

E-Commerce: High Potential Regions

Finding regions within Germany that have a high population density and a low retail store density by utilizing Web Scraping, Geocoding and the Foursquare API

Project Report

July 25, 2021

by Kevin Götz



Table of Contents

1. Introduction	2
1.1 Background	2
1.2 Problem.....	2
1.3 Objective & Relevance	2
2. Methodology.....	3
2.1 Data Sources & Collection.....	3
2.3 Exploring & Cleaning the Data	5
2.3.1 Zipcode Locations.....	5
2.3.2 Foursquare API.....	7
2.4 Feature Engineering	7
2.6 Data Visualization.....	8
3. Results	9
3.1 Venues & Categories.....	9
3.2 Top 20 high Potential Locations.....	10
3.3 Distribution of the KPIs	11
3.4 Potential by City, Region and Germany	12
4. Discussion.....	14
4.1 Discussion of Results.....	14
4.2 Discussion of Methodology.....	14
5. Conclusion & Outlook	16
Appendix	17

1. Introduction

1.1 Background

Amid slowing economic activity, COVID-19 has led to a surge in e-commerce and accelerated digital transformation. As closures became the new normal, consumers and businesses have become increasingly "digital," offering and buying more goods and services online, increasing e-commerce's share of global retail from 14% in 2019 to about 17% in 2020. These and other findings are reflected in a new report, "COVID-19 and E-Commerce: A Global Review", from UNCTAD and eTrade for all partners, which reflects the industry's strong global and regional changes in 2020. (source: <https://unctad.org/news/how-covid-19-triggered-digital-and-e-commerce-turning-point>, 19.07.2021)

The trend toward e-commerce is likely to continue during the recovery from COVID-19, but the reopening of physical stores will probably have a slowing effect on the growth of E-Commerce companies.

1.2 Problem

Some changes in consumers' shopping habits may be longer-lasting, but there may be several push and pull factors for consumers to leave their house again for an in-person shopping experience offline. After more than one year living in a lockdown the consumers probably are looking forward to experiencing the normal everyday life again and not sit in their flats in front of their screens (push factors). Also, the retail shops are probably offering widespread sales campaigns to attract potential customers and drive revenue again to counteract the lost year of commerce (pull factor).

With the lifting of the restrictions in Germany the e-commerce companies no longer have the "monopoly" on shopping and the competition gets more balanced again. The e-commerce industry needs to find strategies to keep up the momentum and not lose their newly gained customers.

1.3 Objective & Relevance

A possible lever for customer retention and customer acquisition could be a geographical analysis of the retail landscape and population. Assuming that consumers with sparse shopping options in their area are more likely to order online, this project aims to find regions within Germany that have a high population density as well as a low retail density. If an e-commerce company is focusing their marketing strategies towards those regions it may uplift their ROI.

2. Methodology

2.1 Data Sources & Collection

The data origins from four different sources in total, which are explained in chronological order below:

1. Web scraping Zipcode Information:

The basis of the analysis are locations which are identified by zipcode, city, quarter, district and region. The data was collected from the website "<https://home.meinestadt.de>" because it has the most detailed and clean zipcodes for Germany compared to "https://en.wikipedia.org/wiki/List_of_postal_codes_in_Germany" (not detailed enough) or "<https://worldpostalcode.com/germany/>" (not clean enough).

The scraping consists of two steps:

a. Scraping the URLs for the regional pages of Germany:

<https://home.meinestadt.de/deutschland/postleitzahlen> (22.07.2021)

Postleitzahlen nach Bundesland	
> Baden Württemberg	> Niedersachsen
> Bayern	> Nordrhein-Westfalen
> Berlin	> Rheinland-Pfalz
> Brandenburg	> Saarland
> Bremen	> Sachsen
> Hamburg	> Sachsen-Anhalt
> Hessen	> Schleswig-Holstein
> Mecklenburg-Vorpommern	> Thüringen

b. Scraping the regional pages for zipcode data:

<https://home.meinestadt.de/bayern/postleitzahlen> (22.07.2021)

Postleitzahlen in Bayern				
PLZ	Stadt	Stadtteil	Landkreis	Bundesland
63739	Aschaffenburg	Aschaffenburg-Österreicher Kolonie		Bayern
63741	Aschaffenburg	Aschaffenburg-Strietwald		Bayern
63743	Aschaffenburg	Aschaffenburg-Schweinheim		Bayern
63755	Alzenau		Kreis Aschaffenburg	Bayern
63762	Großostheim		Kreis Aschaffenburg	Bayern
63768	Hösbach		Kreis Aschaffenburg	Bayern
63773	Goldbach/Unterfranken		Kreis Aschaffenburg	Bayern
63776	Mömbris		Kreis Aschaffenburg	Bayern
63785	Obernburg a. Main		Kreis Miltenberg	Bayern
63791	Karlstein a. Main		Kreis Aschaffenburg	Bayern

Special care was applied when scraping the "meinestadt"-Website. The basic requests from the HTTP-Client (Requests- or Urllib-Library) were blocked from the server (HTTP Error 403: meaning access to the requested resource is forbidden. The server understood the request but will not fulfill it.) because they were missing the usual request-header containing an user-agent information. This user-agent information is usually available when browsing in person and a request without it was probably detected as a crawler. Using the header ("User-Agent": "Mozilla/5.0 (Windows NT 6.1)") and an uniformly randomized time delay (sleep function with `numpy.random.uniform`) solved the problem. Final data collected:

2. Scraping Wikipedia:

Wikipedia was used to gain insights about the population and area of the location. In a first step a search string consisting of the zipcode and the quarter name (or city name, if quarter was not applicable, e.g. for small villages) was build. This search string was then concatenated with the Wikipedia search URL having the settings on "Sort by most relevant" which got the results while testing:

<https://de.wikipedia.org/w/index.php?search=63928+Walldürn&title=Spezial%3ASuche&go=Artikel&ns0=1>

(22.07.2021)

Suchergebnisse

Suchen

Erweiterte Suche: Sortieren nach Relevanz

Suchen in: (Artikel)

Der Artikel „63928 Walldürn“ existiert in der deutschsprachigen Wikipedia nicht. Du kannst den Artikel erst
Wenn dir die folgenden Suchergebnisse nicht weiterhelfen, wende dich bitte an die Auskunft oder suche n

Walldürn

Walldürn ist eine Stadt im Neckar-Odenwald-Kreis in Baden-Württemberg, bekannt durch die Wallfahrt zum Blutwunder von **Walldürn**. Sie gehört zur europäischen

25 KB (2.616 Wörter) - 13:53, 10. Jul. 2021



Heppdiel

verläuft die Landesgrenze zu Baden-Württemberg, gleichzeitig Ortsgrenze zu **Walldürn**. Höhenlinienbild auf dem BayernAtlas der Bayerischen Staatsregierung (Hinweise)

2 KB (93 Wörter) - 20:12, 19. Jul. 2021

In a second step the result page was crawled for the first entry's URL (here: Walldürn) and this URL was then requested to get to the Wikipedia page of the quarter, city or village (depending on the data).

