Informatics

K-Means Clustering of Fashion Behavior

A Language-Focused Approach

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Software und Information Engineering

eingereicht von

Florian Dedov
Matrikelnummer 11913611

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Ass.in Mag.a rer.nat. Dr.in techn. Julia Neidhardt
Mitwirkung: Projektass. Dipl.-Ing. Thomas Elmar Kolb, BSc

Wien, 2. Oktober 2022

Florian Dedov

Julia Neidhardt

TU WIEN Informatics

Clustering of Customer Behavior in Fashion E-Commerce

A Language-Focused Approach

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Software and Information Engineering

by

Florian Dedov
Registration Number 11913611

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Ass.in Mag.a rer.nat. Dr.in techn. Julia Neidhardt
Assistance: Projektass. Dipl.-Ing. Thomas Elmar Kolb, BSc

Vienna, 2nd October, 2022

	Florian Dedov	Julia Neidhardt

Erklärung zur Verfassung der Arbeit

Florian Dedov

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 2. Oktober 2022

Florian Dedov

Danksagung

An dieser Stelle möchte ich mich bei Dr. Julia Neidhardt für ihre erstklassige Betreuung und Hilfe beim Verfassen dieser Arbeit bedanken. Ebenso gilt mein Dank Dipl.-Ing. Thomas Elmar Kolb, welcher mich sowohl technisch als auch inhaltlich stets unterstüzt hat.

Acknowledgements

At this point I would like to thank Dr. Julia Neidhardt for her first class support and mentorship while writing this work. Furthermore, I would like to thank Dipl.-Ing. Thomas Elmar Kolb who always supported me technically as well as in terms of subject-related questions.

Kurzfassung

Die Analyse von kundenbezogenen E-Commerce-Daten wird zunehmend wichtiger. Die meisten Analysen in diesem Bereich fokussieren sich auf die Produkte, wobei die Analyse des tatsächlichen Benutzerverhaltens oftmals zu kurz kommt. Um jedoch Kundengruppen besser ansprechen und wirklich verstehen zu können, was deren Präferenzen und Beweggründe sind, ist eine solche Analyse notwendig. Des Weiteren verlassen sich die meisten vorhandenen Arbeiten, in diesem Bereich, auf visuelle Features, um Benutzerverhalten zu gruppieren oder vorherzusagen. In dieser explorativen Arbeit verwenden wir einen alternativen Ansatz und untersuchen in welchem Ausmaß es möglich ist, bedeutende und aussagekräftige Kundencluster im Fashionbereich zu finden. Diese Cluster basieren primär auf der natürlichen Sprache, welche die Kunden in Rezensionen, von bereits gekauften Produkten, verwenden. Wir können zeigen, dass solche Features, welche wir auf Basis dieser natürlichen Sprache konstruieren, uns ermöglichen klare und bedeutende Cluster zu identifizieren. Dies ist ein erster und einleitender Schritt, um die psychologischen Faktoren, welche Kundenentscheidungen beeinflussen, zu verstehen.

Abstract

The analysis of customer-related e-commerce data becomes increasingly valuable over time. While most analysis in this competitive field focuses on the products, the analysis and categorization of the customer behavior itself is oftentimes less emphasized. However, to better target customer groups and to truly understand their preferences and needs, this is required. Furthermore, most existing analyses in that context primarily uses visual features to group instances or predict customer behavior. In this exploratory work, we take an alternative approach and examine to what extent it is possible to create meaningful and expressive clusters of customers in the fashion domain. These clusters are primarily based on the natural language customers use in their reviews of previously purchased products. We are able to show that engineering features based on the natural language used in customer reviews allows for identifying clear and meaningful clusters. This represents a first step into understanding psychological factors that impact customer choices.

Contents

CHAPTER 1

Introduction

1.1 Motivation

E-Commerce is constantly on the rise, which is why it becomes increasingly valuable to analyze user interaction data in order to be able to better understand what drives customer decisions and to predict their future behavior. Machine learning is heavily used by online stores to offer their customers personalized recommendations based on style, size and similar products [8]. Less research is available on more generic customer behavior characteristics in the context of fashion e-commerce [22]. The goal of this work is to use a clustering approach in order to find possible categories of fashion customers. However, rather than identifying these categories directly based on product features, the focus is on utilizing more user-centered features, which are mainly obtained through the analysis of the natural language used by customers in their product reviews. This strategy is based on the assumption that customers will often specify what they like or dislike and which key factors are crucial to them in their decision-making process. Since the fashion domain is strongly related to emotions and lifestyle, it is important to understand psychological factors that impact customer choices. This exploratory work is intended as a first step into this direction. Eventually it should lead to a characterization of users and user groups in a data-driven way that helps to enhance user models in fashion recommender systems.

1.2 Research Questions

The goal of this work is to answer the following two research questions:

RQ1: To what extent can the natural language used by customers, combined with a few other basic features, lead to somewhat robust and significant categorization of users?

RQ2: How important are the features based on the use of natural language compared to the other features that are also used for this categorization?

1.3 Expected Results

Our main aim is to explore whether features, based on the natural language used by customers, when interacting with E-commerce websites, can positively influence the creation of clear and meaningful clusters. Furthermore, we hypothesize that these features will be able to at least somewhat compete with other features in terms of relevance and importance, when it comes to the creation of the resulting clusters. We expect to see a decently strong positive or negative correlation between these language-based features and the resulting clusters. From these clusters we also expect to be able to extract interesting and meaningful interpretations of customer types.

Thus, the results of this work will be a machine learning model that identifies clusters, insights on which features are particularly relevant, as well as an interpretation of the clusters.

1.4 Methodology

Since the aim of this work is an unsupervised categorization of customers, we will use a clustering algorithm. Clustering algorithms group data based on some metric of similarity or proximity. In particular, we will use the K-Means clustering algorithm, which comes with its own advantages and drawbacks. It is a very simple clustering method, which scales easily to large data sets. Furthermore it guarantees convergence and is able to generalize to various shapes and sizes of clusters. The major disadvantage that we encounter when using the K-Means algorithm is that we have to specify the number of clusters k manually. We solve this problem by evaluating different models, using metrics, which do not require labelled data. A detailed description of our process and also of the clustering and the evaluation itself is provided in chapter 6.

In general, K-Means clustering is frequently used in the fashion domain, due to its simplicity [20, 1]. Oftentimes, the algorithm is modified to be less sensitive to outliers or to counteract the dependence on the random initialization. In this work, we will deal with pretty low-dimensional data that does not have too many outliers.

We explore and describe multiple data sets in chapter 3. Also, we explain our criteria and how we select the data set to perform our clustering on. To keep things simple, we choose to use the basic K-Means clustering algorithm without modifications. Furthermore, we test robustness with different samples in order to ensure that the random initialization does not skew the results.

The final results are then interpreted with a team that consists of experts in computer science, psychology and executives of a startup that is operating in the fashion domain.

This allows us to evaluate the quality of the clusters by looking at it from multiple different perspectives.

1.5 Structure of This Work

This work is organized as follows: In chapter 2 we discuss the state-of-the-art and related work. After that, in chapter 3, we explore multiple data sets in order to find one that is compatible with our goals stated above. In chapter 4, we then go deeper into the selected data set and describe it in more detail. We will look at the individual features, select the ones that seem useful for the task at hand and also look at some statistics and distributions. Chapter 5 will then be about data mining and feature engineering. Here we will use the data that is already present to create new features from it and to refine existing features. One key focus point will be to extract natural language features based on the customer's review texts. We will then proceed to do the actual user clustering in chapter 6. Different metrics for finding the optimal number of clusters will be looked at, followed by the clustering itself. Finally, in chapter 7 the results will be discussed. We will look at the various clusters and how they correlate with the individual features. Furthermore, we will offer possible interpretations and strategies based on the results, as well as answer the research questions stated above.

CHAPTER 2

State-of-the-Art

Recommendation systems in the fashion field is a highly researched area [16, 2, 18, 21, 10]. However, most of the work focuses on the products instead of the customers. In addition to that it usually has a strong focus on visual features [10, 21]. In the work of Cassel et al. [2] Amazon customer data was used to build a recommender system, but the approach was trying to improve scalability, felxibility and portability instead of aiming at understanding customer behavior. The work of Kang et al. [10] focuses on recommending products based on the visual preferences of customers. It also generates new images that could match the customers taste. The paper of Yan et al. [21] aims to extract visual features of products for their clustering. It does not aim to cluster end-users. In the work of Monte et al. [16] data is clustered once for the purpose of design and once for the purpose of marketing. The focus is on the product and the natural language used by the customer is not looked at. Finally, in the work of Sevegnani et al. [18] natural language is used for building a fashion recommender system. However, it follows a very active approach, where the customer has to state a query, which is then acted upon by the system. There is no analysis of general customer behavior while online shopping.

In this work on the other hand, we will focus exclusively on clustering customers based on their behavior with a heavy emphasis on natural language features for the reasons mentioned previously.

CHAPTER 3

Exploration of Data Sets

The first task is to explore various fashion e-commerce data sets in order to find one, which could be useful for the goals of this work. This data set has to be centered around customer behavior and should preferably have multiple non-visual features, some of which should optimally include natural language used by the customers.

To find potential candidates, we do some literature review in the hope of finding a structured and professional collection of data sets to use. While doing so, we find two interesting resources [19, 12] mentioned in existing work. First, we have the *FashionXRecsys*[1] challenge, which lists multiple interesting and promising data sets. Second, we also have an additional data set that could be potentially interesting, provided by the RecSys Challenge 2022[2].

