Wydział Matematyki
i Nauk Informacyjnych
POLITECHNIKA WARSZAWSKA

Google Places Clustering and Recommendation system

Mateusz Iwaniuk, Hubert Kowalski

Agenda

01 Project objectives and data sources

02 Data Exploration

03 Reengineering meaningful features

04 Testing different clustering algorithms and tuning them

05 Interpretation of the results

Introduction

Google Maps recommendations:

Not many people know about it, but Google Maps prepares a list of recommended places based out favourite venues. Users can easily see location, ratings and photos of recommended places.

Number of businesses:

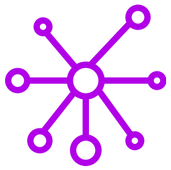
There are more than 200 million businesses and places registered in Google Maps as of 2023. Unfortunately, for the purpose of this project, we only used 15 thousand places.

Popularity:

In 2019 Google Maps was utilized by 67% of all smartphone users, indicating that a significant majority regularly depend on its services.

Objectives

Use unsupervised machine learning methods to uncover **hidden patterns** in places on Google Maps, using information about their **business category**, reviews, **location** and more.



Form and **visualise clusters** of objects that have similar features, with a focus on their location and business type.

Build a **reliable recommendation system** for Google Maps based on acquired clusters of businesses, that adjusts to individual preferences of users.

General project info

The project was created as part of the Introduction to Machine Learning course at Warsaw University of Technology. Its aim was to gain experience with **unsupervised machine learning** methods.

This project was based on data related to places on Google Maps. Its goal was to find hidden patterns in types of google places and build a recommendation system for them.

Data Sources

The dataset used in our project has been meticulously scraped from Google Maps and presents extensive information about businesses across several countries.

Each entry in the dataset provides detailed insights into business operations, location specifics, customer interactions, and much more, making it an invaluable resource for data analysts and scientists looking to explore business trends, geographic data analysis, or consumer behaviour patterns.

phone_number	name	full_address	latitude	longitude	review_count	rating	timezone
Unknown	حصن الظفرة Al Dhafra Fort	PXMJ+H7J حصن الظفرة Al Dhafra Fort - Unnamed R...	23.733973	53.980629	58.0	4.7	Asia/Dubai
971569049905	Rafiullah	MP4R+WQJ Rafiullah - Zayed City - Abu Dhabi - ...	23.657332	53.741881	3.0	3.0	Asia/Dubai

Exploratory Data Analysis

Our original dataset before preprocessing consists of:

- **46 columns** describing business features
- **15 203 rows** - places

It contains a wide range of data types spread across numerous columns. Each column was meticulously curated to ensure ethical integrity.

The most relevant features can be summarised into the following categories:

- **location (geographical coordinates, address, country and city)**
- **rating**
- **review count**
- **business type**
- **whether the business is verified**
- **opening hours**
- **additional info** (website, IDs, phone numbers)

Exploratory Data Analysis

Data types that appear in the dataset are the following:

- **Integers (21):** columns describing opening hours, which are in fact True/False columns because they contain only 0s and 1s
- **Real numbers (5):** latitude, longitude, review_count, rating and geo_cluster
- **True/False values (1):** verified - column indicating whether the place is verified
- **Texts (19):** most of the descriptions, addresses, id's

We meticulously handled the division of data to guarantee there are no **missing values (NAs)**. The rows that contained missing values were erased from the dataset (there were only **5** such rows). This provided a robust and fair **analytical foundation** for our modeling process.

Exploratory Data Analysis

The first look into the dataset resulted in some general conclusions about the data.

Numeric columns:

- It seems that most of the places are located in one region
- Values of **longitude** oscillate about 55, and **latitude** about 25, with small standard deviations. This corresponds to **United Arab Emirates.**
- Number of reviews ranges from only 1 to 256877
- The most popular **rating** for any google place is **4.4**

Fun Fact: the location that had **256877 reviews** in our dataset was the **Dubai Mall**. It is the 16th most reviewed place in the world. The first one is **Masjid al-Haram**

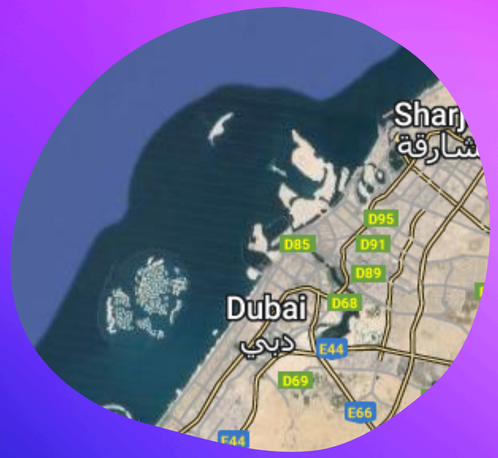
Exploratory Data Analysis

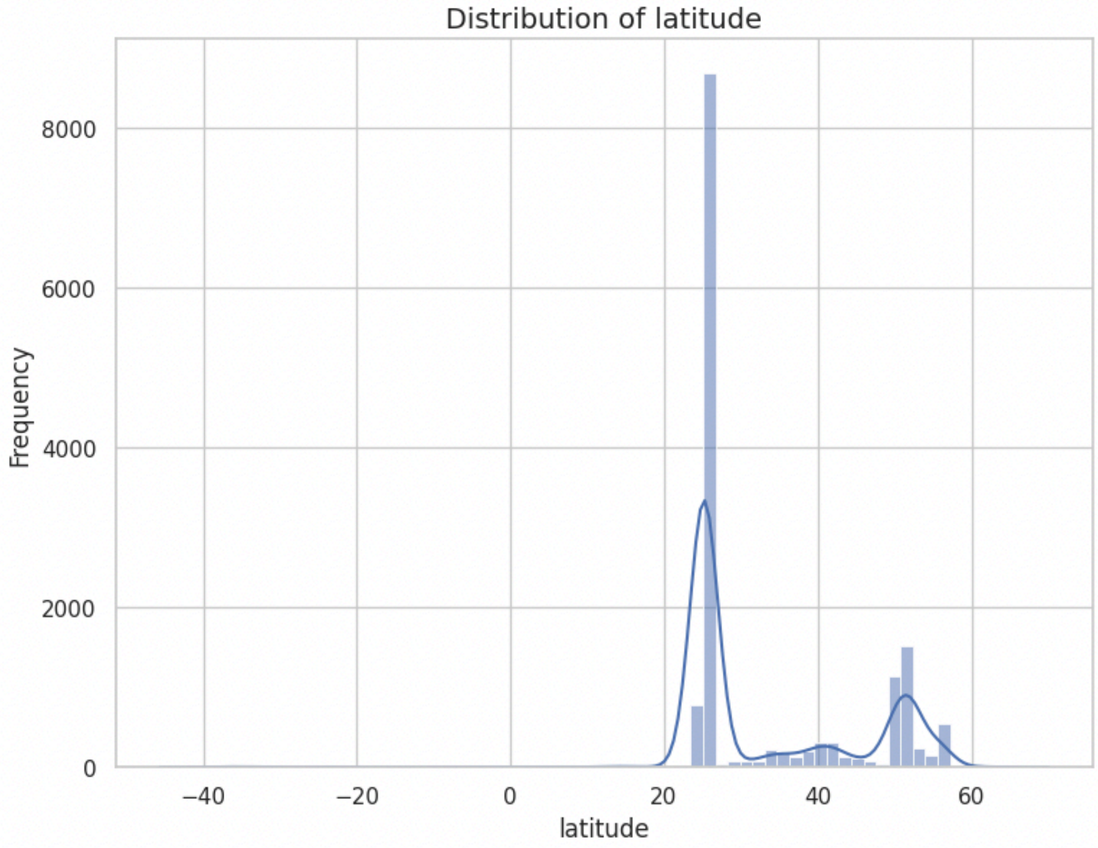
The first look into the dataset resulted in some general conclusions about the data.
Categorical (text) columns:

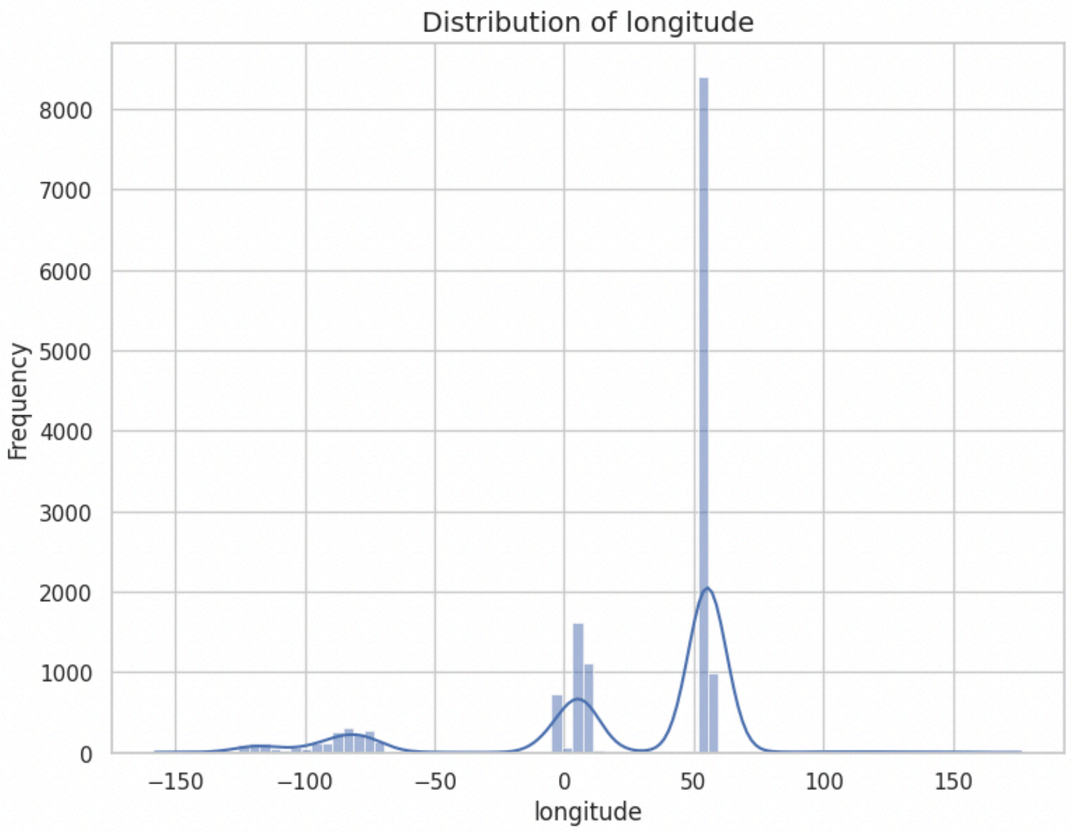
- Names of places on Google Maps are not unique, for example there are multiple locations labelled as **Geldmaat** - ATMs in the **Netherlands.** Places can be identified by their IDs.
- There are 112 unique values for **timezone** in the dataset, while there only exist 24. This suggest that this column is not a valuable information source.
- The most popular type of business is **Hotel**
- Most of the places are located in **Dubai, United Arab Emirates (64%)**
-