Finally the table in the right upper corner (first table on the page) was parsed for area ("Fläche:") and population ("Einwohner:") information before repeating this process for all zipcodes within Germany (N > 7.600, <https://de.wikipedia.org/wiki/Walldürn>, 22.07.2021):

Wappen	Deutschlandkarte
	
Basisdaten	
Bundesland:	Baden-Württemberg
Regierungsbezirk:	Karlsruhe
Landkreis:	Neckar-Odenwald-Kreis
Höhe:	416 m ü. NHN
Fläche:	105,88 km ²
Einwohner:	11.601 (31. Dez. 2020) ^[1]
Bevölkerungsdichte:	110 Einwohner je km ²
Postleitzahlen:	74731, 63928
Vorwahlen:	06282, 06285, 06286
Kfz-Kennzeichen:	MOS, BCH
Gemeindeschlüssel:	08 2 25 109
Adresse der Stadtverwaltung:	Burgstraße 3 74731 Walldürn
Website:	www.wallduern.de
Bürgermeister:	Markus Günther (CDU)

Special care was applied when scraping Wikipedia. Because of regulations to quantity and speed of crawling ("<https://de.wikipedia.org/robots.txt>") a randomized time delay (sleep function with `numpy.random.uniform`) was applied again.

3. **Geocoding of Latitude and Longitude Coordinates (OpenStreetMap API via Geocoder Library):**

The distance to retail shops can only be computed (Step 4) when the location data is enriched with coordinates. The library "Geocoder" and an OpenStreetMap API were used to obtain the latitude, longitude and the bounding boxes.

4. **Adding the Retail Information (Foursquare API):**

As a last step for the data collection the Foursquare Developer API ("Personal" account) was used to map Fashion Retail Shops with their distance to the location data. The Foursquare API therefore uses the Latitudes and Longitudes from the previous step and returns all the stores' names, categories, sub-categories, latitudes and longitudes and distance to the requested location in a user-defined circumference. The circumference (or radius = r) was computed for every location while looping through coordinates for API-requests, applying the formula " $A = \pi r^2$ " where A is known from the crawled Wikipedia pages and π (pi) is a constant: $r = \text{np.sqrt}(A / \pi)$.

The rate limits on the userless API calls per hour (5000 calls max) led to a sleep-function of 1 seconds before every API call to not exceed the rate limit.

The final result was saved as a separate feather-file containing the API results which was joined with the location data on the `wiki_url` (unique to each website) later in the process. See Appendix for final data as a table impression.

2.3 Exploring & Cleaning the Data

The exploration and cleaning was basically split into two parts: First the Web Scraping and Geocoding of the zipcode locations and second the API calls for the surrounding venues in those locations. The first part was by far more time consuming and is therefore emphasized.

2.3.1 Zipcode Locations

Cleaning the data was the most time consuming part (right after the data collection) during this project. Especially the web scraping was error prone because for every zipcode/city entry from the first web scraping, two Wikipedia pages had to be scraped for the information on area and population: The first page is the search result page from Wikipedia, the second page is the details page of the city/quarter/village from Wikipedia (details in 2.2.1 Data Sources & Collection). This equals to nearly 15.000 Wikipedia pages crawled and they were not always built equally (HTML-wise).

Only the rows with clean information on area, population and coordinates can be used for our analysis and therefore the messy and missing data were dropped before appending the store data from Foursquare and analyzing our results. The reasons for those missings were versatile. After inspecting some of them by hand (using the scraped `wiki_url`) there were 4 main reasons:

- The details page had no table with area and population (e.g. `"/wiki/Leinatal"`)
- The details page had a table but is built differently regarding the HTML so it couldn't be parsed (e.g. `"/wiki/Anrode"`)
- The coordinates couldn't be fetched with the Geocoder library despite population and area available on Wikipedia (e.g. `"/wiki/Gefell"`)
- The search string for Wikipedia didn't bring the correct result on top so the wrong page was scraped (e.g. `"/wiki/2.Tennis-Bundesliga(Herren)"` as a result of searching "Mannheim-Neckerau")

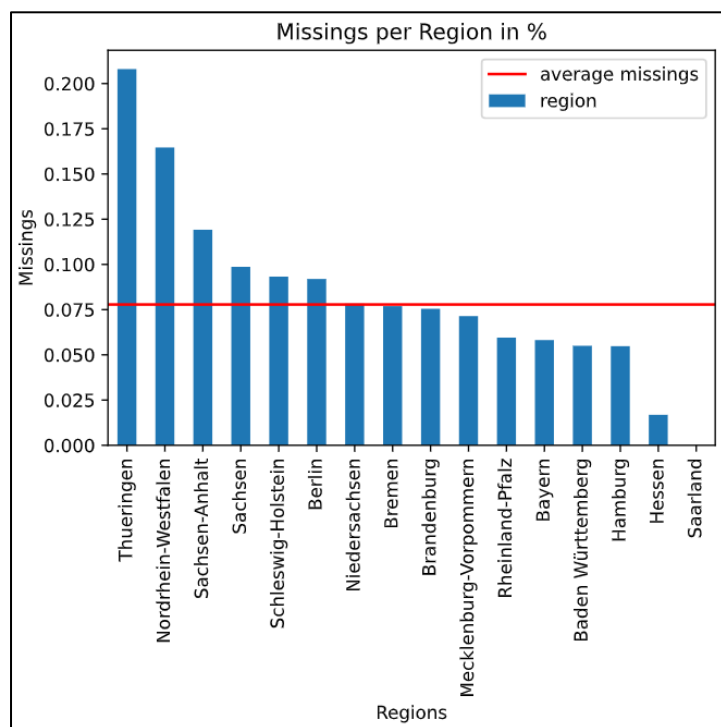
Therefore a "rescraping" and cleaning of the missing or wrong information was adapted. Some cities have NaN values for population or area because there were two values in the table cell on the Wikipedia page. The table

parsing had to be changed to a regex. This was especially useful since the naming of the table rows was different across URLs: “Fläche” vs “Fläche:”.

Some cities were recognized as "Landkreis" (category “above” a city, e.g. county) by the Wikipedia search algorithm and had the first position in the result set. The second result had to be scraped instead.

For the rest of the data the population and area had to be cleaned and relevant information had to be extracted from a population string like this: '22.385 (Stand: 31. Dez. 2015)[1]' with the first number being the population and part in between the parentheses being the date of the valid timestamp of the population data.

