What does he SAY AND DO?
text analysis is to much effort
Customer behavior
Combination of qualitative and quantitative helped
return rate / reasons
Outsourcing the manual work
Focussing on the bad reviews
PAIN
fears
frustrations
obstacles
Needs a lot of time
too many reviews to look through them
Don't know which questions to ask
current solutions show only "useless" information
Integration of several sources complicated
cannot analyze review from calls because of GDPR
I have many products and only for w few many and mostly only some reviews
GAIN
Confidence in results: show how the result was created / examples
Assigning the right category to a review
Intuitive sharing of results within the company
Export function
Labelling the (textual) reviews to send to the right department
Automatic topic extraction of bad reviews
Alerting system
responsability for custeomer feedback is spread out across the company
Empathy Map courtesy of www.eventmodelgeneration.com
Source: Adapted from XPLANE

Figure 27: Combined empathy map from all the interviews conducted in the market research phase

Workflow

Our workflow has been punctuated by sprint which are pre-determined time periods for which a certain set of goals is announced. At the end of every sprint, we look back on what was successfully achieved and what needs to be focused on more, during the next period.

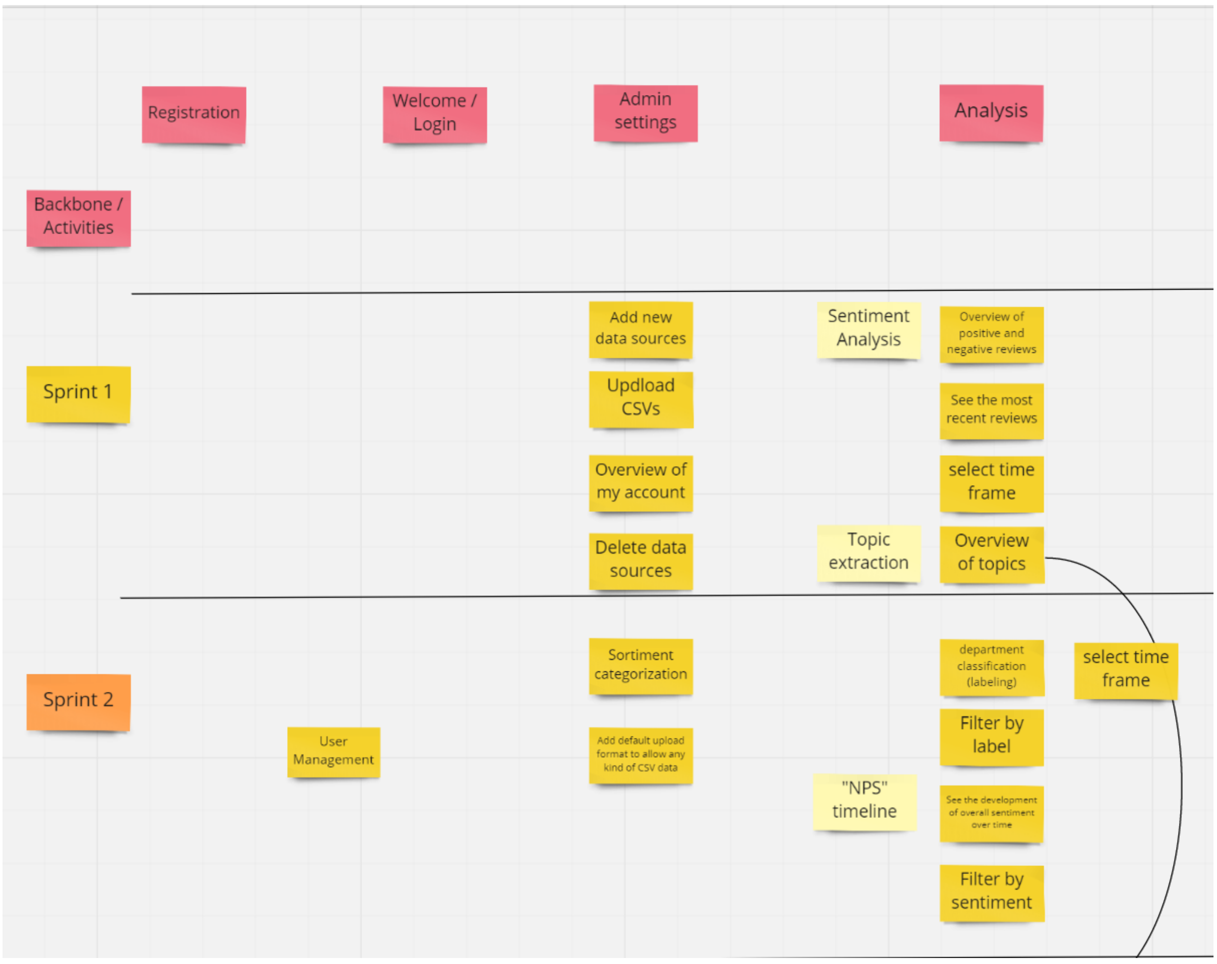


Figure 28: Overview of the tasks programmed for the 1st and 2nd sprint