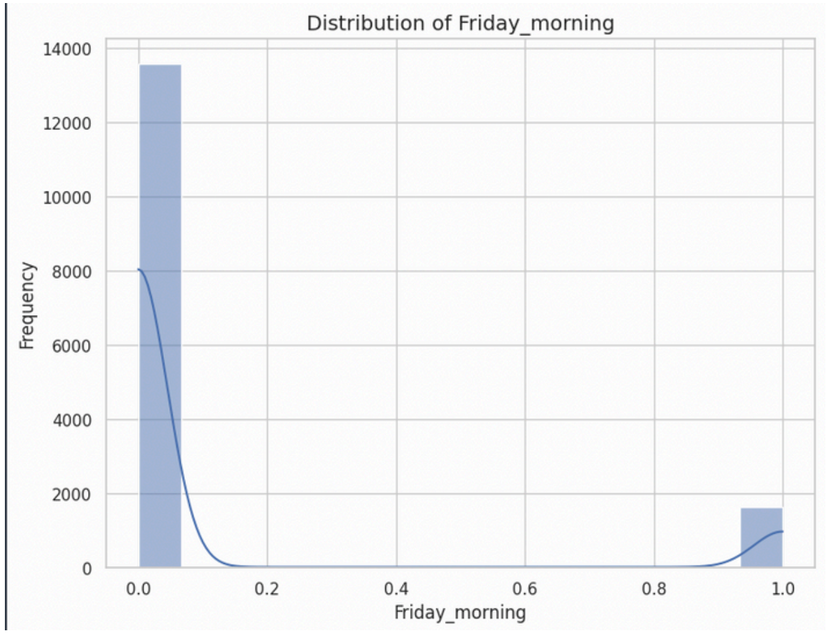
The next slide presents the distributions of the most important features.