Many of the data sets provided by *FashionXRecsys* have a strong visual focus and are product-centered rather than customer-centered. The *Large-scale Fashion* data set [13] and the *DeepFashion2* data set [4] contain images along with some metadata and categorical information. They could both be very useful when attempting to recommend fashion items based on visual similarity. Furthermore, they can be useful for recognizing fashion items or types of items. However, since they are primarily visual and especially since they are not in any way focused on the customer, they are not useful in the context of this work.

The *Clothing Fit* Dataset for Size Recommendation [15] seems to be more interesting, because it contains user-centered information. However, as the name already suggests, it is solely focused on size and fit information. The data is split into two different files, which represent data from two different stores (*Modcloth* and *RentTheRunway*). Alongside

[1] https://fashionxrecsys.github.io/fashionxrecsys-2022/

[2] http://www.recsyschallenge.com/2022/

some basic identifiers, the data contains detailed information about sizes, proportions and user ratings on how well certain pieces of clothing fit on them. All this makes this data set very useful for the task of size or fit recommendation. However, the data set is not very interesting when it comes to general fashion recommendations, based on user preferences. It is reasonable to assume that almost all customers are interested in clothes that fit them well. Thus, this data set does not provide any features that can be used to group or categorize customers based on their unique preferences (e.g. favorite color, price range etc.) beyond the dimension of size and fit. Nevertheless, we could keep this data set in mind for additional size-focused analysis. If we notice for example that there is a category of users, which focuses a lot on size and fit or struggles with this topic a lot, this might turn out to be a great complementary data set to go into further analyses, provided that the two sources of data are somewhat compatible.

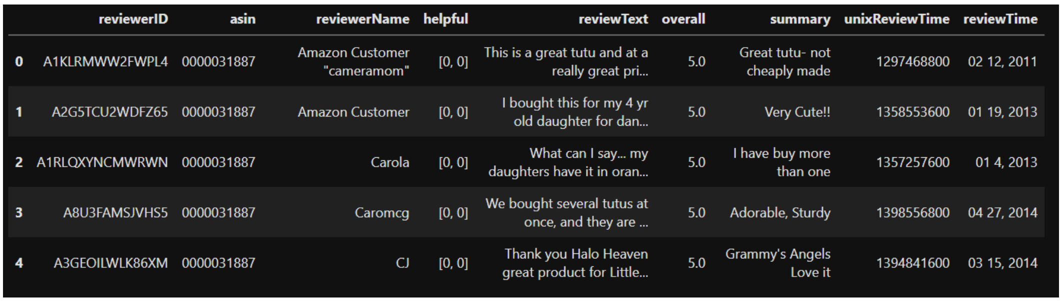
Another size-related data set is the *ViBE* data set [7]. It contains size information about humans and metadata about clothes. However, the humans that the data is referring to are not customers but models, which makes a customer-centered analysis impossible. In addition to that, the data has an unusual structure. There is a text file for each item, that contains unstructured textual information. One line in such a text file is one feature of the item. This makes the data set quite hard to use. Therefore, we keep looking for a better alternative.

The *Street2Shop* data set [11] and Pinterest's *Shop The Look* Dataset [9] are almost entirely focusing on visual aspects of the fashion items. They provide a little bit of additional metadata (like positional information and categories) but there are no features that are related to customers and also no language-oriented features, which makes the two data sets uninteresting for this work. They are mainly useful for clustering products based on their visual features.

One data set that seems promising at first is the *Alibaba iFasion* data set [3]. On the FashionXRecsys website it says that the data set contains over one million outfits and click behaviors of over three million users. This sounds like a very well-suited data set for the task of user-centered clustering. However, unfortunately, it turns out to be very hard to use, since the data consists mostly of presumably Chinese text. The image links could be used for visual product clustering, but for the purposes of this work, this data seems to unfortunately be mostly unusable, especially because there is essentially no documentation.

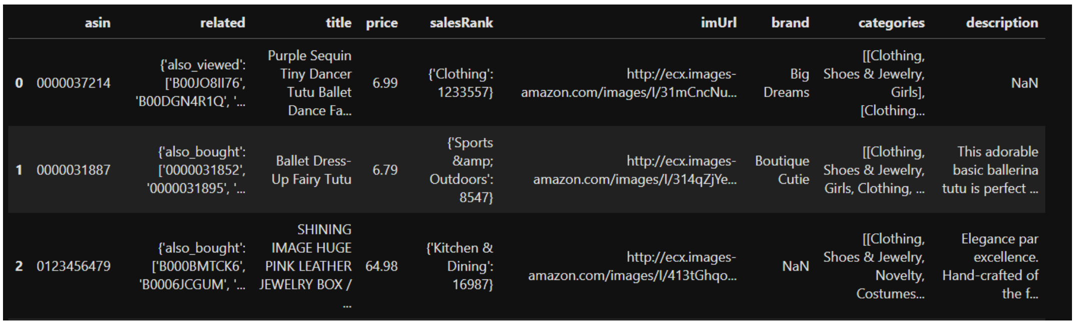
Another interesting data set is the *Polyvore* data set [5]. It contains outfits that consist of multiple fashion items, which were selected by real users. The features provided include the price, the category and an image of the product. Since the outfits were put together by actual users, the data is user-centric and could be useful for clustering customers. However, there is not much information about user behavior (i.e. product reviews etc.)

and the data does not contain any information about buying behavior. So even though this data set seems interesting, there might be better alternatives.



	reviewerID	asin	reviewerName	helpful	reviewText	overall	summary	unixReviewTime	reviewTime
0	A1KLRMWW2FWPL4	0000031887	Amazon Customer "cameramom"	[0, 0]	This is a great tutu and at a really great pri...	5.0	Great tutu- not cheaply made	1297468800	02 12, 2011
1	A2G5TCU2WDFZ65	0000031887	Amazon Customer	[0, 0]	I bought this for my 4 yr old daughter for dan...	5.0	Very Cute!!	1358553600	01 19, 2013
2	A1RLQXYNCMWRWN	0000031887	Carola	[0, 0]	What can I say... my daughters have it in oran...	5.0	I have buy more than one	1357257600	01 4, 2013
3	A8U3FAMSJVHS5	0000031887	Caromcg	[0, 0]	We bought several tutus at once, and they are ...	5.0	Adorable, Sturdy	1398556800	04 27, 2014
4	A3GEOILWLK86XM	0000031887	CJ	[0, 0]	Thank you Halo Heaven great product for Little...	5.0	Grammy's Angels Love it	1394841600	03 15, 2014

Figure 3.1: Amazon Data Set Reviewer Data



	asin	related	title	price	salesRank	imUrl	brand	categories	description
0	0000037214	{'also_viewed': ['B00JO8II76', 'B00DGN4R1Q', '...	Purple Sequin Tiny Dancer Tutu Ballet Dance Fa...	6.99	{'Clothing': 1233557}	http://ecx.images-amazon.com/images/I/31mCncNu...	Big Dreams	[[Clothing, Shoes & Jewelry, Girls], [Clothing...	NaN
1	0000031887	{'also_bought': ['0000031852', '0000031895', '...	Ballet Dress-Up Fairy Tutu	6.79	{'Sports &amp; Outdoors': 8547}	http://ecx.images-amazon.com/images/I/314qZjYe...	Boutique Cutie	[[Clothing, Shoes & Jewelry, Girls, Clothing, ...	This adorable basic ballerina tutu is perfect ...
2	0123456479	{'also_bought': ['B000BMTCK6', 'B0006JCGUM', '...	SHINING IMAGE HUGE PINK LEATHER JEWELRY BOX / ...	64.98	{'Kitchen & Dining': 16987}	http://ecx.images-amazon.com/images/I/413tGhqo...	NaN	[[Clothing, Shoes & Jewelry, Novelty, Costumes...	Elegance par excellence. Hand-crafted of the f...

Figure 3.2: Amazon Data Set Product Metadata

This leaves us with the remaining two data sets, one listed on the FashionXRecsys website [14] and one used for the RecSys Challenge 2022[3]. Both data sets contain information about user behavior, like buying or clicking on products, together with some additional metadata. The problem with the *Dressipi* data set[3] however is that it is anonymized for the purposes of the challenge. The data can be used to train and evaluate a model, but we cannot know what the various features represent. This disqualifies the data set for our purposes. The Amazon review data set [14] on the other hand, seems to be exactly what we are looking for. It contains information about the buying behavior of users (see Figure 3.1), their written product reviews and some additional metadata (see Figure 3.2) like price and sales rank. The data is user-centered and we can use the review texts in the context of natural language processing (NLP) to engineer new features. Furthermore, images of the individual products are provided and can be used to eventually craft some visual features as well. These, combined with other existing or engineered features, can be used for clustering alongside with the NLP features, to see if the natural language used by customers has a significant influence on the resulting clusters.