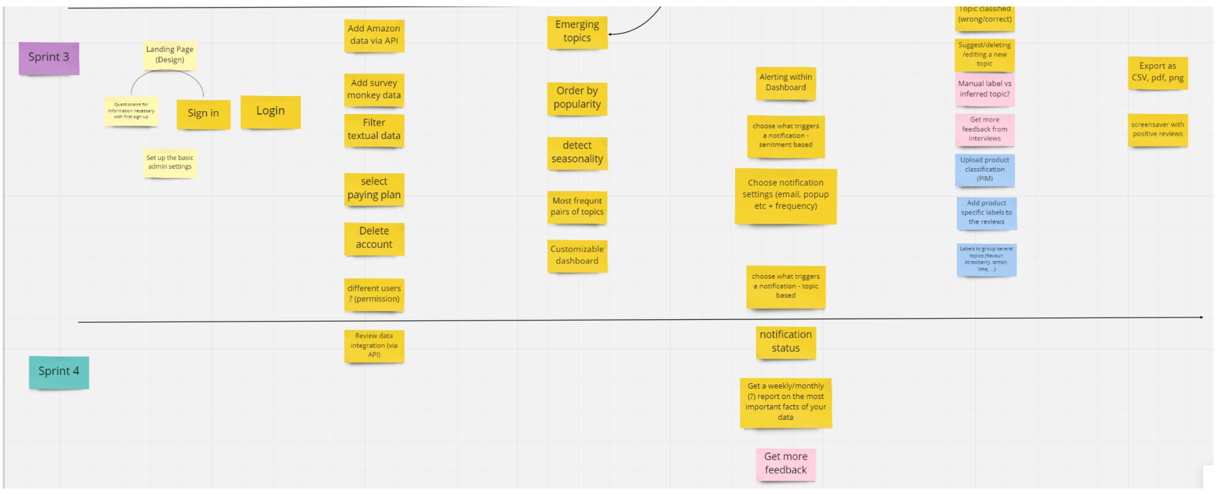

Sprint 3
Landing Page (Design)
Questionaire for Information necessary with first sign up
Sign in
Login
Set up the basic admin settings
Add Amazon data via API
Add survey monkey data
Filter textual data
select paying plan
Delete account
different users ? (permission)
Emerging topics
Order by popularity
detect seasonality
Most frequnt pairs of topics
Customizable dashboard
Alerting within Dashboard
choose what triggers a notification - senitment based
Choose notification settings (email, popup etc + frequency)
choose what triggers a notification - topic based
Topic classified (wrong/correct)
Suggest/deleting /editing a new topic
Manual label vs inferred topic?
Get more feedback from interviews
Upload product classification (PIM)
Add product specific labels to the reviews
Labels to group several topics (flavour: strawberry, lemon, lime, ...)
Export as CSV, pdf, png
screensaver with positive reviews
Sprint 4
Review data integration (via API)
notification status
Get a weekly/monthly (?) report on the most important facts of your data
Get more feedback

Figure 29: Overview of the tasks programmed for the 3rd and 4th sprint

User Story Mapping

User story mapping is an exercise where we determine and outline how the user interacts with our tool. During this exercise, we try to optimize the user's experience and prioritize the features that answer best to his needs.