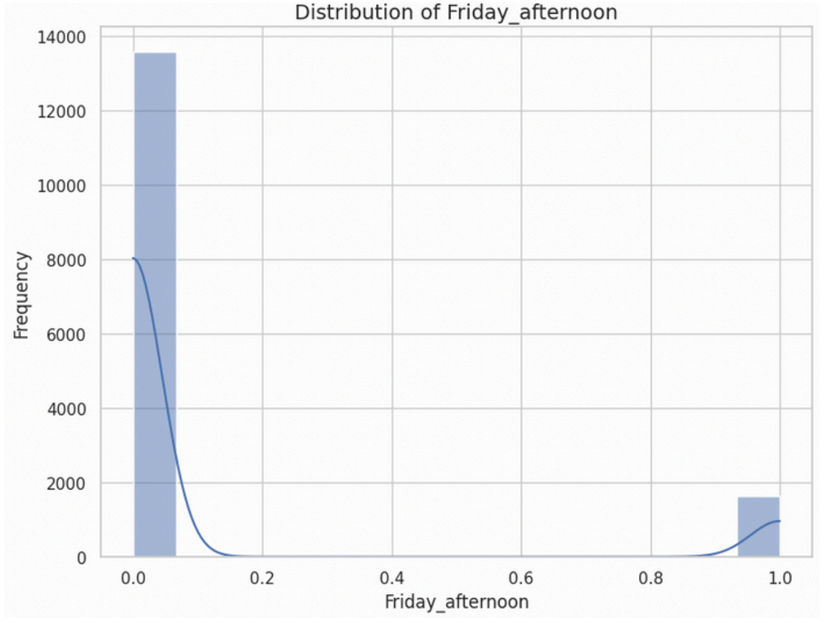
Distributions of Features

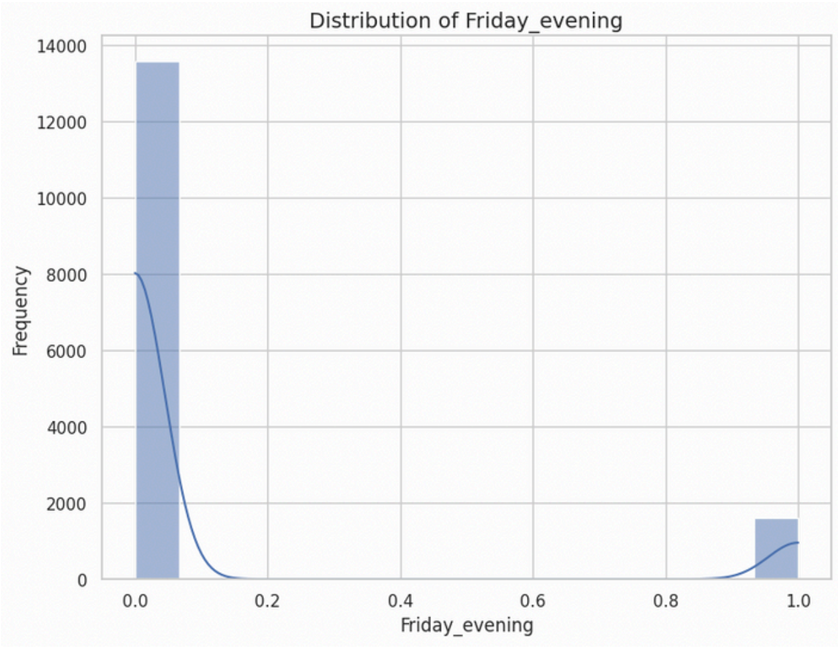


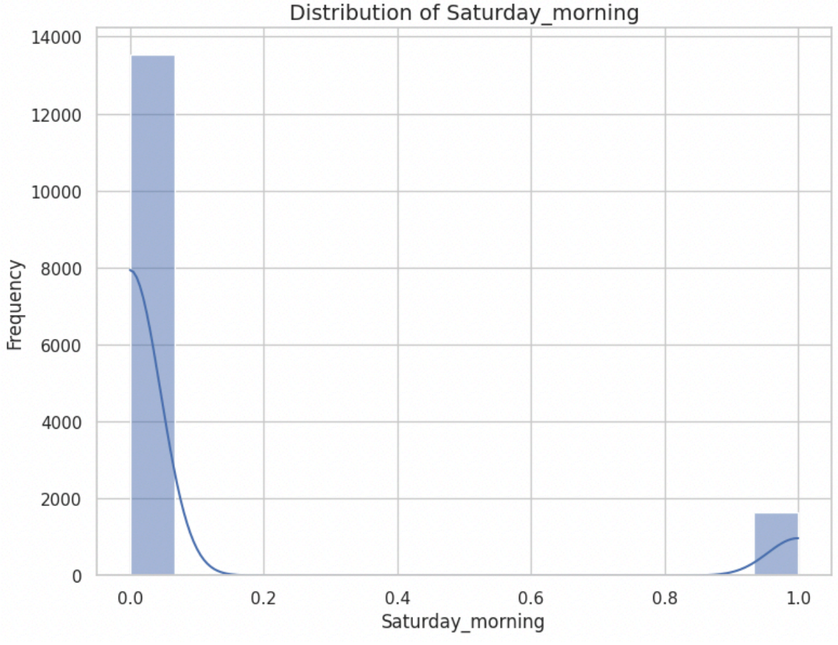


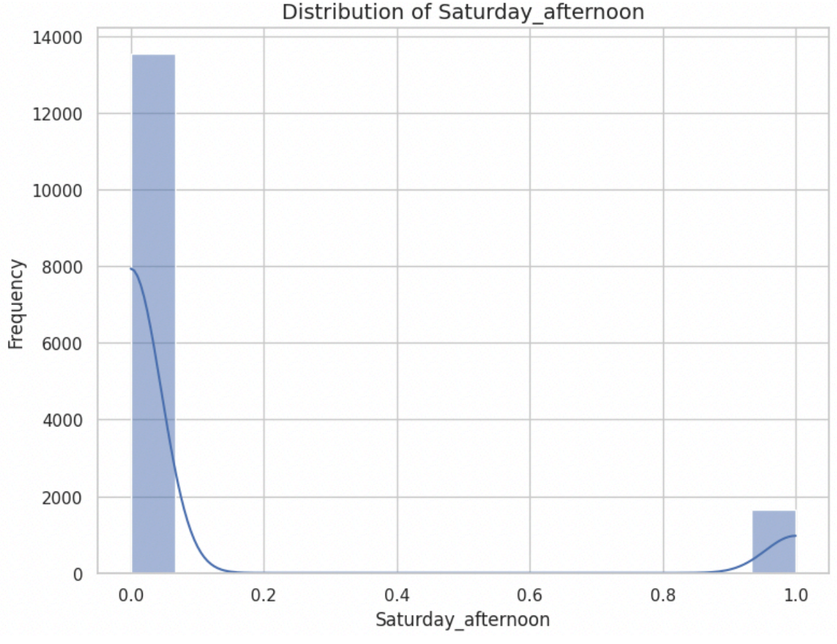


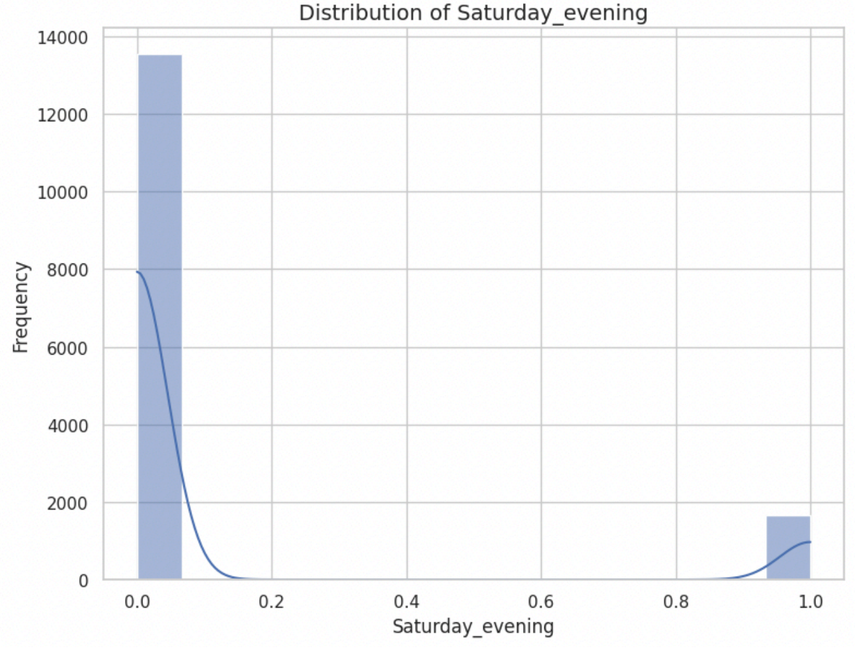

Distribution of Friday_morning
Frequency
Friday_morning


Distribution of Friday_afternoon
Frequency
Friday_afternoon

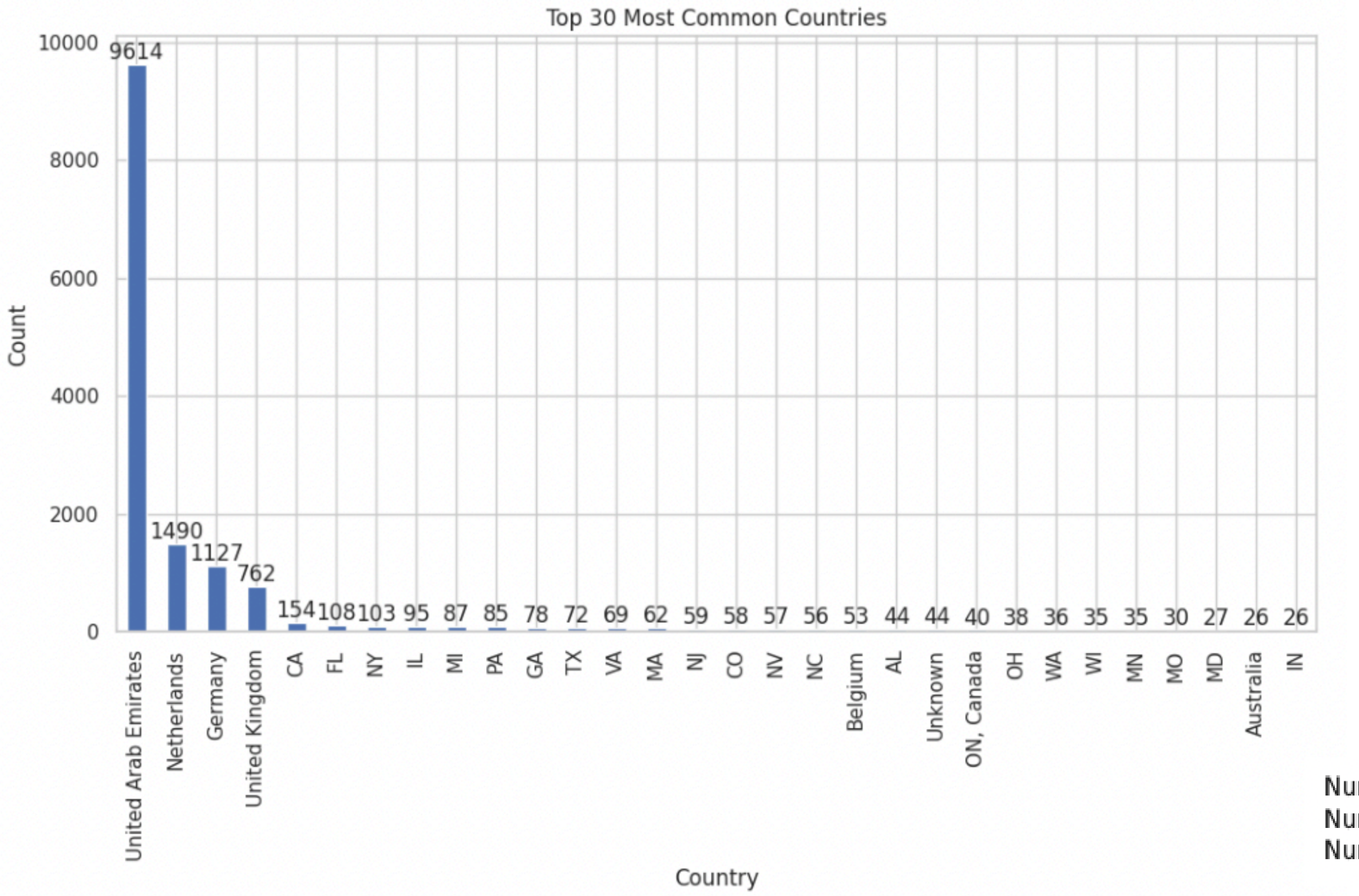

Distribution of Friday_evening
Frequency
Friday_evening


Distribution of Saturday_morning
Frequency
Saturday_morning


Distribution of Saturday_afternoon
Frequency
Saturday_afternoon

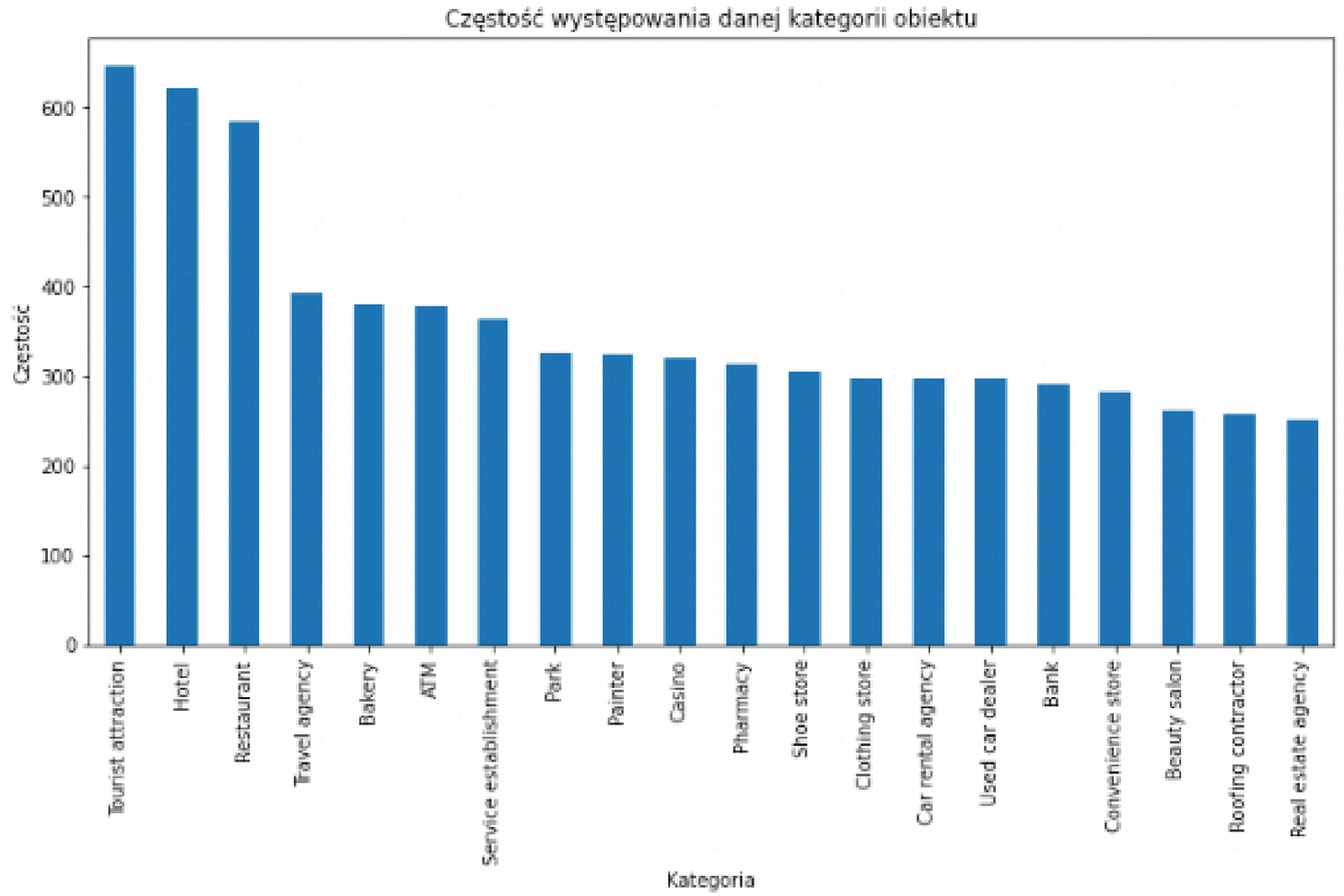

Distribution of Saturday_evening
Frequency
Saturday_evening

Distributions of Features




```
Number of different names for country 'India':  13
Number of different names for country 'Italy':  8
Number of different names for country 'Canada':  9
```

Distributions of Features



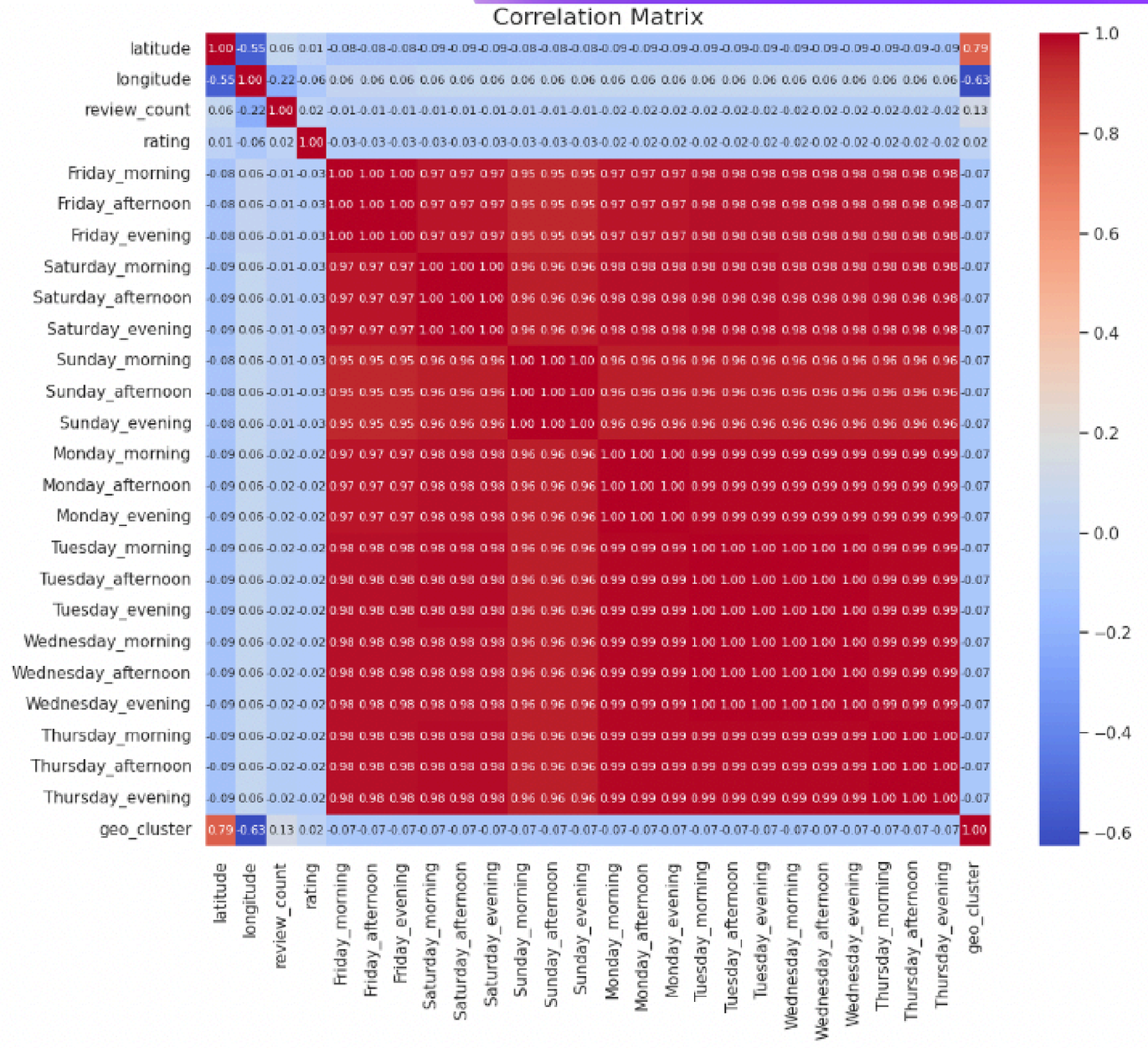
- **Tourist attraction**
- **Hotel**
- **Restaurant**

Correlations between Features

Prior to constructing the models, we conducted an analysis to identify correlations between various features. Plots, such as the one displayed on the slide, were utilized for this purpose. Red colour indicates a strong **correlation** between features.