[3] http://www.recsyschallenge.com/2022/

Data Set	**Reference**	**Focus**	**Problems**
Large-scale Fashion	Liu et al., 2016 [13]	Visual	Only Visual
DeepFashion2	Ge et al., 2019 [4]	Visual	Only Visual
Clothing Fit	Misra et al., 2018[15]	Size	Only Size
ViBE	Hsiao et al., 2019 [7]	Size and Other	Data About Models Not Customers
Street2Shop	Kiapour et al., 2015 [11]	Visual	Only Visual
Pinterest Shop The Look	Kang et al., 2018 [9]	Visual	Only Visual
Alibaba iFashion	Chen et al., 2019 [3]	Click Behavior and Visual	Not Documented Unusable (Language)
Polyvore	Han et al., 2017 [5]	Outfit Creation	User Behavior Not Relevant Enough
Dressipi	RecSys Challenge, 2022[4]	Click Behavior	Anonymized Data
Amazon Reviews	McAuley et al., 2015 [14]	Buying and Reviewing Behavior	-

Table 3.1: Data Set Overview

So after looking at the various data sets, we decide to use the Amazon review data [14]. In particular, we use the subset focusing on *Clothing, Shoes and Jewelry*, since this is the niche that our analysis is targeting. Additional analysis of other data sets will not be part of this work. It is useful however to mention that some of the data sets listed above could be very interesting and seem compatible with the Amazon review data set. Especially the Dressipi data[4] is promising for a comparison of results across data sets. For this however, it would be necessary to contact the owners in order to obtain the deanonymized data.

[4] http://www.recsyschallenge.com/2022/

CHAPTER 4

Data Set Description and Preprocessing

4.1 Data Set Overview

Now that we have selected a data set to work with, we will look deeper into it, in order to prepare for the data mining and feature engineering. The Amazon data set is split into two parts. The first part is the data representing the buying and reviewing behavior of the customers. Here we have an *identifier* for the *user* and one for the *product* bought. In addition to that, we have the *review* of the product, a *summary* of the review, the given *rating* and some other features like *timestamps* and the reviewer's *username*. The second part is the metadata for the various products. Here we have the same product identifiers that can be found in the first part of the data set. The metadata includes the product *description*, the *price*, the *sales rank*, an *image link* and a few more features.

Before we further analyze, drop or scale features, it makes sense to merge the data on the product identifier, so that we end up with one data frame that contains everything. After doing that, each row represents a user buying and rating a product. The metadata of the respective product is also contained in the same row, which makes the data frame redundant but easier to work with.

Broadly speaking, we can categorize the features of the data set into four categories - the *identifiers*, the *numerical features* that can already be used without further engineering, the features that can be used as the *basis* for the creation of new features, and *dropped features* that do not provide any additional knowledge or insight into the customer behavior for the purposes of this work.

4.2 Preprocessing

4.2.1 Identifiers and Numerical Features

The two identifiers in the data set are the *ReviewerID*, which identifies the customer and the *ASIN*, which identifies the product. The only two numerical features that we will use without further processing are the *price* and the given *rating*. Making use of the *sales rank* will require further relativization, since the number is always specified relative to a category. We can see the maximum sales ranks for a few selected categories in the table down below (Table 4.1). Achieving a higher sales rank in a small niche (i.e. max rank 38,660 for Computers & Accessoires) is much easier than ranking high in a major general category (i.e. max rank 3,597,088 for Clothing). This needs to be taken into consideration when adjusting and using the sales rank as a feature for clustering. There is one more numeric feature, which we will not use for this work. It indicates whether other users found the given review helpful or not. This feature might be interesting with respect to the product, but it doesn't tell us much about the customer that we are looking to categorize.

Category	**Max Rank**	**Category**	**Max Rank**
Jewelry	1,720,545	Sports & Outdoors	2,973,577
Watches	300,143	Clothing	3,597,088
Beauty	1,204,670	Baby	315,134
Computers & Accessoires	38,660	Shoes	1,090,188

Table 4.1: Sales Ranks

Another thing that is noteworthy is that there are a lot of Not-a-Number (NaN) values in the data set for the price feature. There are 278,677 entries and only 113,912 contain a valid price, which is a bit more than 40%. To resolve this issue, we have multiple options. One option is to drop the price feature entirely and focus on the other metrics. However, we decide to not do this, because of the the price a customer usually pays is a presumably important piece of information, which we want to use at least as a descriptive feature after clustering. Another option is to impute the missing values, which can also be done in multiple different ways, like using the mean price or taking the price of a similar product. The problem with this option is that we will probably not be able to estimate the actual price closely enough for the price to be a meaningful feature, especially considering that we would be artificially filling up almost 60% of the data. This leaves us with the final solution, which is to drop all the entries where the price is missing. Dropping these entries will lead to a new issue though. In the original data set every reviewer occurs at least a certain number of times to make the data representative and useful for predictions or analyses. Dropping certain entries might potentially lead to some reviewers only occurring once or twice in the data set afterwards. Because of that, we choose to limit the data to users that occur at least three times after dropping the rows where the price was missing. This leaves us with 19,271 entries, which can be used for clustering. Even though we lose a large part of the data by doing all that, the data set is still large enough to allow for a representative and meaningful cluster analysis. The price will be used descriptively after the clustering is done, in order to examine how the resulting clusters relate to the “unseen” price feature.

4.2.2 Base Features

Besides these numeric features, we also have some features that cannot be used right away, but which will serve as the basis for the feature engineering in the next section. The most obvious one is the *review text*. It will be the basis for all our NLP features that we create. We choose to not use the *summary*, *title* or *description* in this work. The *summary* is essentially a stripped down version of the *review text* and the other two features do not contain any natural language used by the user. Another interesting feature is the *image url*, which we can use to download the product images and extract visual features from them. Two additional features that we will not use in this work are the *brand* and the *categories*. The *brand* has a lot of NaN values and is mostly present for watches. By using that feature and crafting features based on it, we would either limit our data set to mostly watch purchases or we would have to deal with a lot of NaN values. The *categories* feature is a very product-centered one, which contains multiple categories per product. It could be useful as the basis for additional features in future work, but will be dropped in this work, due to the focus on user-centered and NLP features.

4.2.3 Dropped Features

Last but not least, there are a couple of features, which are not very useful in the context of this work. There are two columns that include *timestamps*, which indicate when a review was posted. Also, there is a column, which contains the *username* of the reviewer. These three features, will most likely not be able to add any value to our cluster analysis. The last column that remains is the one containing *related products*. This column contains ASINs of products that either Amazon considers to be similar to the respective product or that people have also bought, when buying the given product. This feature could be useful for recommending products based on similarity. However, it does not seem to be of much value when categorizing customers.

4.2.4 Additional Features

The table below (Table 4.2) gives us an overview of the features we have and what we are going to do with them. The identifiers and the numerical features will be kept the way they are and maybe extended. Such an extension makes sense, when we look at the distribution of the *price* feature for example (see Figure 4.1).

Identifiers	**Numerical Features**	**Base Features**	**Dropped Features**
ReviewerID	Price	Sales Rank	Title
ASIN	Overall Rating	Review Text	Review Summary
		Image URL	Description
			Helpfulness
			Username
			Timestamps
			Categories
			Brand
			Related Products

Table 4.2: Amazon Data Set Feature Overview

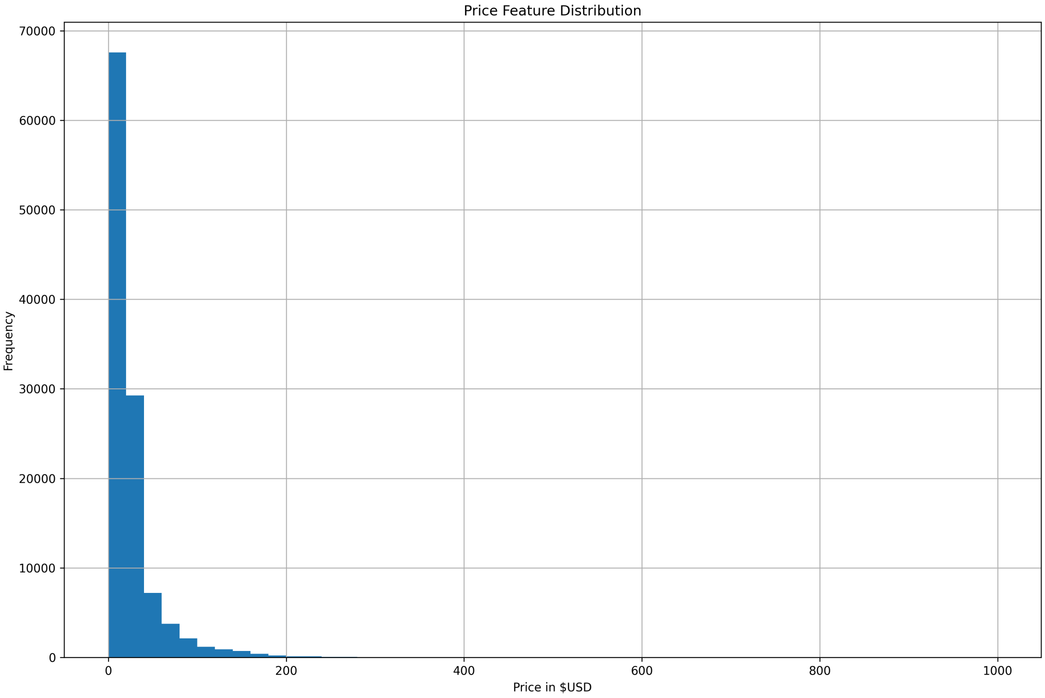


Figure 4.1: Price Feature Histogram

We can see that the price feature has a very skewed distribution. Since we are going to use it descriptively after the clustering, we can keep this feature and do not have to replace it. However, it will make sense to create some additional feature based on it (like taking the logarithm or looking at the standard deviation), to see if those give us more insight into the clusters than the raw absolute numbers.