The 16 regions in Germany had different data quality with Saarland being on top as cleanest and Thueringen being the dirtiest data (because of slightly different HTML layouts and information across regions):



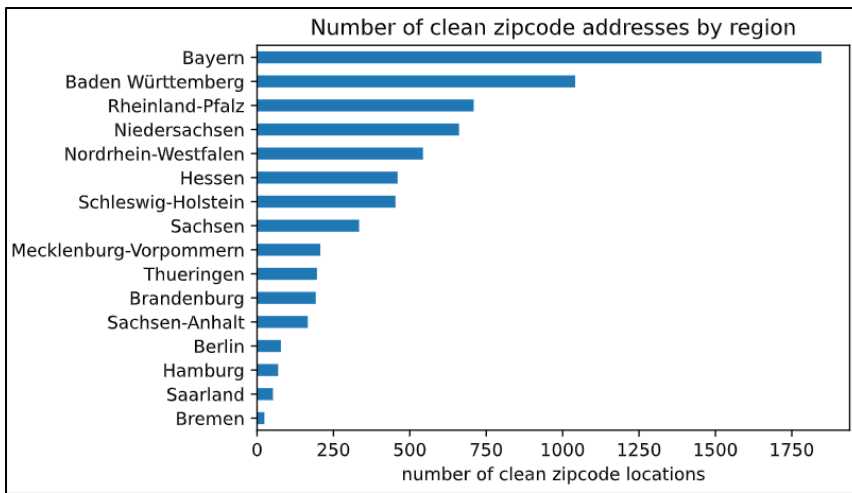
In the end there were 601 out of 7721 (= 8%) zipcode locations in total that were not evaluable due to missings in one of the following three: area, population or coordinates.

There were also a lot of outliers for the area size of the scraped locations. The most suitable way to check for outliers is using the area of the location in km² because we can assume that the zipcode size should be normally distributed around a population mean and the sizes of the regions should not differ too much (otherwise the area gets split into multiple zipcodes, probably). Therefore, we use the Z-Scores and standard deviations (SD). Since we only check for areas bigger than usual and we want to be 99% sure, we pick the one-sided Z-Score for that confidence from the normal distribution: 2.33. So all plausible values fall within the mean + 2.33 * SD. As a result, the highest plausible value for an area is 881 km². There were 64 location deleted that were bigger than this threshold.

After the first cleaning the remaining locations were geocoded to add the coordinates for the later API use and the visualization. The same search string that was used earlier for Wikipedia was also used as the input for the geocoding with OpenStreetMap via the Geocoder library. Even though there was an error handling included, which in case tried to catch the coordinates with the ArcGIS system, there were still some locations that couldn't be found and they were deleted as well.

Three variables had to be available for every location to proceed with the analysis: area, population and coordinates. All locations that had a missing in one of those were discarded. The cleaned table resulted in 7030 entries compared to the old table with 7721 entries. 9% of the entries were dropped due to data cleaning. The cleaned data distribution of the regions is as following:

E-Commerce: High Potential Regions



2.3.2 Foursquare API

The Foursquare API offers an option to narrow down the type of venue you are searching for by applying a category ID in your URI. The details are described here: <https://developer.foursquare.com/docs/build-with-foursquare/categories/>. There are top level categories that combine multiple venue types that can be grouped together. In the case of this project the category "Clothing Store" (ID: 4bf58dd8d48988d103951735) was used to only find the relevant venues.

The data had to be cleaned in three ways:

1. **Duplicates:**

Even though a venue should be unique with its coordinates there were 20 duplicate venues in total that were dropped. The first entry for every location was kept.

2. **Wrong Categories:**

Even though only the Clothing Store category and its corresponding subcategories were chosen, there were still some categories that had nothing to do with clothes. E.g. 'Construction & Landscaping', 'Electronics Store' or 'Doctor's Office'. Those categories and their venues were deleted and only venues from the following categories were kept: 'Clothing Store' (as mentioned) + 'Outlet Store', 'Miscellaneous Shop', 'Leather Goods Store', 'Jewelry Store', 'Factory', 'Bridal Shop', 'Adult Boutique', 'Arts & Crafts Store'.

3. **Wrong Venues:**

Some categories that were fashion and clothes oriented had venues that surely weren't. Therefore all venues with the following string within their name (regex contains) were deleted as well: 'gerät', 'technik', 'raiffeisen', 'maschine'.

2.4 Feature Engineering

The goal of this project was to get the locations within Germany that have a high potential for E-Commerce sales and marketing activities. So first "potential" was defined as the following: A location (city/village/quarter with unique zipcode) with a high population density and a low retail store density.

Consequently, three features had to be computed from the existing data:

- **population density:**

A float computed from the Wikipedia info on population and area. Population divided by area is the population density.

- **retail density:**

A float computed from the Foursquare info on the shop count within the predefined radius divided by the Wikipedia info on the locations area. The complement of retail density (**retail sparsity** = 1 – retail density) is later used to compute the potential correctly.

- **potential:**

The potential is a combination of the previous two variables and was computed as following:

$$1/2 * (\text{normalized}(\mathbf{population\ density}) + (\text{normalized}(\mathbf{retail\ sparsity}))$$

Potential is the final variable that is evaluated for every neighborhood. It is normalized (0-1 range) so every neighborhood has a kind of percentile rank ranging from 0-1 for comprehensibility.

2.6 Data Visualization

The final visualization helps to condense the data into easily comprehensible insights on where to take action as an E-Commerce company in Germany. The high potential regions were those with a potential score of 0.7 or higher.

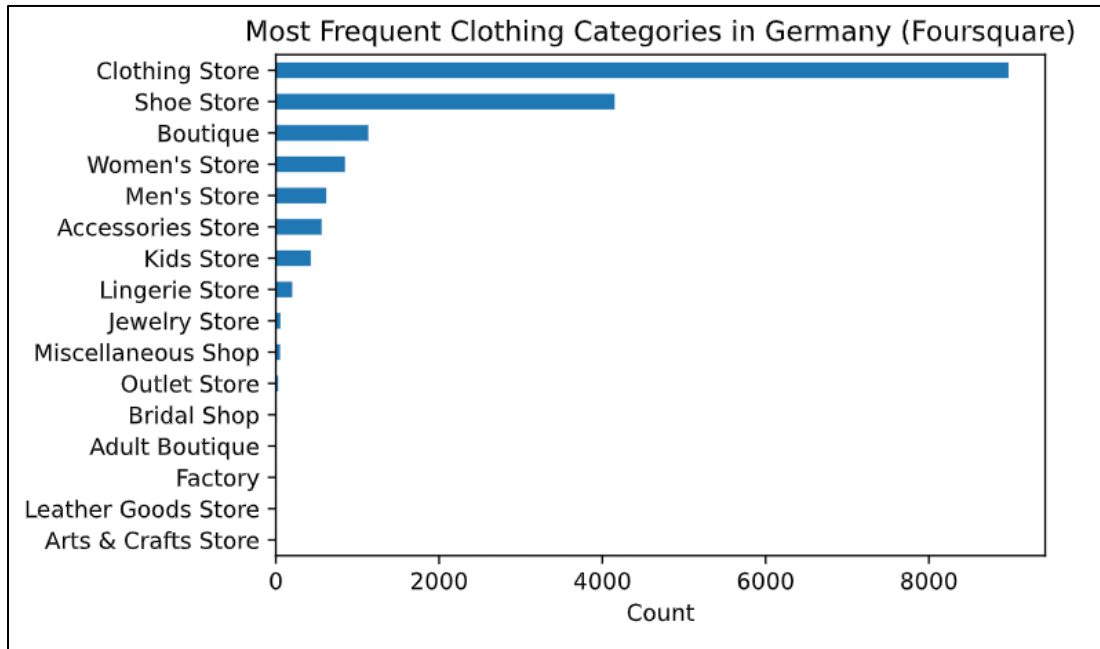
The first part is a styled table containing the top 20 locations with the most important data (zipcode, quarter, city, region, wiki_url and the three KPIs mentioned in the chapter “Feature Engineering”) at one glimpse.