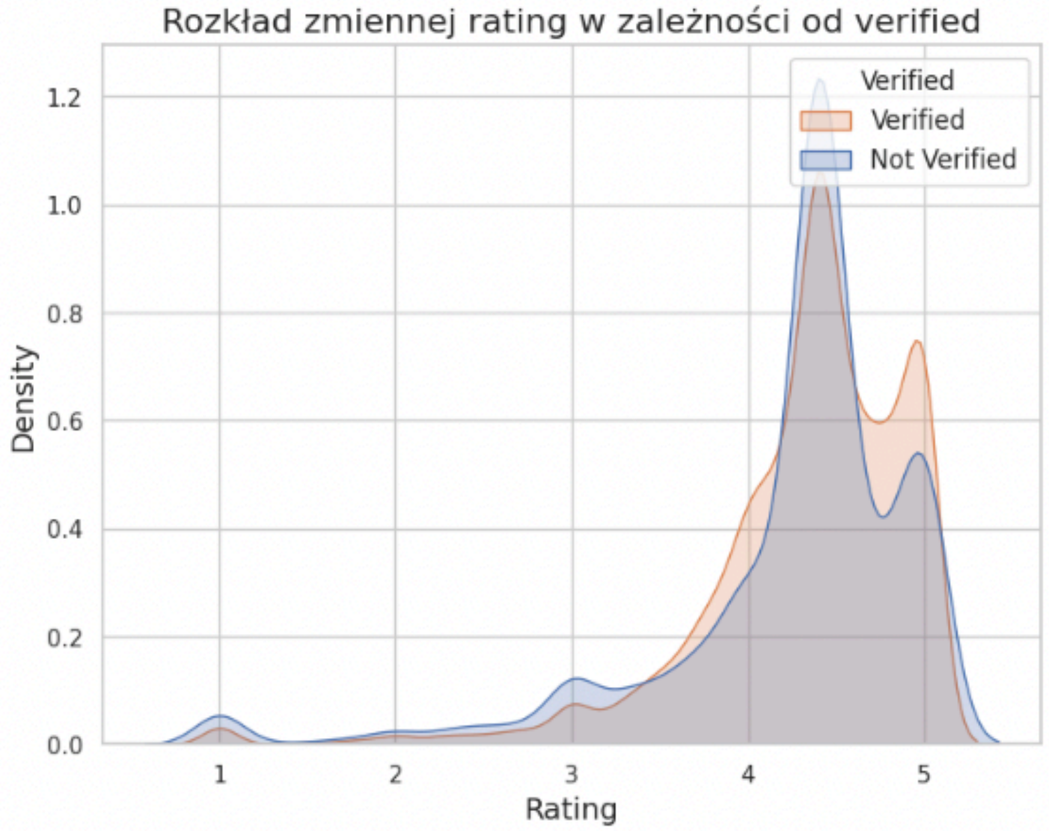
It is evident that there is a correlation between:

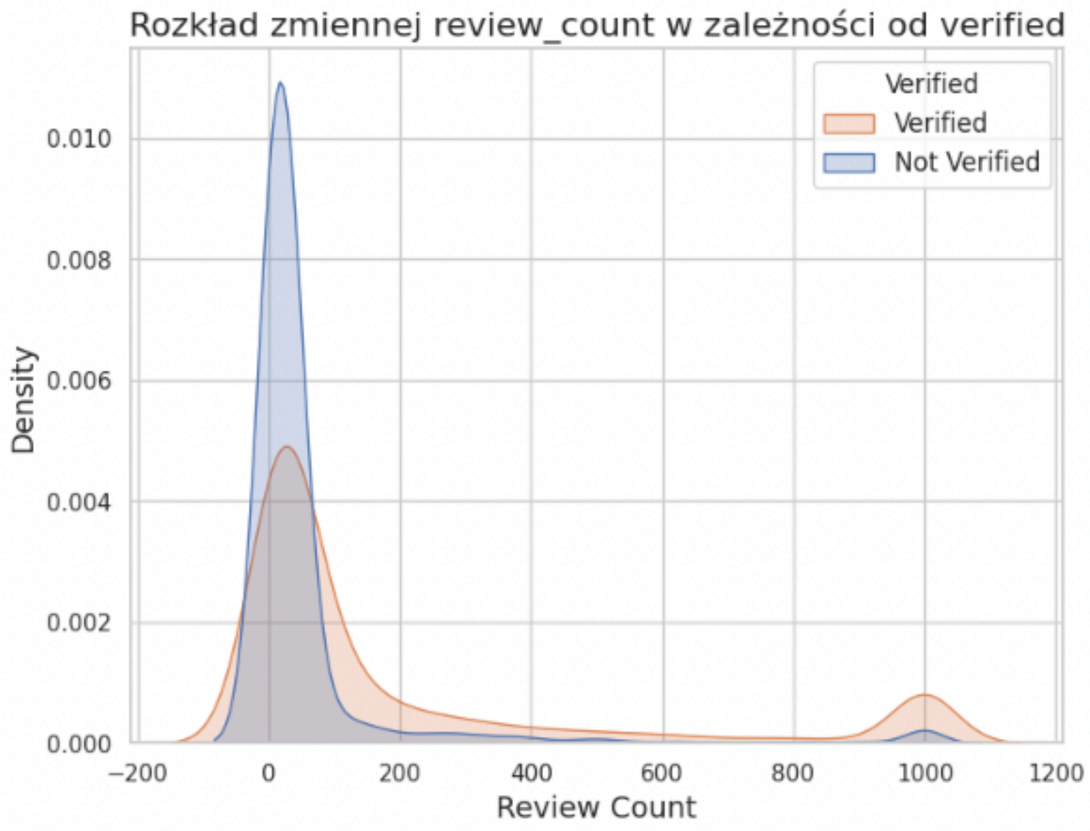
- All the features describing **opening hours**. This indicates that the data was i**ncorrectly sraped** from Google Maps.
- **longitude** and **latitude**
- **longitude** and **review_count**.



Correlations between Features

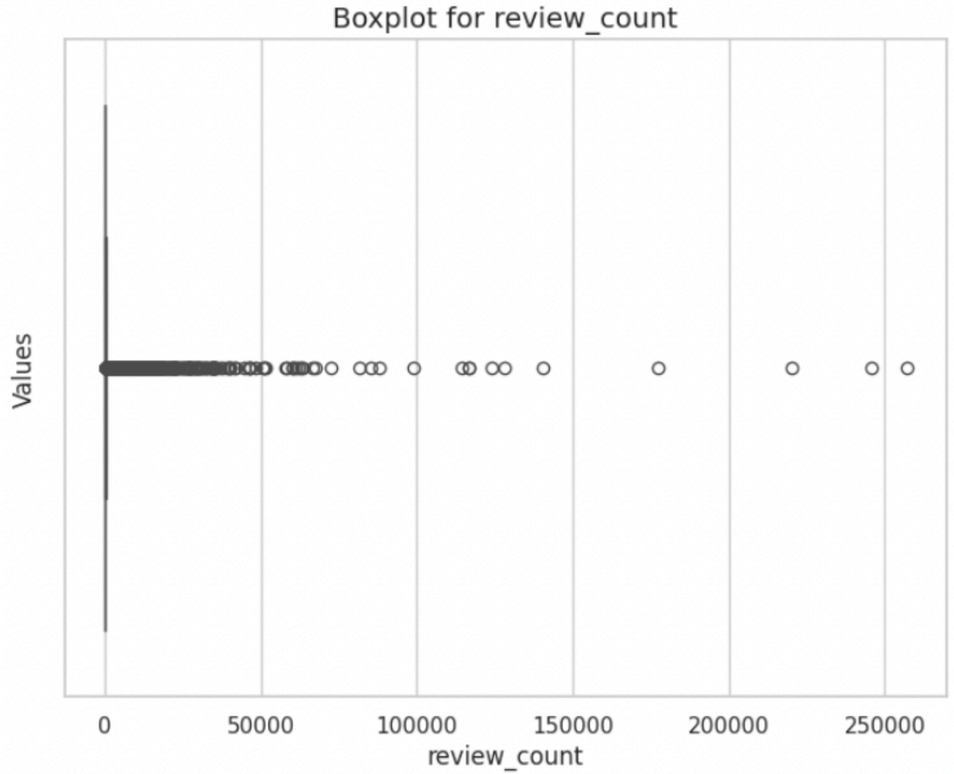
We also checked whether the fact that a business is **verified** has an **impact** on its reviews. As expected, the answer is positive.