The other base features will be used to extract new features from them and they will not be used the way they are right now. We need to turn them into representative numerical features, which in case of the *sales rank* means relativizing it and in the case of the *review texts* and the *image URLs* means extracting entirely new features from them. Our clustering algorithm will need numerical data. We cannot just feed text or images into it, unless we want to input meaningless ASCII codes and pixels. Extracting these features will be the focus of the next section.

CHAPTER 5

Data Mining and Feature Engineering

5.1 Relativization of Sales Rank

Having discussed all the available features, we now proceed to engineer additional features that we will use for clustering or as descriptive values afterwards. For this, we will start by relativizing the sales rank feature to make it more representative. We already mentioned that the sales rank represents the popularity inside of one specific category. A high rank in a very broad and general category with many different items is more impressive than a high rank in a very narrow niche category with only a couple of items. Thus, we want to transform the feature a into numerical value, which can give us information about the cross-categorical popularity. The formula down below (Formula 5.1) shows how we choose to do this. We define the relative sales rank SR_{rel} to be the result of dividing the absolute sales rank SR_{abs} by the maximum rank available in the data set SR_{max_c}, for the respective category c.

$$SR_{rel} = \frac{SR_{abs}}{SR_{max_c}} \tag{5.1}$$

Since we are limited to the information provided by this data set, we have to use the maximum rank available in the data set rather than the actual maximum rank of the category. However, this should be sufficient to take into account the differences between a very broad and a very narrow category. We do not explicitly include any minimum rank into the formula, since the best rank of any category is always one. With this new *relative sales rank* feature, we have a numerical indicator for the cross-categorical popularity of the various items. Every value will be a number between zero and one. The closer the metric is to zero, the more popular the item, relative to the size of the category it is in.

5.2 Natural Language Features

Next, we will craft the central features of this work, the natural language features. As already mentioned, the focus of this work is primarily to cluster customers based on their behavior, which includes the reviews they write and the words they use when doing so. Our goal here is to extract topics that the customers write about when reviewing a purchased product. We are trying to determine what their major concern is when stating their opinion regarding an item. For this, it is important to first define a limited amount of topics to look for in the first place. It is not reasonable to just look at each review individually and to try to find one or two key topics per text. It makes more sense to look at all the reviews and extract the most commonly used words, in order to then determine a few key categories to look for in each individual review. While doing this, we split the words into nouns, verbs and adjectives in order to look for topics in the various types of words. If we were to just examine the most common words in general, we might get a lot of meaningless conjunctions or we could get only one word type, because it occurs more often in general than the other types, regardless of the topic.

```
Adjectives: ('great', 78707), ('good', 68904), ('comfortable', 58203),
↪  ('nice', 47601), ('small', 43475), ('little', 39252)...

Verbs: ('have', 138569), ('wear', 123711), ('look', 85931), ('get',
↪  76842), ('buy', 76648), ('love', 75616)...

Nouns: ('size', 91905), ('shoe', 82022), ('color', 50351), ('foot',
↪  40495), ('pair', 39807), ('time', 39314)...
```

Figure 5.1: Most Common Words

Above (Figure 5.1) you can see the six most common words per word type. For this work however, we consider the top 300 words per word type. By looking at the most used words manually we can spot a couple of different underlying categories. Here, we choose to limit ourselves to the five most obvious and commonly spotted patterns. Those are: *quality words*, *color words*, *beauty words*, *price words* and *size or fit words*. When looking at the 100 most common words, most of them can be associated with one of the categories above.

To now craft five features that quantify the language use of the customer, we need to come up with a way of categorizing and filtering the individual words. For this, we choose one or two representative *reference words* for the whole category. We then calculate a *similarity score* [6] between the reference words and the word that has to be categorized. If this score surpasses a certain threshold, the word is counted as a word belonging to that category. Notice that, in theory, this means that one word could belong to multiple categories.

Category	**Word A**	**Threshold A**	**Word B**	**Threshold B**
Quality	"quality"	>0.5	-	-
Color	"color"	>0.5	-	-
Beauty	"beautiful"	>0.5	"cute"	>0.6
Price	"price"	>0.5	"cost"	>0.5
Size	"size"	>0.4	"fit"	>0.6

Table 5.1: Reference Words Thresholds

In the table above (Table 5.1), you can see the chosen reference words and the respective thresholds. We choose 0.5 as the default similarity threshold and adjust it by increasing or decreasing it by 0.1 if necessary. For example, we choose 0.6 as the threshold for "fit" because 0.5 includes too many words that do not conceptually belong to size-related language use. Likewise, we use 0.4 as the threshold for "size" in order to find more words that are actually size-related. For calculating the similarity score as well as for determining whether a word is a noun, verb or adjective, we use the Spacy [6] module in Python.

The five features that we now create based on these categories represent the number of words used in the respective reviews per topic. However, one problem that we might encounter with this approach is that some customers probably tend to write very long reviews in general, while other customers limit their texts to a few words. Thus, looking at the absolute word counts per topic might be misleading. The features are very skewed (see Figure 5.2).

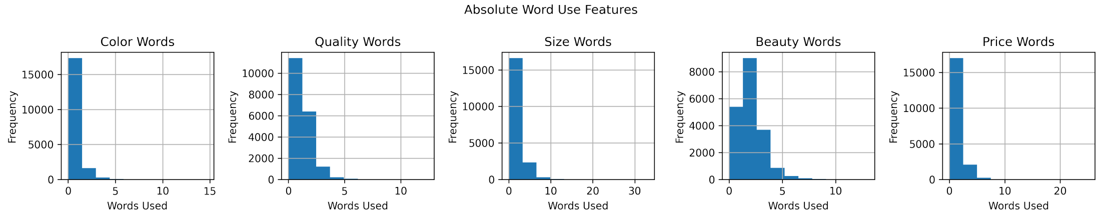


Figure 5.2: Absolute Word Use Features Histograms

Someone who writes a thousand words has a higher probability to include words of any category than someone who writes only ten. Because of this, we will make this feature relative to the total number of words used in a review.

$$WC_{rel_{cat}} = \frac{WC_{abs_{cat}}}{WC_{total}} \tag{5.2}$$

The formula above (Formula 5.2) shows how we relativize the feature. The relative word count with respect to a category $WC_{rel_{cat}}$ is the result of dividing the absolute word count with respect to a category $WC_{abs_{cat}}$ by the total number of words used the review WC_{total}.

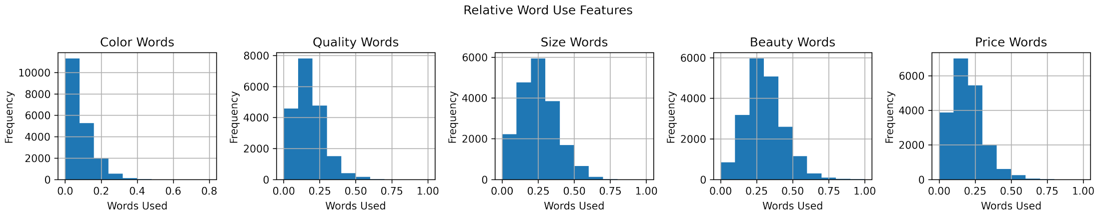


Figure 5.3: Relative Word Use Features Histograms

After we have applied this formula to our features, we can see that the distributions are now much less skewed (see Figure 5.3), which makes them much more representative. The *color words* feature still seems to be quite skewed, indicating that only few people talk about specific colors.

5.3 Color Diversity Feature

Next, we will engineer a small visual feature that will indicate how diverse the customer is in their choice of color when buying products. Since we don't have any visual metadata about the products, we need to use the *image URLs* that are provided in the data set to access the actual images and extract the dominant colors. To accomplish this, we use a function that utilizes caching to download the individual images in multiple threads. We then take the image and extract the two most dominant hex color codes from the images, using the library *colorthief*[1]. The function is cached so that we do not have to go through the whole process multiple times for the same URL. Once we have the two color codes, we use the Python module *webcolors*[2] to assign an actual color name to the code. This is done by finding the closest color code, which has a label and using it. We do this in order to reduce the number of distinct values that we have.

To find, which color is the closest one, we iterate over all colors we have labels for and calculate the squared distance to the color, we are trying to label. Since, we are dealing with hex color codes, we first have to transform those to RGB values (red, green, blue) before we compute the differences for the individual channels. The first two digits of the hex code represent the red value, the second two the green value and the last two the blue value. All we have to do is to convert these numbers from hex to decimal, by multiplying the first digit of each pair by 16 and the second digit by one. After doing this, we end up with three separate values, representing the red, green and blue values of

[1] https://github.com/fengsp/color-thief-py

[2] https://pypi.org/project/webcolors/

our color. The squared distance d from our sample color SC and our reference color RC is now calculated by using the formula down below (Formula 5.3).

$$d(SC, RC) = \sum_{c \in \{r,g,b\}} (SC_c - RC_c)^2 \tag{5.3}$$

The control variable c represents one of the three channels of the color. We calculate the squared distance between all three channels and sum up the results to get our metric. The color, which minimizes this metric with regards to our sample color, will be chosen as the reference color to take the label from.