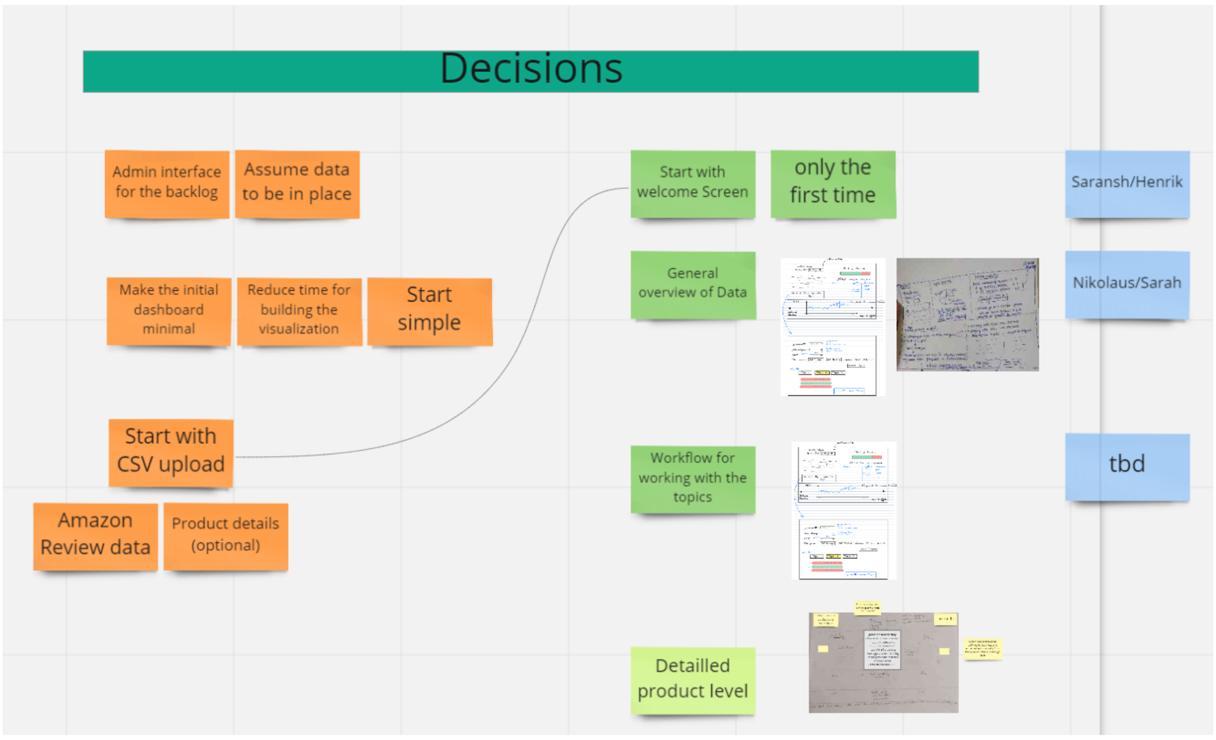

Decisions
Admin interface for the backlog
Assume data to be in place
Make the initial dashboard minimal
Reduce time for building the visualization
Start simple
Start with CSV upload
Amazon Review data
Product details (optional)
Start with welcome Screen
only the first time
General overview of Data
Workflow for working with the topics
Detailled product level
Saransh/Henrik
Nikolaus/Sarah
tbd