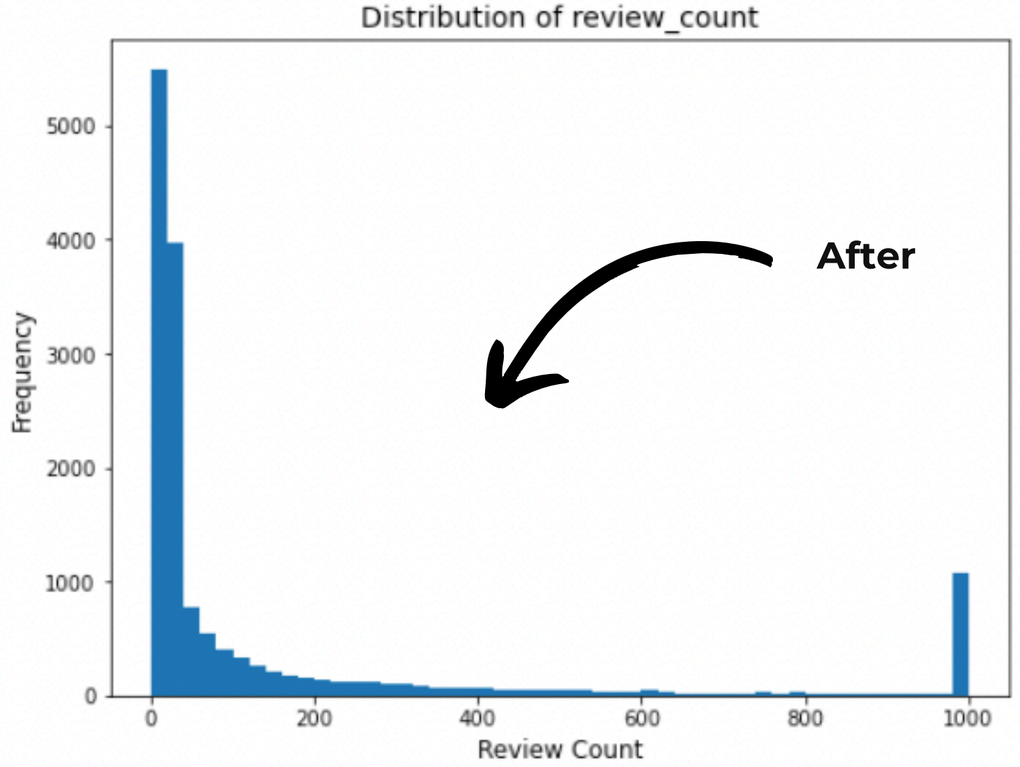




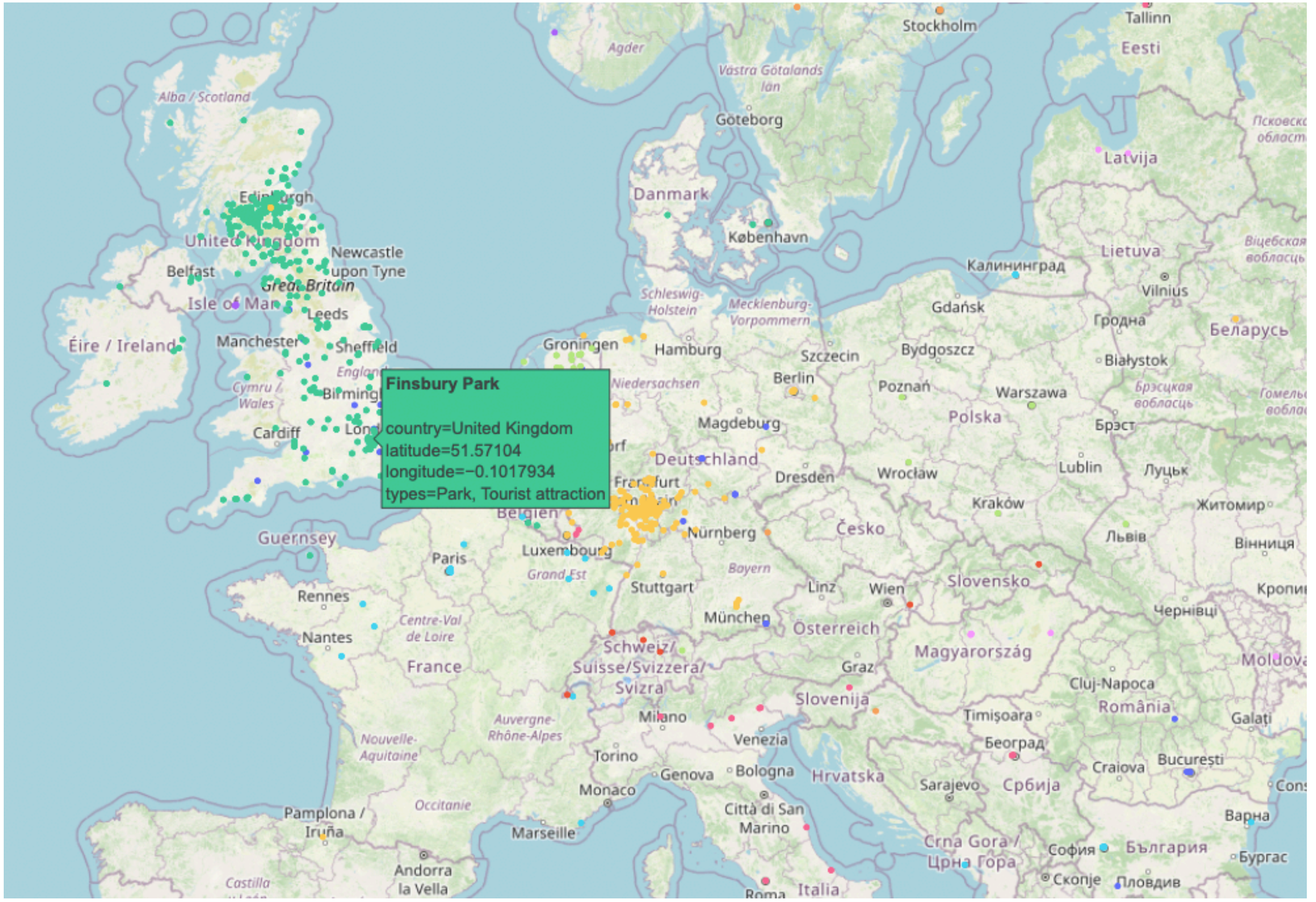
Outliers - extreme observations

We have noticed that all continuous numeric variables have some outliers (extreme observations). However we decided to leave them in columns such as: longitude, latitude, rating, because in these columns outliers were a natural consequence of the meaning behind the data. We only eliminated **outliers** in **review_count** column.





Map of the world



Feature Engineering

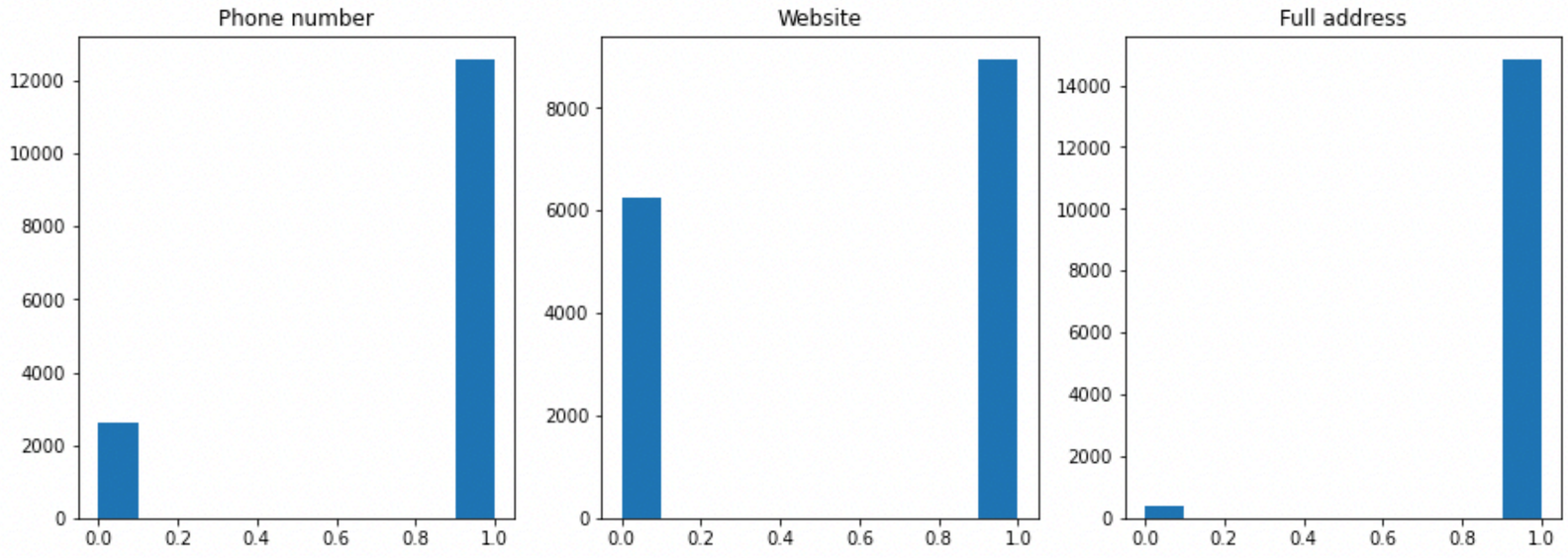
We decided that not all column that were included in the original dataset are a **valuable** source of information. To reduce the noise in data and its complexity, we decided to **remove** the following **columns**:

- **business_id, place_id**
- **timezone**
- **place_link** (link to the location on Google Maps)
- **state** (whether the object was open at the time of scraping the data)
- **Friday_morning, Friday_afternoon, Friday_evening, … , Thursday_evening** (binary columns describing opening hours that contained errors)
- **geo_cluster** (predefined clusters based on geographical location)

Feature Engineering

The next step was to transform some other columns into numerical ones. We decided that information such as phone number, website or full address are not necessary for our business case. We decided to form new features based on those columns. The only information that we keep is whether the phone number (or website or address) is **given** (then the new column takes the value of 1) or is **Unknown** (then the new column takes the value of 0). As an effect we obtained the following binary columns:

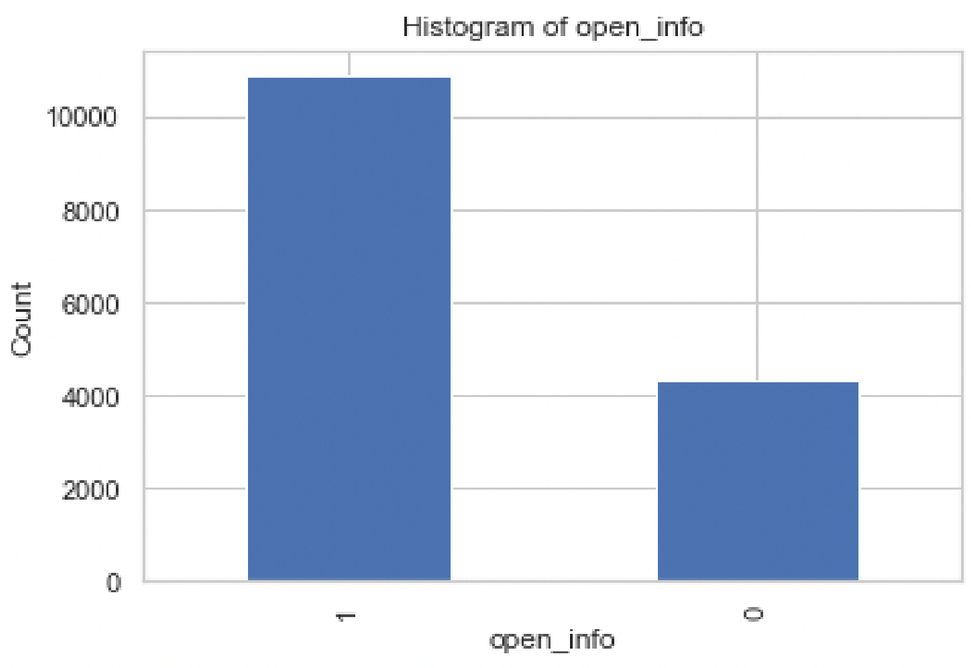
- **phone_number (0/1)**
- **website (0/1)**
- **full_address (0/1)**



Feature Engineering

Efforts have been taken to retrieve information about **opening hours** from the resources included in the original dataset. However, we decided to **drop** most of this information as well. This was due to the following:

- About **30%** of the opening hours were ***Unknown***
- Those places that had opening hours included shared similar patterns in terms of opening times (mostly **9 AM - 6 PM**)
- The attempt to retrieve information from text columns with opening hours resulted in about 80% accuracy, not 100% accuracy, leading to more noisy data.