The second and most important part is an interactive Folium HeatMap showing the clusters of high potential locations with detailed information within the marker text. See results for more details and download the Notebook for interactive exploring.

3. Results

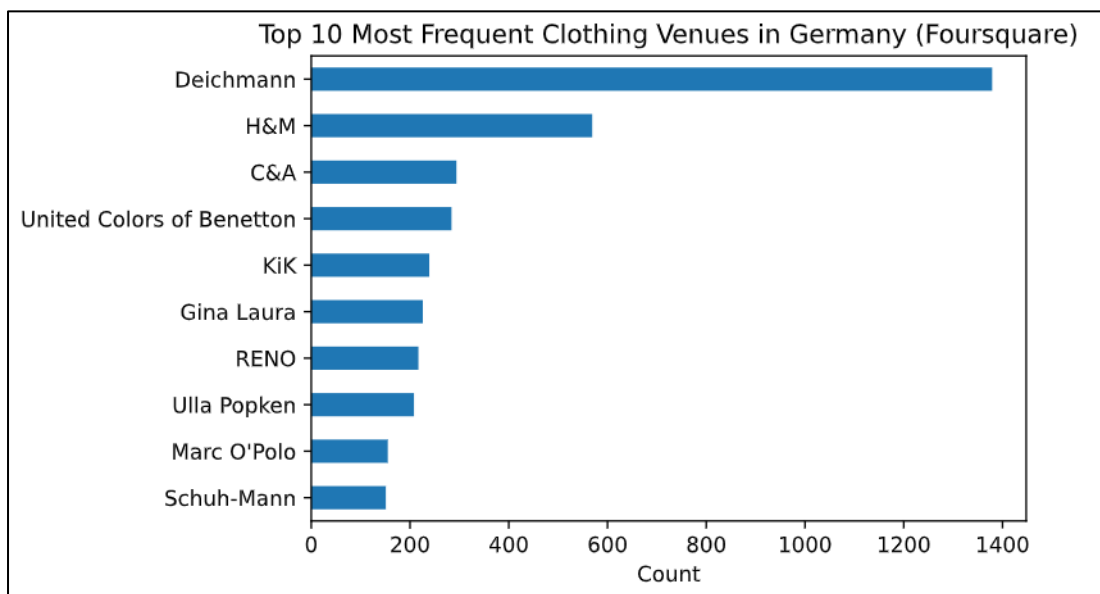
3.1 Venues & Categories

The results do not only cover the potential of the regions, but also the exploratory parts of the analysis beforehand. Since we are counting venues from Foursquare within Germany it is important to get an impression of what kind of clothing stores the analysis deals with. The venue categories are ranked by counted venues in the following graph:



Within the top level category “Clothing Store” (which are technically all of the listed categories above) the biggest category by far is “Shoe Store”, followed by small boutiques and women’s stores.

The dominance of shoe stores is mainly driven by Deichmann, that leads the most frequent clothing venues in Germany (Foursquare). Place number seven “RENO” and number 10 “Schuh-Mann” also sell shoes and contribute to the result above. Most of the brands are placed rather in the budget domain than the luxury domain:



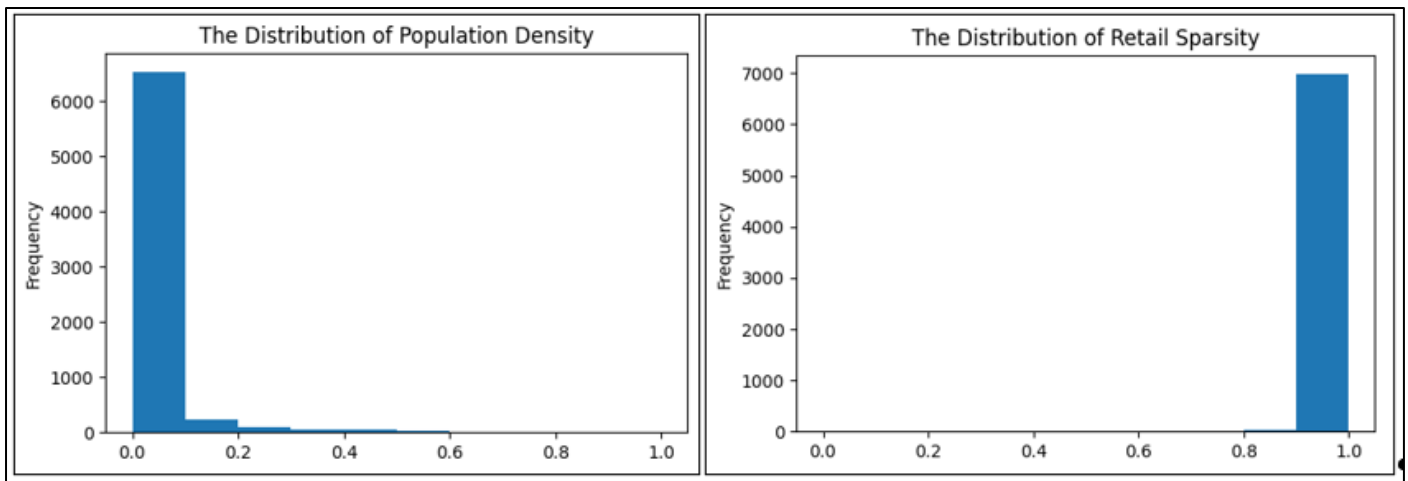
3.2 Top 20 high Potential Locations

Within the 20 most potent locations for E-Commerce are mostly the main cities from the regions Berlin, Hamburg and Bayern. They make up 80% of the Top 20 ranking and the Top 10 has a potential from 0.91 and above. The dominant driver is a very high retail sparsity matched with a medium to high population density in most of the cases. The table below shows geographical details and the three KPIs of the zipcode locations and also includes the web scraped wiki_url for further information about the city and its quarter:

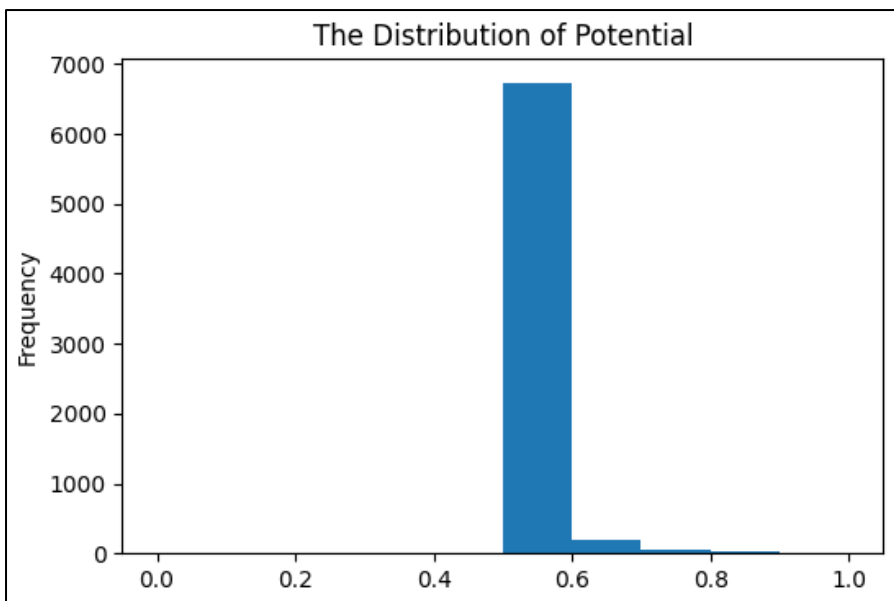
	zipcode	city	quarter	region	learn_more	population_density	retail_sparsity	potential
0	20253	Hamburg	Hamburg-Hoheluft-West	Hamburg	/wiki/Hamburg-Hoheluft-West	1.00	0.86	1.00
1	10369	Berlin	Berlin-Fennpfuhl	Berlin	/wiki/Berlin-Fennpfuhl	0.83	0.99	0.98
2	70182	Stuttgart	Stuttgart-Heusteigviertel	Baden Württemberg	/wiki/Heusteigviertel	0.94	0.87	0.97
3	13357	Berlin	Berlin-Gesundbrunnen	Berlin	/wiki/Berlin-Gesundbrunnen	0.80	0.97	0.95
4	20259	Hamburg	Hamburg-Eimsbüttel	Hamburg	/wiki/Hamburg-Eimsbüttel	0.94	0.83	0.95
5	22049	Hamburg	Hamburg-Dulsberg	Hamburg	/wiki/Hamburg-Dulsberg	0.75	0.99	0.93
6	12043	Berlin	Berlin-Neukölln	Berlin	/wiki/Berlin-Neukölln	0.73	0.99	0.93
7	12353	Berlin	Berlin-Gropiusstadt	Berlin	/wiki/Berlin-Gropiusstadt	0.74	0.98	0.92
8	10969	Berlin	Berlin-Kreuzberg	Berlin	/wiki/Berlin-Kreuzberg	0.77	0.95	0.92
9	81675	München	München-Haidhausen	Bayern	/wiki/Au-Haidhausen	0.76	0.94	0.91
10	81541	München	München-Au	Bayern	/wiki/Au-Haidhausen	0.76	0.94	0.91
11	10405	Berlin	Berlin-Prenzlauer Berg	Berlin	/wiki/Berlin-Prenzlauer_Berg	0.78	0.90	0.90
12	79100	Freiburg im Breisgau	Freiburg im Breisgau-Vauban	Baden Württemberg	/wiki/Vauban_(Freiburg_im_Breisgau)	0.71	0.97	0.90
13	50677	Köln	Köln-Neustadt/Süd	Nordrhein-Westfalen	/wiki/Neustadt-Süd_(Köln)	0.71	0.97	0.89
14	22089	Hamburg	Hamburg-Eilbek	Hamburg	/wiki/Hamburg-Eilbek	0.68	0.98	0.89
15	13439	Berlin	Berlin-Märkisches Viertel	Berlin	/wiki/Berlin-Märkisches_Viertel	0.65	0.99	0.88
16	80339	München	München-Schwanthalerhöhe	Bayern	/wiki/Schwanthalerhöhe	0.75	0.89	0.87
17	10243	Berlin	Berlin-Friedrichshain	Berlin	/wiki/Berlin-Friedrichshain	0.73	0.91	0.87
18	04315	Leipzig	Leipzig-Volkmarisdorf	Sachsen	/wiki/Volkmarisdorf	0.63	1.00	0.87
19	10711	Berlin	Berlin-Halensee	Berlin	/wiki/Berlin-Halensee	0.64	0.98	0.86

3.3 Distribution of the KPIs

The two KPIs “population density” and “retail sparsity” are the foundation for the superior KPI “potential”. They all follow a differently skewed distribution schema and finally balance each other out. The distribution of the Population Density is strongly skewed to the right with most of the locations having only 20% of the maximum population density and below. On the other hand, the retail sparsity is even stronger skewed (to the left) with almost all locations having a sparsity of 90% and above:

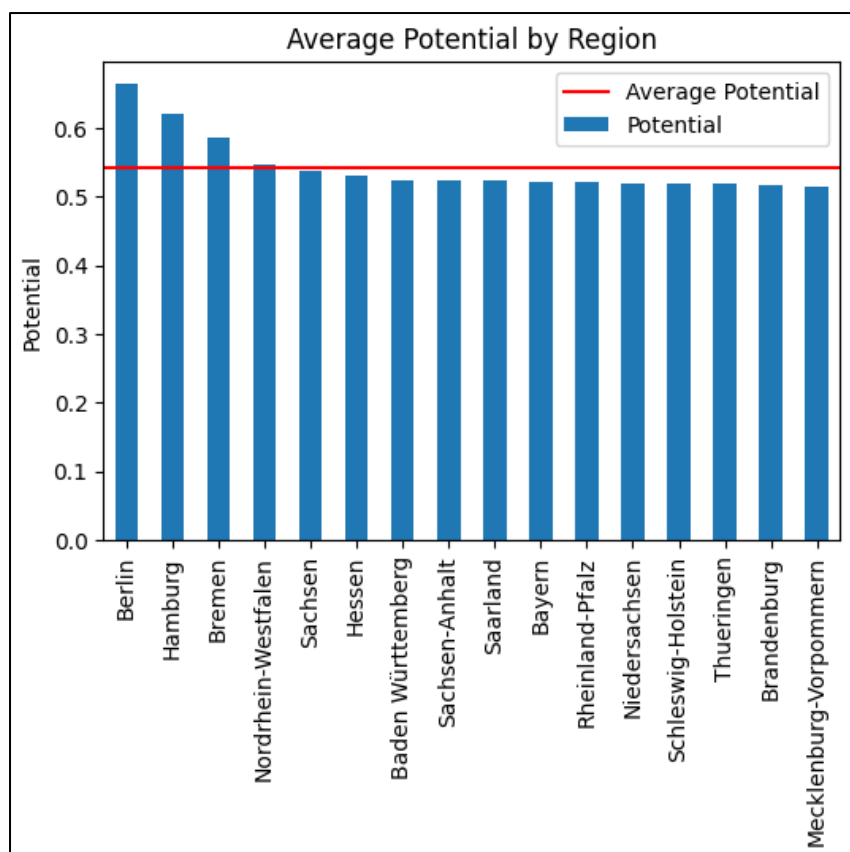


Since those two KPIs contribute evenly to the final KPI “Potential”, they balance each other out and leave most of the locations at a value of about 50%-60% of the maximum potential:

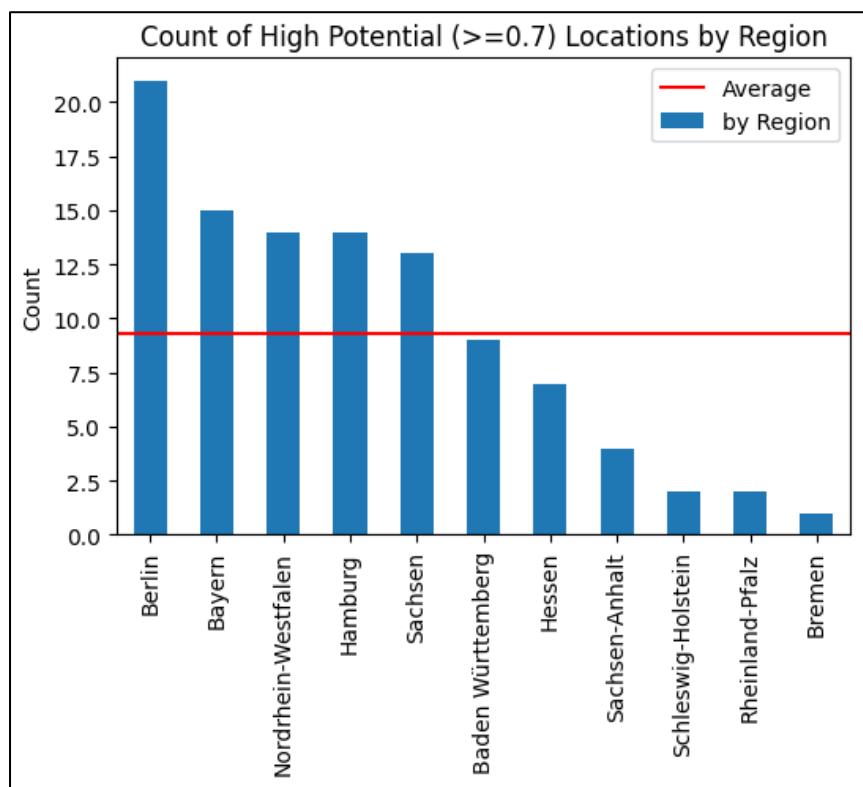


3.4 Potential by City, Region and Germany

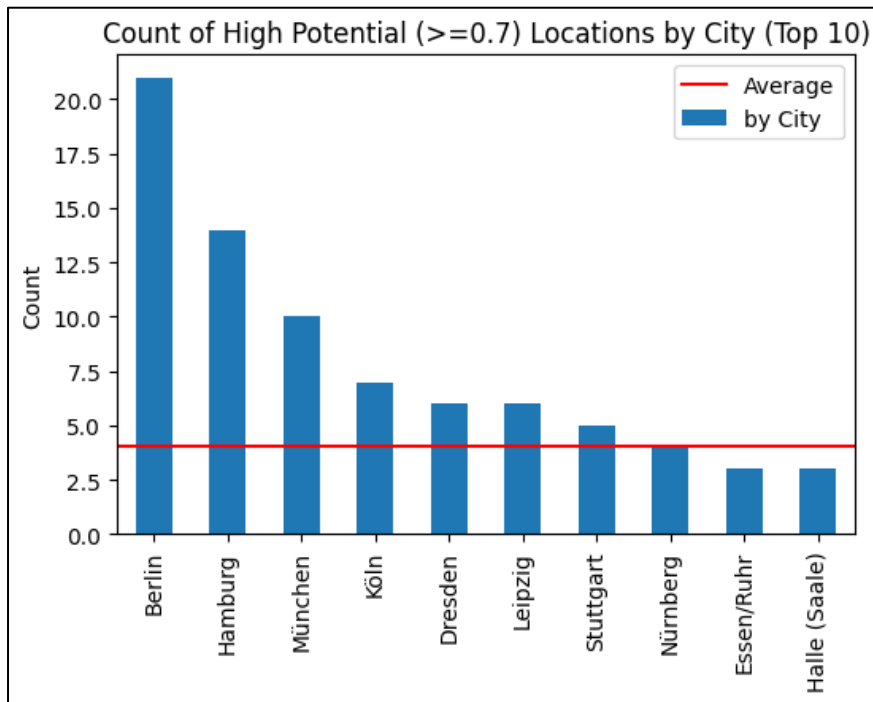
Even though the top contenders mainly reside within 3 regions, the regions overall are quite balanced in their potential. Three of the 16 German regions are on top with a bigger margin than the rest:



The differences become more obvious when looking at the count of high potential (≥ 0.7) locations within the regions, where Berlin has the definitive lead over all the other regions. Bremen is on last place in this ranking:

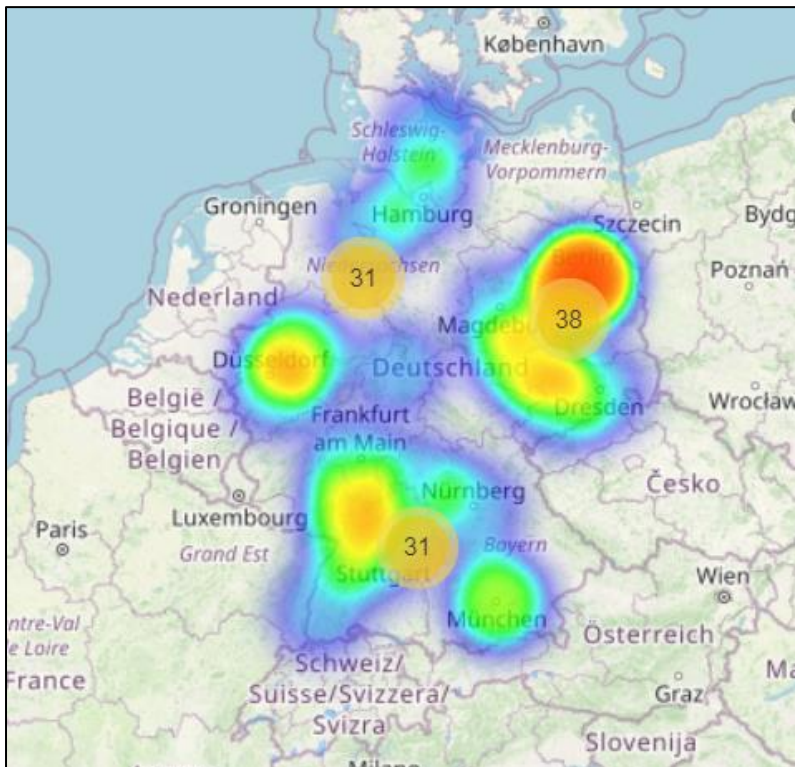


The average potential by city is quite balanced too and therefore only the count of high potential (≥ 0.7) locations is shown here. Berlin again has the lead with over 20 high potential regions, followed by Hamburg and Munich:



The high potential regions are distributed around 3 clusters geographically:

- **South (31%):** The south extends from Freiburg and Munich up to Nuremberg and Frankfurt with Munich (10), Stuttgart (5) and Nuremberg (4) in the lead.
- **East (38%):** The east extends from Dresden and Halle (Saale) up to Berlin. Berlin leads the ranking with 21 high potential areas, followed by Dresden (6) and Leipzig (6).
- **West & North (31%):** The West & North extends from Cologne and Kassel up to Kiel. Hamburg (14) has the most high potential locations, followed by Cologne (7).