Figure 30: User story mapping done in a collaborative fashion

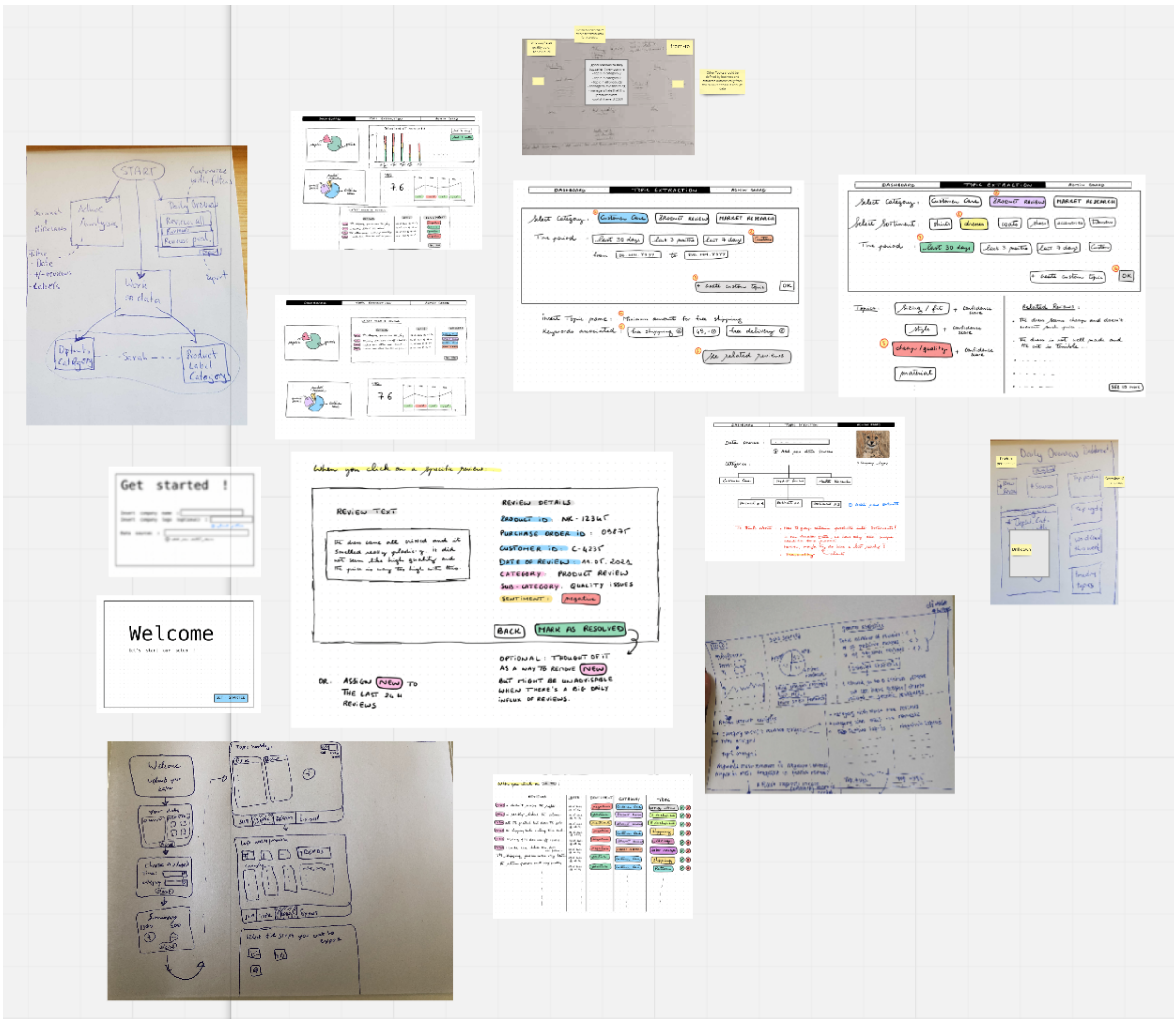


Figure 31: Various sketches of the Visualization tool

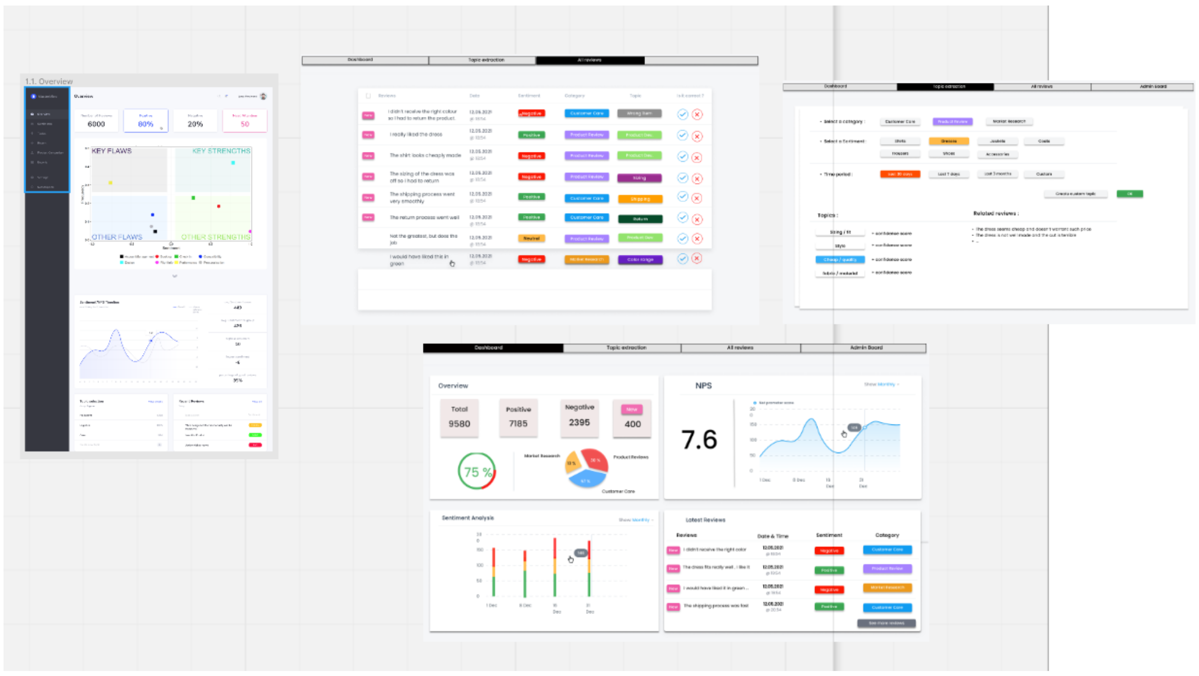


Figure 32: Prototypes developed using the *Figma* tool