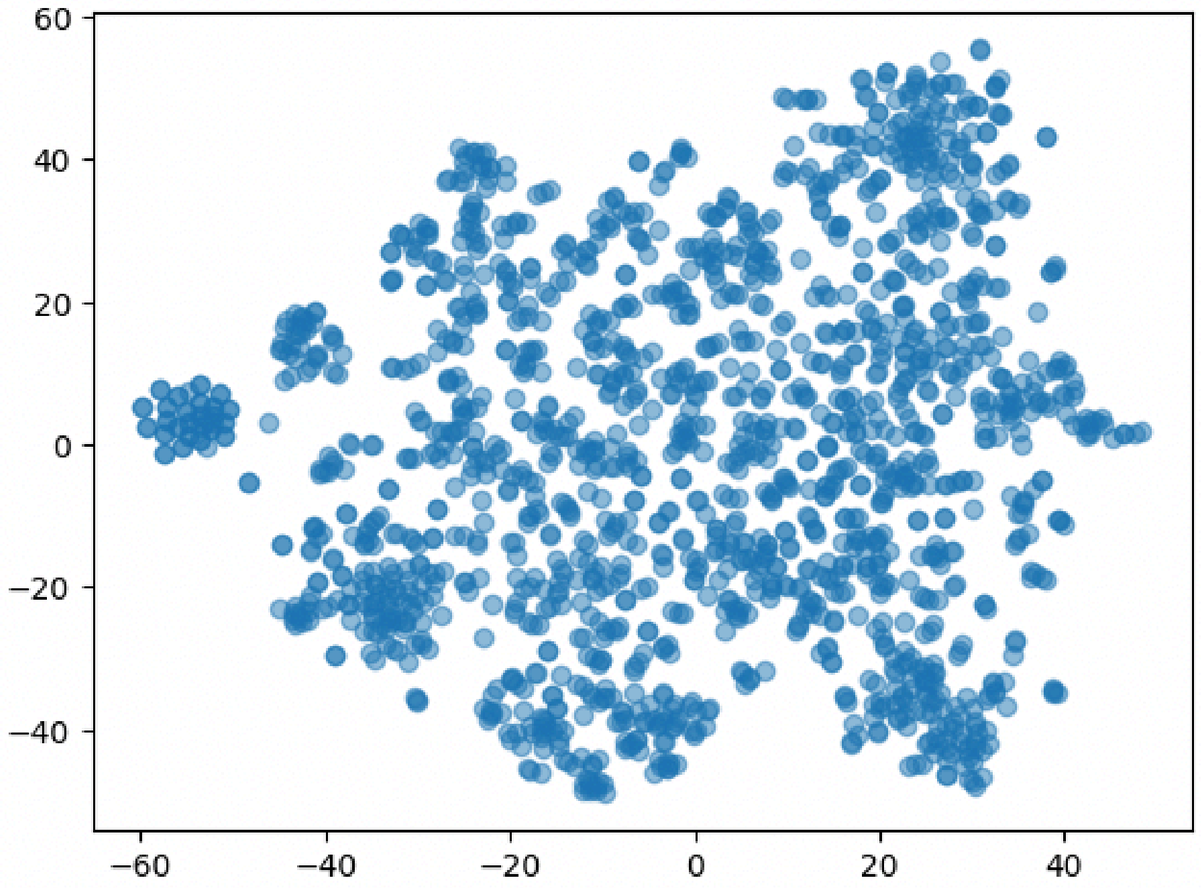


Hence the decision to only include information about the fact of **unknown** opening hours vs. **given**. The new binary column was called: **opening_info (0/1).**

Retrieving business type

The column in the original dataset called “**types**” contained information about the labels of business types that the location had. Unfortunately, multiple labels could be assigned to one location. In order to retrieve information about a general type of the business, we harnessed **cutting-edge NLP algorithms**.

We performed **embeddings** for each unique type of business activity. The results were then mapped to **2D** space and are shown on the graph.



Retrieving business type

We wanted to reduce the amount of different business types, resulting in more general information. For example, we decided that the **difference** between "Cathedral' and 'Catholic cathedral' is **too small** to be taken into account by our recommendation system.

Hence, we performed **clustering** on the embedding vectors. This resulted in more **general groups** of business **types**. We selected the number of clusters to be 600.

Here we can see a cluster of **fashion stores:**

['Linens store' 'Clothing store' 'Designer clothing store' 'Dress store', "Women's clothing store" "Men's clothing store", ... 'Custom t-shirt store' 'Fashion design school', 'Work clothes store' 'Garment exporter' 'Underwear store']

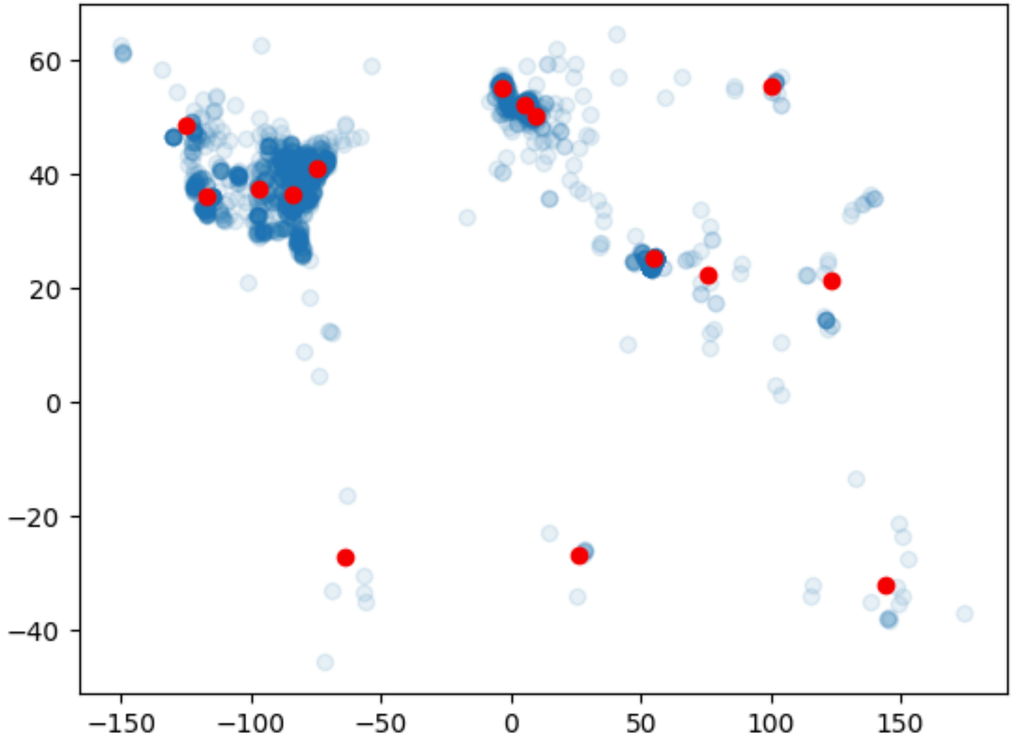
Retrieving business type - majority voting

cluster_type for this row = **27**

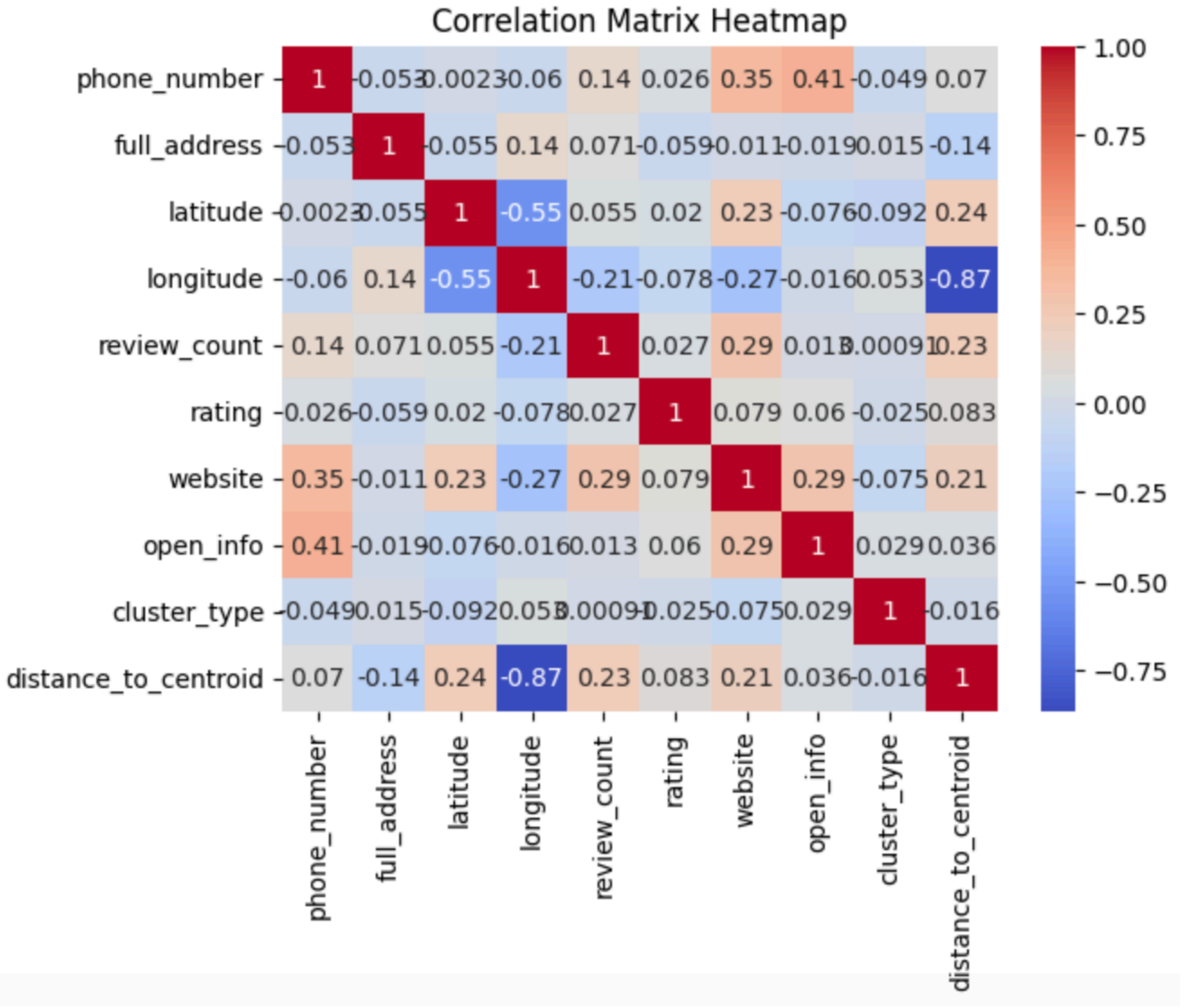
Feature Engineering

We decided to engineer another feature from scratch, which is the **distance to the nearest agglomeration**. This is to represent whether the certain google place is surrounded by a dense group of other businesses, or whether it is far from civilisation.