To now craft our visual feature - and also in order to craft or enhance other features - we need to group the data by the *reviewer id*. Remember, that we are not clustering products or individual reviews, but customers. For the purpose of grouping by the user identifier, we need to first drop all the columns that we can not aggregate. Furthermore, we need to think about, which aggregation function we want to use for the columns we want to keep.

Column	**Aggregate Function**
Price	Mean & Std
Overall Rating	Mean
Relative Rank	Mean
<All Word Features>	Mean
Colors	Special Formula

Table 5.2: Aggregation Functions For Grouping Features

You can see which aggregate functions were used for the individual columns in the table above (Table 5.2). We use the *mean* and the *standard deviation (Std)* function for the price, because we craft two separate features from one feature here. For the colors we extracted and labeled we do not use a simple aggregate function but a special formula for calculating the *color diversity* Θ. It is the result of dividing the *distinct* number of colors across all purchases n_u by the *total* number of colors across all purchases n.

$$\Theta = \frac{n_u}{n} \tag{5.4}$$

This metric (Formula 5.4) indicates how open-minded or diverse a customer is in their purchases regarding the color. It will be especially interesting to examine if there is any relationship or correlation between this feature and the color-related words used by a customer in their reviews.

5.4 Price-Related Features

The last two features that we will add to our data set are the *relative price standard deviation* and the *logarithm* of the price. It can be interesting to not only look at the average price a customer pays for clothes but also at how much the various purchases deviate from that average. A customer that buys products across all price ranges might end up in a different cluster than a customer that buys only very expensive or only very cheap products. Since, we are interested in the relative standard deviation σ_{rel}, we first calculate the *absolute standard deviation* σ_{abs} and then divide it by the average price μ (Formula 5.5).

$$\sigma_{rel} = \frac{\sigma_{abs}}{\mu} \tag{5.5}$$

Regarding the price, it makes sense to also look at the logarithm of it and see if there are any differences in the results we get. This is especially useful when the distribution of a feature is skewed, which is the case for the price as we saw earlier.

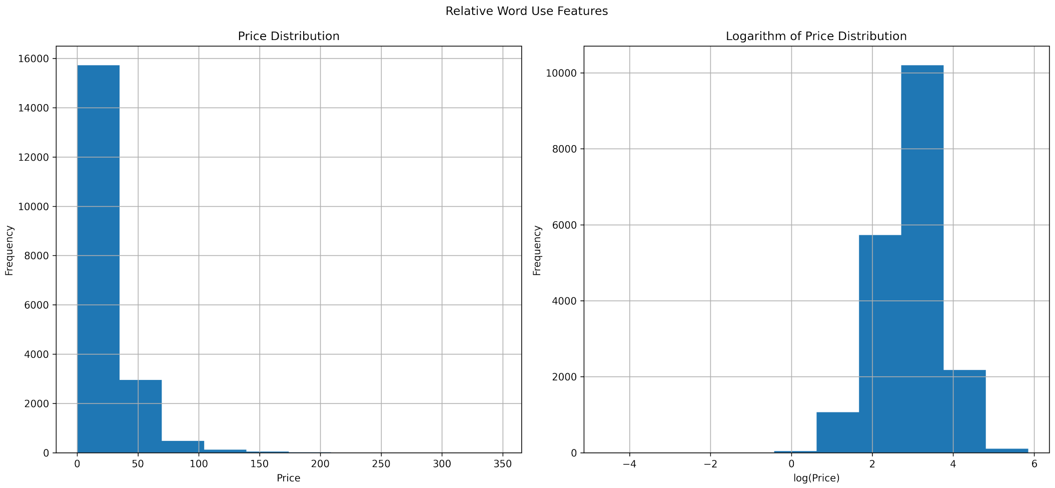


Figure 5.4: Price vs. Logarithmic Price Distribution

We can see (Figure 5.4) that after applying the logarithm to the price feature, it is much more similar to a Gaussian distribution and much less skewed. This makes it more representative and easier to process for the clustering algorithm later on.

5.5 Feature Overview

We are now left with eleven features (excluding the identifier), which can either be used for clustering or as a descriptive metric afterwards. In the table below (Table 5.3), we can see some interesting statistics.

	mean	**std**	**min**	**max**
price	24.38	20.22	0.01	348.00
overall	4.28	0.68	1.00	5.00
relativeRank	0.01	0.02	0.00	0.39
color__words	0.08	0.08	0.00	0.80
quality__words	0.17	0.10	0.00	1.00
price__words	0.19	0.11	0.00	1.00
size__words	0.25	0.13	0.00	1.00
beauty__words	0.30	0.13	0.00	1.00
relativePriceStd	0.68	0.35	0.00	2.96
colorDiversity	0.21	0.07	0.02	0.48
logPrice	2.93	0.75	-4.61	5.85

Table 5.3: Remaining Features

It is not surprising to see that our normalized features, all range between 0 and 1. What is surprising however, is to see that the *relative sales rank* seems to be very small (meaning high ranking) and the *overall ratings* seem to be very positive. The average products seem to have a pretty high sales rank compared to the maximum sales rank available for their respective category in the data set. Also, even though the overall ratings range from one to five stars, the mean ranking given is 4.28, which is quite high, especially considering the low standard deviation of 0.68. This has to be kept in mind when later interpreting the results. A more “negative” rating compared to the mean can still be a pretty positive one.

CHAPTER 6

Clustering

After exploring different approaches, we decide to use the K-Means clustering algorithm, to categorize the customers. This clustering algorithm is very sensitive to skewed data, which is why we will have to scale our data before using it. For this, we use the *standard scaler* provided by scikit-learn [17] with the default settings.

6.1 Scaling of the Data

What this scaler does is standardizing the features by making them independent from the mean and dividing all values by the standard deviation, resulting in *unit variance*. The resulting standard score z is the result of subtracting the mean μ from our data point x and dividing the output by the standard deviation σ, as shown in the formula below (Formula 6.1).

$$z = \frac{x - \mu}{\sigma} \tag{6.1}$$

Scaling the data like this is crucial when performing K-Means clustering. The reason for this is the way this algorithm determines the clusters. When starting the K-Means clustering is (usually randomly) initializes the so-called *centroids*, which represent the centers of the clusters. It then calculates the distance between the data points and the centroids, to determine, which data points belong to, which clusters. Once all points have been assigned, it adjusts the centroids, to be in the center of all these instances. This process is then repeated until there is no or almost no movement of the centroids anymore. The *Euclidean Distance* metric is used to calculate the distance between two points (i.e. data point and centroid).

$$d(x,c) = \sqrt{\sum_{i=0}^{n}(x_i - c_i)^2} \tag{6.2}$$

In the formula above (Formula 6.2) we can see how this Euclidean distance is being calculated. Here, x is the data point and c is the centroid, we want to calculate the distance to. The control variable i represents one dimension out of all the n dimensions we have. In this context the number of dimensions n is the number of features we use for clustering. We square the distance between the two points in each dimension, sum up the results and then take the square root of this sum.

It now becomes apparent why it is necessary to scale the features before starting the clustering process. If most of our features have rather small values and one feature has huge values compared to the others, this feature will automatically influence the distances between the points the most. It will be much more important than all the other features when determining the clusters. Thus, it makes sense to scale the features to unit variance, so that they all lie in a similar range.

6.2 Optimal Number of Clusters

6.2.1 Inertia Metric

With our features now being scaled, there are still some uncertainties regarding the clustering process. Clustering in general is a very fuzzy matter. Due to its unsupervised nature, it is oftentimes hard to evaluate how well it works and what the optimal settings are. In contrast to supervised learning we do not have any "correct" answers to compare our model to, in order to score it. Nevertheless, there are established ways to figure out what the ideal settings are. First and foremost we are interested in determining the optimal number of clusters that we should use. The most simple and straight-forward approach is to run the K-Means clustering on the data with different values for k (number of clusters) and to calculate a metric that indicates the quality of the results. One such metric is the *inertia* of the model. It tells us how close the data points are to their cluster centers.

$$Inertia = \sum_{i=1}^{N}(X_i - C_k)^2 \tag{6.3}$$

The formula above (Formula 6.3) shows how it is calculated. We calculate the squared distance between each individual instance X_i and the centroid C_k of the cluster k they belong to. We then sum up all the distances to get the model's inertia. In the formula, i

is the control variable and N is the total number of instances.

Our goal is to get a model that has a low inertia but also a low number of clusters. The inertia will obviously decrease with each additional cluster that we add. So we need to find a balance between keeping k low and keeping the inertia low. The best-practice way to determine the optimal number of clusters using this metric is to do it visually, by plotting the inertia for all k-values, which are considered and to find the so-called *elbow*. Since it is a visual method, the elbow can not be defined exactly in a mathematical way. It is the point at which the curve suddenly becomes less steep. In other words, it is the point at which adding more clusters does no longer decrease the inertia in a massive way.