4. Discussion

4.1 Discussion of Results

The average potential for 13 of the total 16 regions in Germany is balanced around 0.5 Potential, which means that those regions have 50% of the maximum Potential on average. Only three of the regions seem to not follow this trend: Berlin, Hamburg and Bremen. The Commonality of those regions is that they are the only City-States in Germany. City States are administrative divisions that only cover cities and they are characterized by the fact that they are comparably small and densely populated because they are mainly one big city.

This finding is seen in the other regions as well, where the cities dominate the high potential locations and the rural areas are characterized by low potential. This is quite surprising because the potential for E-Commerce was expected to be in the more rural areas where the inhabitants do not want to make a long drive to the next city to go for clothes shopping and rather order online. Even though the potential is calculated without weights and population density and retail sparsity are included equally, it seems that the population density has the upper hand. This can make sense when it comes to OOH marketing campaigns, where the total impressions and interactions (e.g. QR Code) are a driver of ROI and the amount of people walking by those ads is therefore crucial.

Looking at the Top 20 most potential locations, a different picture emerges though. One could think that in those densely populated areas there has to be a high retail density. Looking at the numbers, it occurs to be the other way around: Most of the leading locations have a high retail sparsity of above 90% of the maximum sparsity within the dataset. This has mainly two reasons:

First, the histograms of the KPIs show a clearly skewed distribution. While almost all the locations have a high retail sparsity (above 0.9 and nearly none below 0.7), the population density is more balanced, with most locations having 0.2 or lower, but the big cities having 0.8 and above. Therefore, cities with a high population density separate themselves from the rest, while retail sparsity has a much smaller range and is much more dense around 0.9, which leaves less space for improvement. The reason for the high average retail sparsity has methodological reasons and is discussed in 4.2.

The second reason is best explained with an example: In the Top 20 list of high potential locations the neighborhood "Berlin-Gropiusstadt" is on the 8th place with a high retail sparsity of 0.98. This is irritating because of the huge shopping mall "Gropius Passagen" that is within this neighborhood, but not within the dataset. This can occur a) when the API doesn't list this mall or its stores, b) when the search radius was not large enough so it doesn't cover the mall from the geocoded latitude/longitude position, or c) when the predefined venue categories don't match the category of this specific mall. Since "Shopping Mall" is not listed within "Clothing Stores" as a venue category (see: <https://developer.foursquare.com/docs/build-with-foursquare/categories/>), option c) is highly possible.

4.2 Discussion of Methodology

The major point of discussion when it comes to data collection and data quality is the scraping of Wikipedia. If it was a "static" approach, like the scraping of the zipcodes, where the websites that have to be scraped are known, it would have been more controllable. But with this approach, every location had to be searched by the Wikipedia algorithm, triggered by the request that looped through the zipcode locations. The first hindrance was to get to the details page via the top result of the triggered search page. Even though there were several attempts tested by hand and the best method (zipcode plus name of the quarter as a search string) was chosen accordingly, there were some cases where the wrong data was scraped.

An example is the city "Düsseldorf" with its different quarters. Even though the search string was best practice, the top result on Wikipedia wasn't the city's quarter, but the city itself and the wrong area and population was scraped. This can be seen here:

search_string	wiki_title	wiki_url	wiki_area_sqkm	wiki_population	latitude	longitude
40210 Düsseldorf-Stadtmitte	Düsseldorf	/wiki/D%C3%BCsseldorf	217.41	620523.0	51.221939	6.784423
40213 Düsseldorf-Carlstadt	Düsseldorf	/wiki/D%C3%BCsseldorf	217.41	620523.0	51.222142	6.773394
40629 Düsseldorf-Ludenberg	Düsseldorf	/wiki/D%C3%BCsseldorf	217.41	620523.0	51.256357	6.866151

E-Commerce: High Potential Regions

Even though those three are all different quarters and the details website should be different, the Wikipedia URL is the same and so are area and population. An important notice is, that the latitude and longitude, that were geocoded in the following step, are correct and different for the three quarters. That means, that those three quarters have the same area and therefore search radius for Foursquare, but they also have different venues because the starting points (coordinates) are different.

Even when the correct Wikipedia page was scraped, the underlying HTML was not always built the same so the scraping didn't work or sometimes the Wikipedia page lacked the population info, so the location couldn't be used because no density KPIs could be computed.

Because of the versatility of this "dynamic" approach and the high number of missings and false data, it would be more appropriate to use an API or any other more reliable approach for data collection. Future scrapings of Wikipedia could also be programmed more sophisticated maybe use google as a primary search engine.

Another point of argument is the extremely high density and skewness of the retail sparsity histogram. The KPI is concentrated around a mean of 0.99 from a maximum of 1, which brings up questions on the usefulness of this metric. Retail sparsity is the invers of retail density (normalized), which itself is a measure for venues per square kilometer. The picture below illustrates the root cause. There are only two locations with a normalized retail density of 0.7 and above and they are so dense in venues, that every other location is retail sparse compared to this.

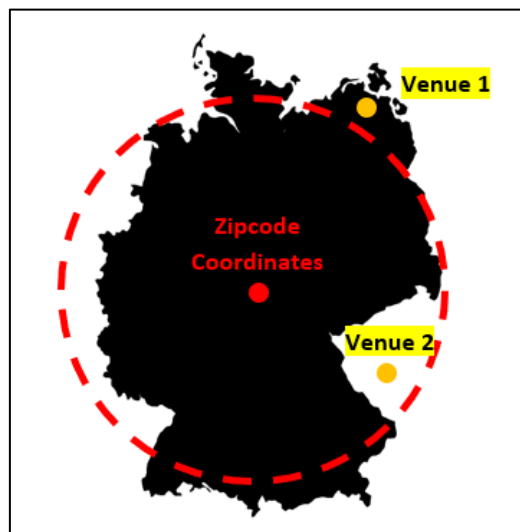
Through normalization those extreme values were set to 1 (retail density) and every other value is a kind of percentage of that maximum, due to the formula of the normalization:

	wiki_title	wiki_area_sqkm	venue_count	retail_density_per_sqkm	retail_density_per_sqkm_norm
3182	Hamburg-Altstadt	1.2	85	70.833331	1.000000
3192	Hamburg-Sternschanze	0.6	39	64.999997	0.917647

A solution could be the use of a standardization instead of a normalization, so the outliers are not impacting the rest of the data and the KPIs retail sparsity and population density can still be added together for a final standardized score of potential. The initial approach was an normalization because it is easier to comprehend and interpret scores ranging from 0-1 compared to interpreting Z-scores. Also a mixture of first standardizing density and sparsity, then normalizing the potential, could be reasonable.