To achieve that, we performed **clustering** based only on **geographical** data to retrieve centres of agglomerations called **centroids**.



Validation results of exploration and feature engineering



Increase data quality

To sum up the preprocessing stage, we used the following techniques to enhance the clarity and quality of the dataset.

Outliers

We overwritten values that did not align with the average values of a specific feature. By setting them to a high, fixed number we ensured that they do not outshine remaining, less extreme values.

Feature engineering

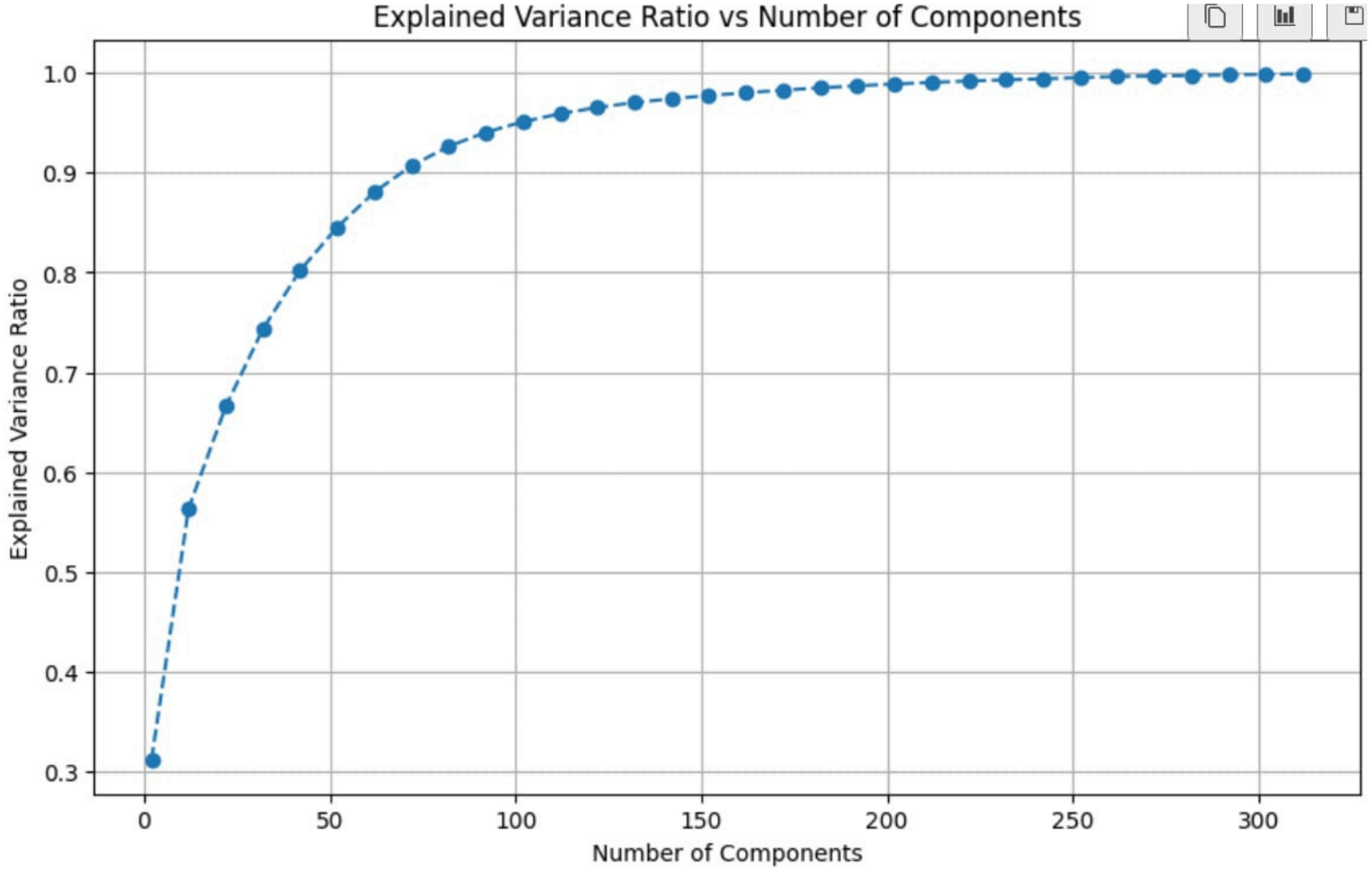
We leveraged existing features, even those with inherent faults, to engineer new ones with a higher potential to enhance model accuracy.

Scaling

We standardized all features to have values between 0 and 1, ensuring that each feature has an equal opportunity to differentiate observations in clustering models.

Reducing the dimensionality of the dataset

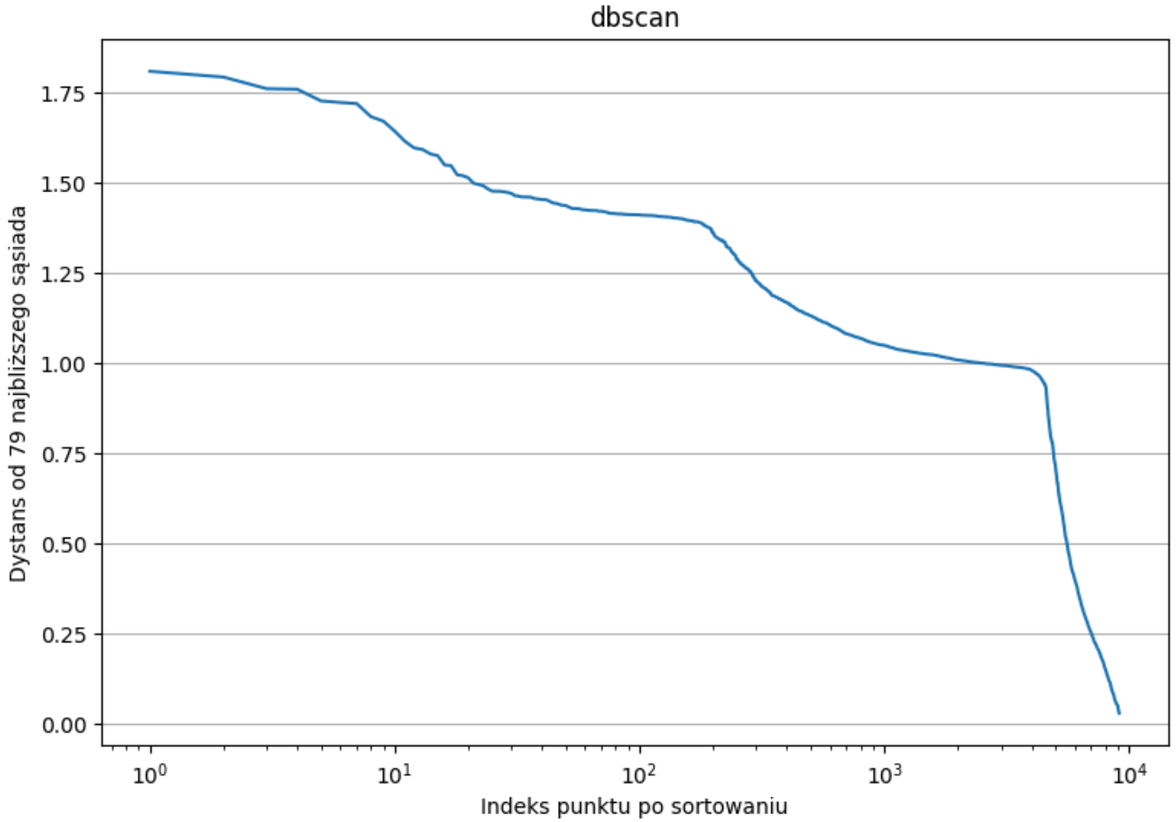
Using PCA we reduced the number of features to only **40.** However, this accounts for over **80 %** of variance in the data.