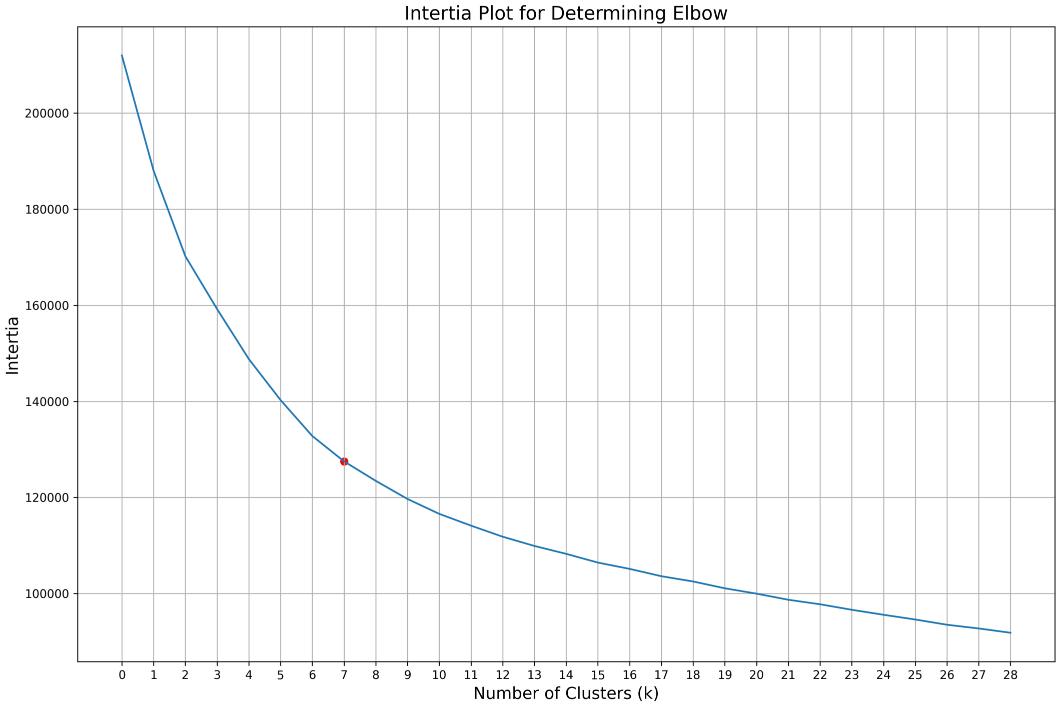


Figure 6.1: Inertia Plot

In the figure above (Figure 6.1), we can see the inertia plot for our data set, when running it through a K-Means clustering algorithm. For the sake of reproducibility it makes sense to mention that the clustering was done with a *random state* of 20. When looking at this graph, we can see that it is not too obvious where the elbow is. It seems like the curve flattens the fastest at around seven clusters, but it is also not necessarily wrong to choose six or eight clusters as the elbow point here.

6.2.2 Silhouette Score

We can see that it is not all too clear, what the optimal number of clusters is by just looking at this graph. Looking at the inertia is a decent method when the elbow is very obvious, which is not the case here. Thus, we will use a second, more sophisticated, metric to evaluate the quality of our clustering - the so-called *silhouette score*. In scikit-learn [17], the silhouette score is defined as the mean *silhouette coefficient* S of all samples. The formula down below (Formula 6.4) shows how it is calculated. Here, b is the mean distance from each element to the nearest cluster it does not belong to, whereas a is the mean distance between the individual elements of the same cluster.

$$S = \frac{b - a}{max(a, b)} \tag{6.4}$$

In theory now, the number of clusters that produces the highest silhouette score (mean silhouette coefficient), should be considered optimal. The larger the distance between the individual clusters and the closer together the instances of the same cluster at the same time, the higher the silhouette score. We can plot this metric for a couple of different k-values, to see if we get a clear result.

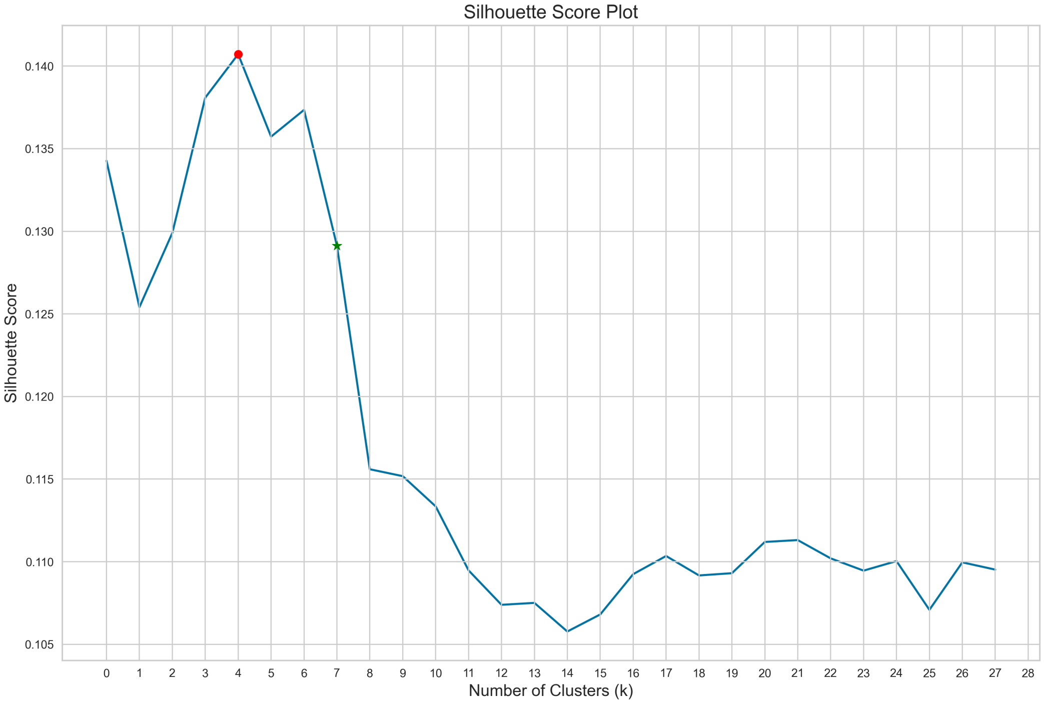


Figure 6.2: Silhouette Score Plot

If we just look at the raw silhouette score for different the k-values (Figure 6.2), it seems like four clusters (and thus customer categories) seem to be optimal. This contradicts the result that we got from the elbow analysis of the inertia plot, where we had something between six and eight as the optimal value for k. Also, we can see that there is another spike in the plot for six clusters, which is not as high as for four clusters, but would align better with the results from before. Some further analysis seems necessary.

To further examine the silhouette results, we plot more advanced silhouette plots for the k-values that we are considering. Even though seven was our initial choice after looking at the inertia plot, we mainly consider four and six as possible values for k, due to the rapid decrease we see in the silhouette score plot, when going beyond six.

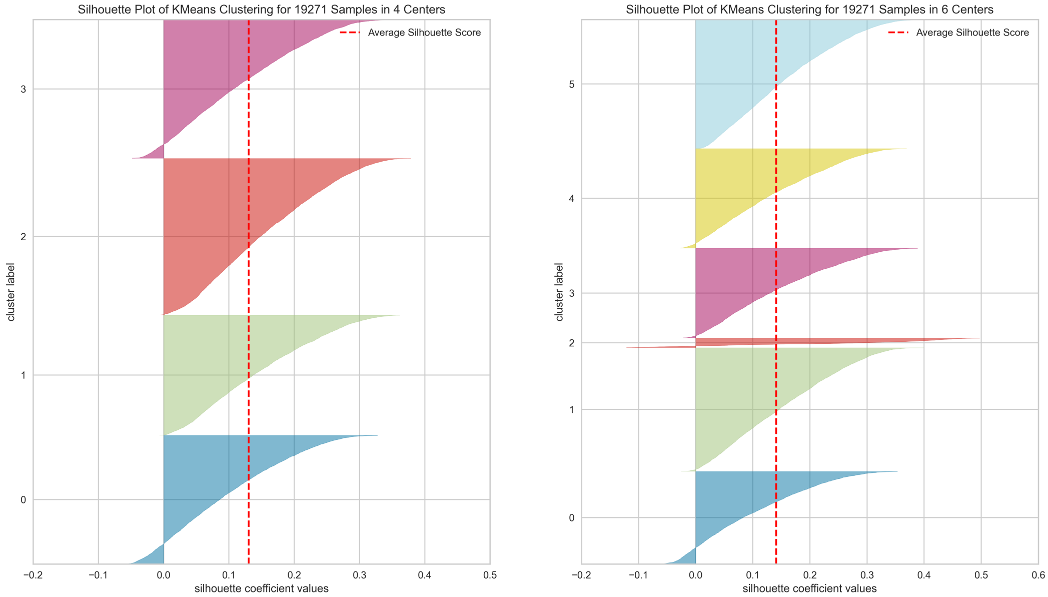


Figure 6.3: Silhouette Plot

In the graph above (Figure 6.3) we can see the structure of the clusters (four clusters left, six clusters right). Each horizontal colored line represents the silhouette coefficient value of one individual instance in a cluster. The instances of each cluster are sorted by this value in descending order. The vertical red dashed line indicates the average silhouette score.

What we like to see in a silhouette plot are clusters that have roughly the same size (number of instances) and where enough values of each cluster reach at least the average silhouette score. By using these criteria, we can see that choosing six for our k-value is problematic. Five out of six clusters look pretty good but there is one cluster (the fourth one), which is extremely small and has only a few instances with a very high score. This is not what we consider optimal in this context. Similar results can be noticed when plotting such a graph for five and seven clusters (not shown here).

So it seems like four is actually the best number for attempting to find the most optimal clusters. It is important to mention that we chose to not use any price-related features for the clustering (and thus also not for determining the optimal number of clusters), because we want to use them descriptively afterwards, to examine to what extent the resulting clusters are related to how much money customers spend.