The use of Foursquare was due to convenience of a free of charge KPI with nearly 100,000 API calls a day. There are other providers like Tripadvisor or Google Maps that could be applicable and give slightly different results. A quality check of the Foursquare data is nearly impossible without using other APIs and processing the same data in a repeatable approach, which was out of scope for this project.

Last but not least was the radius to search for surrounding venues computed using the area that was scraped from Wikipedia. There problem is that the zipcode locations are not perfect circles, of course. The radius is therefore a best guess with the same area size as the location, but not the same form. Due to that fact there are probably some missing and some wrong venues attributed to an area. A sample drawing was added for illustration. Venue 1 is not within the radius but within the location and Venue 2 is within the radius but not within the location:



5. Conclusion & Outlook

This project tried to find regions within Germany that offer a high potential for E-Commerce companies and their marketing and sales efforts. The potential was defined by the population density and the retail sparsity, that were computed by using data from a zipcode locations website, Wikipedia and the Foursquare API.

Even though the dynamic scraping approach with Wikipedia was lacking some precision and therefore data quality / data veracity, the cleaned results can be viewed as the most up to date database on zipcode locations with area and population. The last national census in Germany was in 2011 and the data from the official government statistics website is therefore outdated. There was no other source than Wikipedia with such a vast and up to date knowledge on small villages, cities and their quarters.

The results show clearly that there is regional variance in potential and where E-Commerce companies can concentrate their activities to open up those areas. The results (like the Top 20 table) can be checked manually with the Wikipedia URL to obtain more and deeper insights about the prospect location and discuss possibilities. The most potent regions are summarized in an interactive Folium Heatmap that can help to get a first impression of the German landscape and the geographics, plus some basic information on the locations when clicking on the markers.

The results should be taken with a grain of salt, though. Especially the computation of potential and the normalization / standardization has room for improvement, as mentioned in the methodology discussion. Future research could try different approaches that are less skewed regarding retail density / sparsity to get a more realistic picture of the locations' potential.

Future research could also use different venue category codes in the Foursquare API URI compared to just using "Clothing Stores" and its sub-categories. The problem with shopping malls, as mentioned in the results discussion. Those shopping malls could be a big difference in assessing the potential of a location, because they can host dozens of shops and therefore are a big contender aiming for fashion customers in this region.

Last but not least the attribution of venues to neighborhoods could be improved. Using a custom radius that is dependent on the area size is the best proxy so far, but maybe future Data Scientists find more appropriate ways like using the bounding boxes and a more sophisticated approach of geocoding.

The biggest possibility is the national census 2022 in Germany. This data will include area, population by age and gender and zipcode addresses that are the most accurate and up to date. Especially the demographic splits for the local population data are a huge opportunity for every retail companies, that can then tailor their definition of potential so sex and age and get a more accurate picture of the high potential regions.

Appendix

2.1.1 Web scraping Zipcode Information (final data table impression):

	zipcode	city	quarter	district	region	search_string
7621	98743	Gräfenthal	None	Kreis Saalfeld- Rudolstadt	Thueringen	98743 Gräfenthal
7622	98744	Meura	None	Kreis Saalfeld- Rudolstadt	Thueringen	98744 Meura
7623	98744, 98746	Schwarzatal	None	Kreis Saalfeld- Rudolstadt	Thueringen	98744 Schwarzatal
7624	98746	Katzhütte	None	Kreis Saalfeld- Rudolstadt	Thueringen	98746 Katzhütte
7625	99084	Erfurt	Erfurt- Altstadt	None	Thueringen	99084 Erfurt- Altstadt
...

* search_string was computed using the first zipcode and the quarter name if applicable, else using the first zipcode and the city name

2.1.2 Web scraping Wikipedia (final data table impression):

wiki_title	wiki_url	wiki_area_sqkm	wiki_population	wiki_population_date
Walldürn	/wiki/Walld%C3%BCrn	105.879997	11601.0	31. Dez. 2020
Eberbach	/wiki/Eberbach	81.169998	14267.0	31. Dez. 2020
Innenstadt/Jungbusch	/wiki/Innenstadt/Jungbusch	4.550000	31286.0	31. Dez. 2019
Neuostheim/Neuhermsheim	/wiki/Neuostheim/Neuhermsheim	5.240000	7387.0	31. Dez. 2015
Schwetzingenstadt/Oststadt	/wiki/Schwetzingenstadt/Oststadt	4.410000	22385.0	31. Dez. 2015
...

* the wiki_title & wiki_url was scraped from the search results page (top entry). First entry's URL is correct, that's how Wikipedia parses non ASCII strings apparently

** the wiki_area_sqkm, wiki_population & wiki_population_date was scraped from the detail wiki_url-page and is cleaned in this picture

2.1.3 Geocoding with OpenStreetMap (final data table impression):

	search_string	latitude	longitude	boundingbox
0	63928 Walldürn	49.6375005	9.3110464	49.6374505, 49.6375505, 9.3109964, 9.3110964
1	64754 Eberbach	49.56522905	9.07583865434271	49.40522905, 49.72522905, 8.9158386543427, 9.2...
2	68159 Mannheim-Mühlau	49.4948819	8.4514877	49.4848819, 49.5048819, 8.4414877, 8.4614877
3	68163 Mannheim-Neuostheim	49.4788199	8.5049695	49.4787699, 49.4788699, 8.5049195, 8.5050195
4	68165 Mannheim-Schwetzingenstadt	49.480345391387736	8.482308669466056	49.320345391388, 49.640345391388, 8.3223086694...
...

* search_string was used to search for the locations with the Geocoder Library

2.1.4 Venue Data with Foursquare API (final data table impression):

	zipcode_index	wiki_url	venue_name	venue_category	venue_subcategory	search_radius	venue_distance	venue_latitude	venue_longitude
0	0	/wiki/Walld%C3%BCrn	KUHN Maßkonfektion - Schneeberg	Clothing Store	Apparel	5805	4696	49.637920	9.245910
1	2	/wiki/Innenstadt/Jungbusch	Zara	Clothing Store	Apparel	1203	975	49.486893	8.445949
2	2	/wiki/Innenstadt/Jungbusch	Hollister Co.	Clothing Store	Apparel	1203	1054	49.485769	8.447503
3	2	/wiki/Innenstadt/Jungbusch	H&M	Clothing Store	Apparel	1203	1102	49.485317	8.447513
4	2	/wiki/Innenstadt/Jungbusch	Lemis Modehaus	Clothing Store	Apparel	1203	441	49.492000	8.455678
...

* search_radius and venue_distance are in the unit meters

** zipcode_index is the index of the zipcode table. If API requests fail and script stops one can easily track to the last request call and start from there again