CHAPTER 7

Results and Conclusion

In this last section, we will now examine the resulting four clusters and offer possible interpretations for them. Furthermore, we will discuss their usefulness and possible strategies that might result from them. The results where achieved by using the same clustering models that were used in the previous chapter to determine the optimal value for *k*. Keep in mind that clustering is a very fuzzy unsupervised machine learning technique and that the results might differ, when different parameters or even a different data set is used. Discussing the robustness of the clusters is also a key point of this final chapter.

7.1 Results

7.1.1 Results

Once the clustering is done, each customer receives a cluster label (i.e. a number from zero to three representing their cluster). We then one-hot encode the labels so that we end up with four individual binary features instead of one numeric feature. The reason for this is that we want to plot a correlation heat map, to visualize how the individual clusters relate to all the other features. Also, two cluster labels that are numerically closer can represent two much more different clusters than two labels that are further apart numerically. A correlation with the cluster label would thus not provide much insight.

The figure below (Figure 7.1) shows the correlation heat map, which includes our four clusters. As we can see, even though clustering is a very fuzzy unsupervised learning technique, we have pretty clear and expressive clusters that strongly correlate with the individual features.

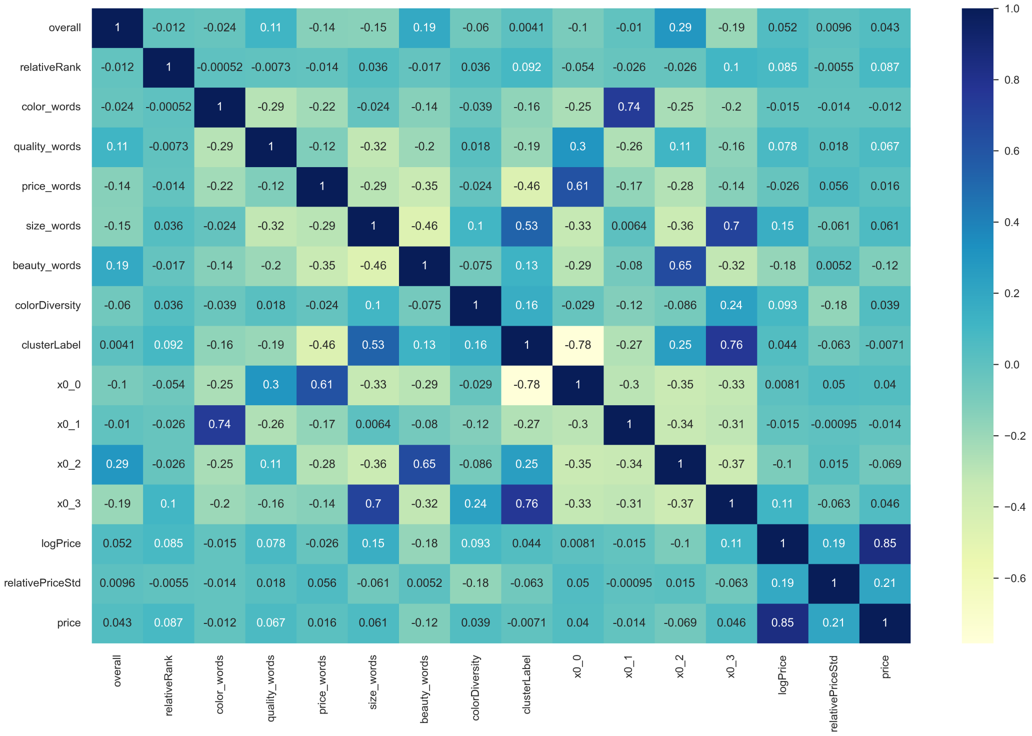


Figure 7.1: Correlation Heat Map of Clusters

7.1.2 Discussion

The *first cluster* (labelled with zero) already leads to some interesting insights. Customers that belong to this cluster seem to frequently use words related to the quality and the price of the products. At the same time, the use of words related to the color, the size or the subjective beauty of the products is quite low. The major focus of this customer category seems to be on getting the best possible quality for what they pay. Surprisingly, none of the price-related features significantly correlate (positively or negatively) with this cluster. The amount of money spent does not seem to be something that can be predicted by this cluster, even though its language is heavily focused on that exact topic. Also it seems like customers in this cluster are not really influenced by the product rank and they don't seem to be overly positive or overly negative when writing reviews.

The *second cluster* has a very strong correlation with the use of color words and a slightly negative correlation with the color diversity feature. Furthermore, it is negatively correlated with the use of quality and price words. Customers that belong to this cluster seem to put a lot of emphasis on the color of their fashion items but not necessarily on visual aspects in general (since the beauty and size word features are not correlated to this cluster). Since it is also the cluster with the lowest color diversity, we could assume that customers that belong to this cluster usually stick to a smaller set of colors when buying fashion items. This is interesting because the color words and the color diversity feature are not correlated at all (positively or negatively) and yet they are both correlated with this cluster.

The *third cluster* is characterized by a high use of beauty-related words and very positive overall ratings of the products (keep in mind that even the average reviews were already very positive). It seems like customers that belong to this cluster focus mostly on what they find subjectively beautiful instead of the price or specific colors. It could thus make sense to further examine the visual preferences of the customers belonging to this category, to improve the quality of the recommendations this group gets.

The *fourth and final cluster* has a very strong correlation with the use of size-related words, tends to give more negative ratings and has a high correlation with the color diversity feature. Furthermore it is the only cluster that has a significant correlation with a price-related feature (in this case the logarithm of the price). A possible interpretation would be that this category contains the customers that spent a relatively high amount of money on clothing that did not fit well enough and thus they were unsatisfied. Since the data is an aggregation across multiple purchases, it could be interesting to further examine the body type and other size-related features of these customers, if the data is available. It could be possible that they are not sufficiently informed about what the different product sizes mean. The Clothing Size Recommendation data set [15] discussed in the data set exploration chapter could be an interesting starting point for further analysis of this customer type.

Overall, it becomes apparent that, with one small exception, none of the price-related features have a significant correlation to the clusters. Even a frequent use of price-related words does not seem to give us any information about how much money customers spend on clothing. Also, it is interesting to see that the only visual feature (color diversity) didn't influence the clustering too much and that the relative sales rank of the products was essentially irrelevant. All of this indicates that (at least in this data set) the use of natural language in customer reviews is more important for the creation of clusters than these features and also than the overall rating given. It confirms our assumption that more emphasis should be put on the more implicit features that represent preferences, values and attitudes of the users rather than focusing only on product characteristics.

7.1.3 Evaluation of Robustness

Due to the fuzzy nature of clustering, a couple of words should be said about the robustness of these results. When changing the random state (seed) of the clustering model, the results differ in minor ways, which is to be expected. However, overall we seem to get roughly the same clusters with similar correlations. Also, before using all of the data available, the analysis has first been done on a smaller sub-sample of the data (only including customers with more than fifteen purchases) and similar results were obtained.

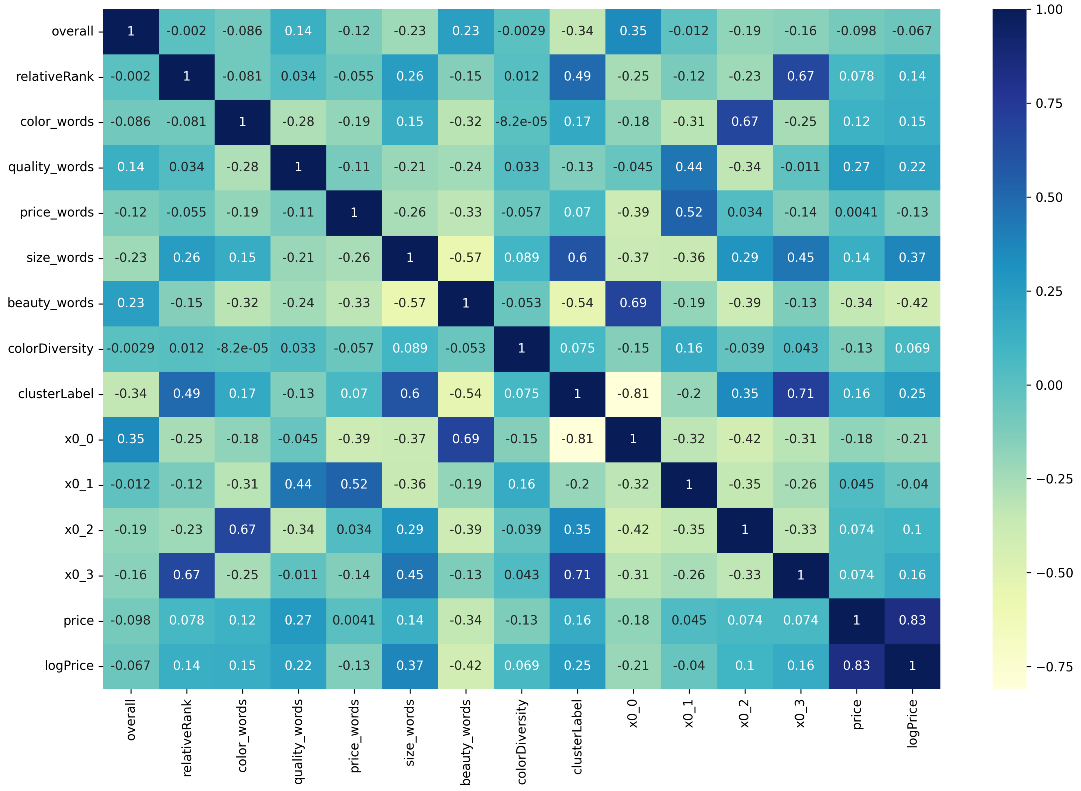


Figure 7.2: Correlation Heat Map of Clusters (Smaller Sample)

The heat map above (Figure 7.2) shows the results of this clustering. We notice that, although the order of the clusters is different, we get roughly the same clusters as with the full data set. One could even argue that this clustering is more representative, since we have 126 customers that have all made at least 15 purchases. Categorizing users based on 15 interactions seems way more reasonable than doing it based on three or four interactions.

Previously our first cluster (labelled with zero) was one where the users focused on quality and pricing in their language. Words related to color, size or beauty were less frequent in this group. In our results above, we can find this same cluster in the second position (labelled with one). The only slight difference here is that this cluster is slightly negatively correlated with the relative rank, indicating a preference for higher ranked items.

Our color-focused cluster, which was the second one before, is now our third cluster. It has a high positive correlation with the color words feature and a negative correlation with the quality words feature. However, it does not have a significant negative correlation to the price words and to the color diversity feature, which was the case before. With this smaller data set it seems like the customers in this group are talking a lot about colors but not necessarily sticking to fewer colors than is usual.

Next up, we have our beauty-related cluster, which gives ratings that are more positive than the average. Previously, this was our third cluster. In this smaller sample, it is the first one. The only difference between the two results here is that we have a negative correlation with the rank and also a negative correlation with the color diversity. These changes might be influenced by an overlap with our last cluster.

Finally, our size-related language cluster remains in the fourth position. We can see that this time there is a strong correlation with the rank, indicating purchases of lower ranking items. This could partly explain the slightly more negative reviews, even though those were present in this cluster before. One thing that is noteworthy is that our third cluster now also tends to give more negative reviews and is also positively correlated with the size words feature. This could mean that the color-focused cluster now also contains some of the size-focused customers from before.

Cluster	**Prev. Position**	**New Position**	**Difference**
Quality & Price Focus	1st	2nd	Negative Rank Correlation
Color Focus	2nd	3rd	No Price Word Correlation
Beauty Focus	3rd	1st	Negative Rank & Color Diversity Correlation
Size Focus	4th	4th	Strong Rank Correlation

Table 7.1: Clustering Results Differences

Previously, the only cluster that had a significant correlation with a price feature was the size-focused one. It is still the one with the strongest positive correlation but we can also see that the color-focused cluster is now correlated with the logarithm of the price as well. This once again indicates that the size and color clusters are now somewhat mixed together here. Interestingly, we can also see that our beauty-focused cluster is now negatively correlated with the price, indicating cheaper purchases on average.

Above (Table 7.1), we can see the slight differences in the two results. All in all, they are very similar. Because of this, it can be assumed that the resulting clusters represent patterns of customer behavior, which can be used to create more accurate and more sophisticated recommendation systems in the fashion field. To further examine the robustness, it would make sense to compare the results to cluster based on a completely different data set, which has the same structure.

7.2 Conclusion

7.2.1 Contribution

In this exploratory analysis we demonstrated that engineering features based on the natural language used in customer reviews allows for identifying clear and meaningful clusters. Furthermore, we showed that these features were more important for the cluster creation than other features like the overall ratings given, the diversity in the colors chosen when buying products and the sales rank of the products. However, the clusters could not be used to describe the spending behavior of customers. There was almost no correlation between the clusters and the average price paid by the customers in these clusters.

We can now look at the two research questions we proposed at the beginning and see if our work confirms our expected results:

- **RQ1:** To what extent can the natural language used by customers, combined with a few other basic features lead to somewhat robust and significant categorization of users?
- **RQ2:** How important are the features based on the use of natural language compared to the other features that are also used for this categorization?

ad. RQ1: We engineered five features based on the natural language used by Amazon customers in their reviews. Those features were used for clustering in combination with other basic features like the average sales rank of bought products or the overall rating given. The results were clear clusters that showed a robustness across different samples and configurations. This confirms our expectations that the natural language of customers can be used to categorize them in a significant way.

ad. RQ2: When looking at our resulting clusters, we can see that they correlate strongly with our features, which are based on natural language use. In fact, the other features used for clustering are significantly less relevant in the results. This confirms our expectations that the language used by the customers influence the clusters more than basic features like the average sales rank or the average ratings given.

7.2.2 Future Work

Further analysis can be done on the clusters described in this work. It can be useful to compare the results of different clustering methods as well as to rotate the feature selection. For example, the price features could be included into the clustering process, while other features could be used descriptively. Also, additional features could be introduced to see if the results change and whether the language-related features remain the most important ones or not. Lastly, it would be interesting to use another similar data set and run a similar cluster analysis on it, in order to compare the results. If the resulting clusters are similar to the ones described in this work, this could point to a general pattern in fashion customer behavior.

List of Figures

List of Tables

Bibliography

[1] Aghamohseni, A., and Ramezanian, R. An efficient hybrid approach based on k-means and generalized fashion algorithms for cluster analysis. In *2015 AI Robotics (IRANOPEN)* (2015), pp. 1–7.

[2] Cassell, A., Muñoz, A., Blain-Castelli, B., Irwin, N. J., Yan, F., Dascalu, S. M., and Harris, F. C. Cars: A containerized amazon recommender system. *Advances in Intelligent Systems and Computing* (2021).

[3] Chen, W., Huang, P., Xu, J., Guo, X., Guo, C., Sun, F., Li, C., Pfadler, A., Zhao, H., and Zhao, B. POG: personalized outfit generation for fashion recommendation at alibaba ifashion. *CoRR abs/1905.01866* (2019).

[4] Ge, Y., Zhang, R., Wu, L., Wang, X., Tang, X., and Luo, P. A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. *CVPR* (2019).

[5] Han, X., Wu, Z., Jiang, Y.-G., and Davis, L. S. Learning fashion compatibility with bidirectional lstms. In *ACM Multimedia* (2017).

[6] Honnibal, M., and Montani, I. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[7] Hsiao, W., and Grauman, K. Dressing for diverse body shapes. *CoRR abs/1912.06697* (2019).

[8] Jaradat, S., Dokoohaki, N., Pampín, H. J. C., and Shirvany, R. Fashion recommender systems. In *Recommender Systems Handbook*. Springer, 2022, pp. 1015–1055.

[9] Kang, W., Kim, E., Leskovec, J., Rosenberg, C., and McAuley, J. J. Complete the look: Scene-based complementary product recommendation. *CoRR abs/1812.01748* (2018).

[10] Kang, W.-C., Fang, C., Wang, Z., and McAuley, J. Visually-aware fashion recommendation and design with generative image models. In *2017 IEEE International Conference on Data Mining (ICDM)* (2017), pp. 207–216.

[11] Kiapour, M. H., Han, X., Lazebnik, S., Berg, A. C., and Berg, T. L. Where to buy it: Matching street clothing photos in online shops. In *2015 IEEE International Conference on Computer Vision (ICCV)* (2015), pp. 3343–3351.

[12] Landia, N., Mcalister, R., North, D., Kalloori, S., Srivastava, A., and Ferwerda, B. Recsys challenge 2022 dataset: Dressipi 1m fashion sessions. pp. 1–3.

[13] Liu, Z., Luo, P., Qiu, S., Wang, X., and Tang, X. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016).

[14] McAuley, J. J., Targett, C., Shi, Q., and van den Hengel, A. Image-based recommendations on styles and substitutes. *CoRR abs/1506.04757* (2015).

[15] Misra, R., Wan, M., and McAuley, J. Decomposing fit semantics for product size recommendation in metric spaces. In *Proceedings of the 12th ACM Conference on Recommender Systems* (2018), ACM, pp. 422–426.

[16] Monte, A., Soares, C., Brito, P., and Byvoet, M. Clustering for decision support in the fashion industry: A case study. In *Advances in Sustainable and Competitive Manufacturing Systems* (Heidelberg, 2013), A. Azevedo, Ed., Springer International Publishing, pp. 997–1008.

[17] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[18] Sevegnani, K., Seshadri, A., Wang, T., Beniwal, A., McAuley, J., Lu, A., and Medioni, G. Contrastive learning for interactive recommendation in fashion. In *SIGIR 2022 Workshop on eCommerce* (2022).

[19] Shirvany, R., and Corona Pampin, H. J. Fourth workshop on recommender systems in fashion and retail – fashionxrecsys2022. In *Proceedings of the 16th ACM Conference on Recommender Systems* (New York, NY, USA, 2022), RecSys '22, Association for Computing Machinery, p. 680–683.

[20] Vincent, O., MAKINDE, A., Salako, O., and Oluwafemi, O. A self-adaptive k-means classifier for business incentive in a fashion design environment. *Applied Computing and Informatics 14* (05 2017).

[21] Yan, C., Malhi, U. S., Huang, Y., and Tao, R. Unsupervised deep clustering for fashion images. In *Knowledge Management in Organizations* (Cham, 2019), L. Uden, I.-H. Ting, and J. M. Corchado, Eds., Springer International Publishing, pp. 85–96.

[22] Yinyin, W. Consumer behavior characteristics in fast fashion.