Clustering algorithms

DBSCAN

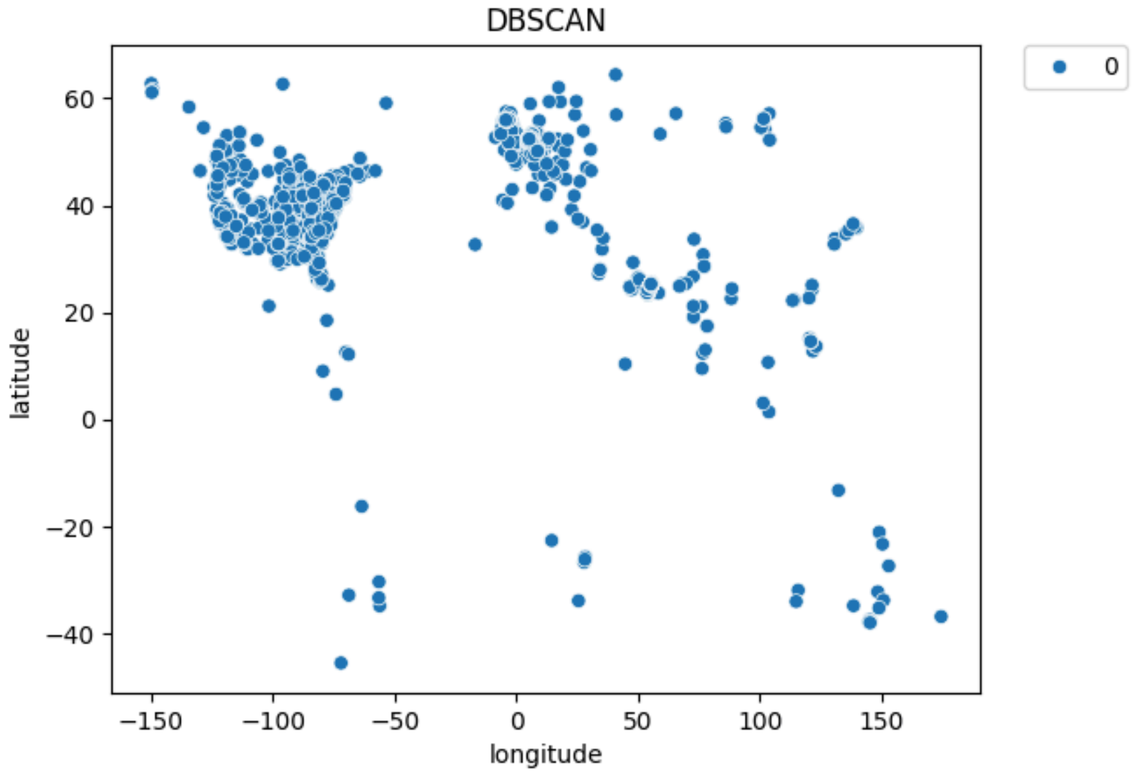
Looking for the best algorithm, we started with DBSCAN. We performed an analysis in order to tune this model to the best of its capacity. Unfortunately, the search for the best parameters did not work as expected.



A lot of points seem to be outliers.

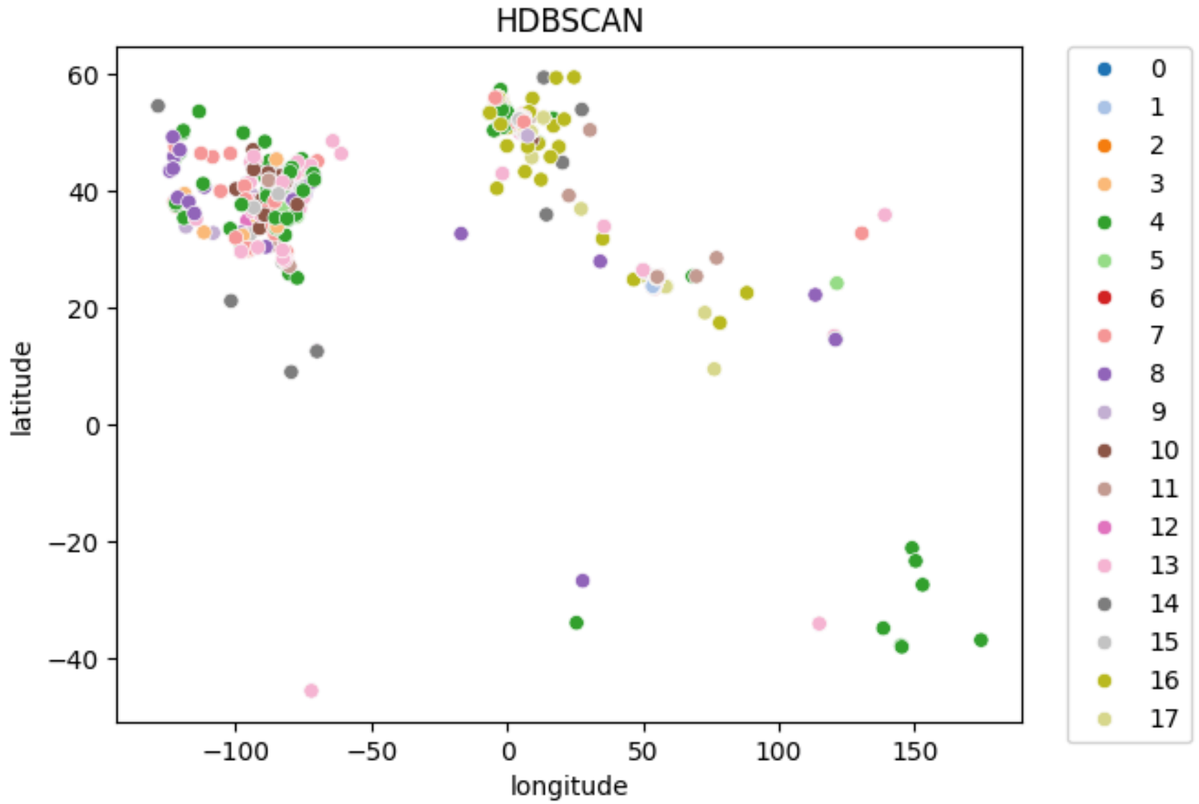
DBSCAN

We decided to use a popular method of hyper-tuning called Grid Search. According to our business case we tried to **minimise** the amount of points **not classified** for any clusters. This resulted in all points being classified to only 1 cluster.



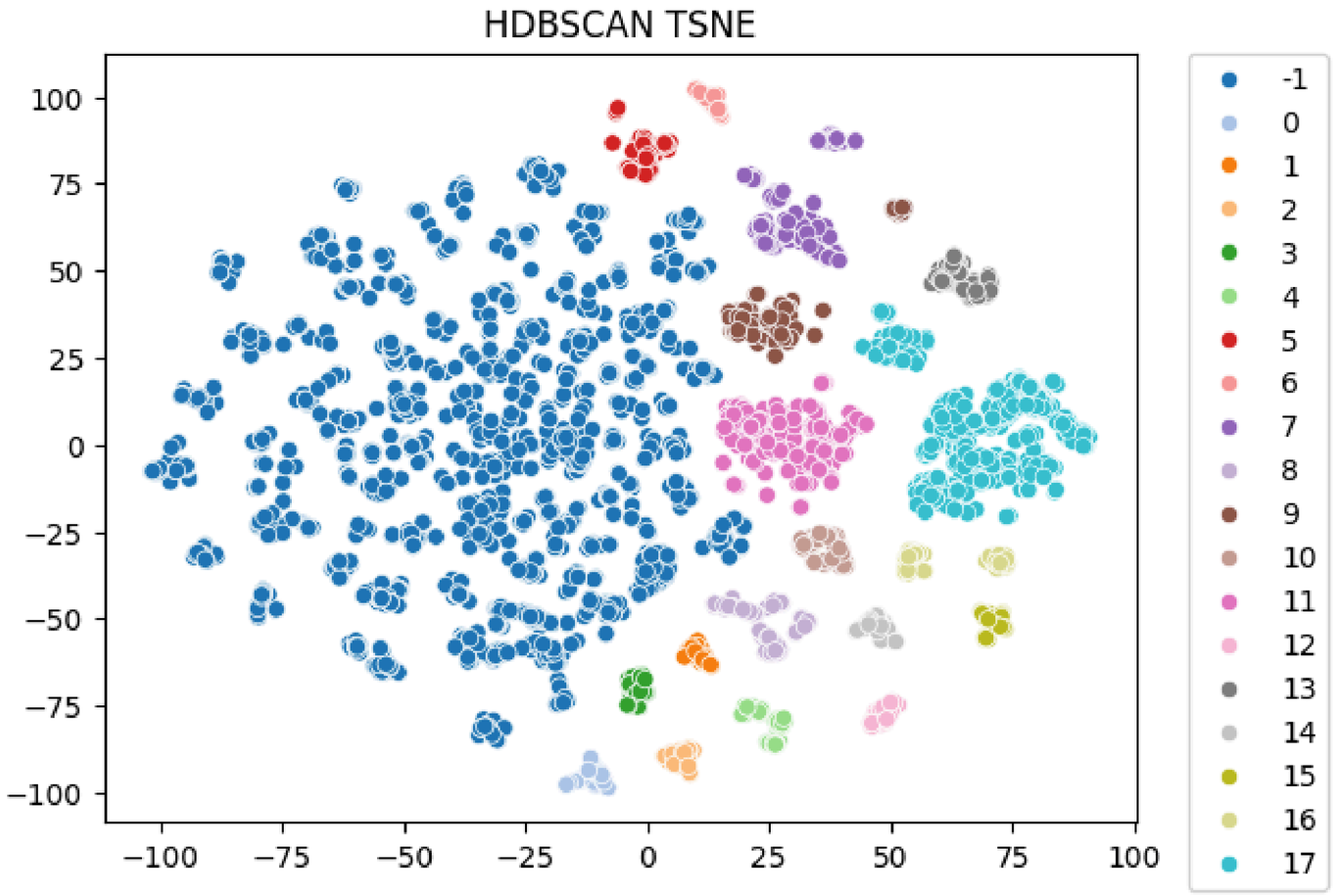
HDBSCAN

We then decided an extended version of DBSCAN to boost the performance of the model. Below graph shows the visualisation of the results.



- Some clusters are similar in terms of the **location** of points
- Other ones are more spread out indicating that there are other factors that influence the cluster label more than the location, such as **ratings, types of businesses**

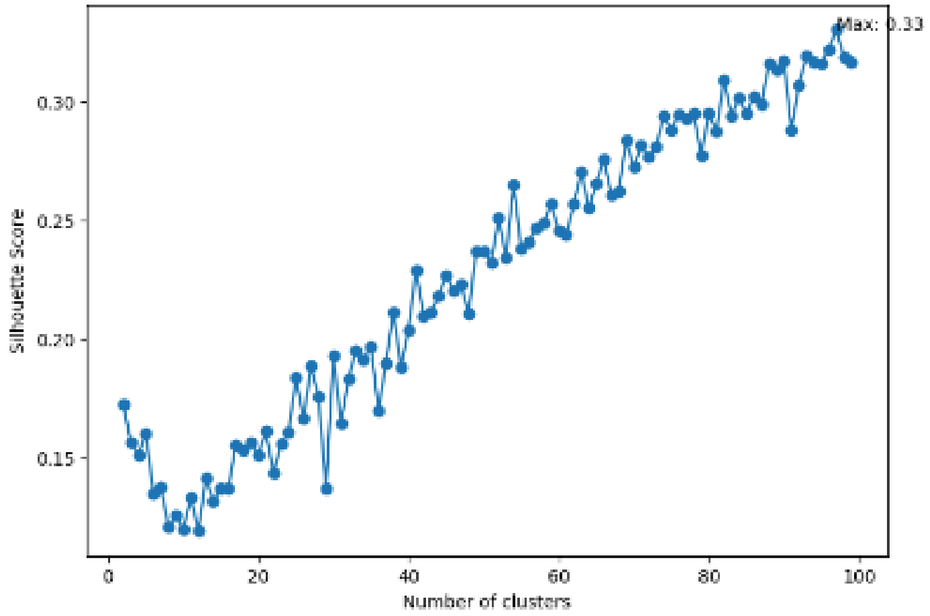
HDBSCAN

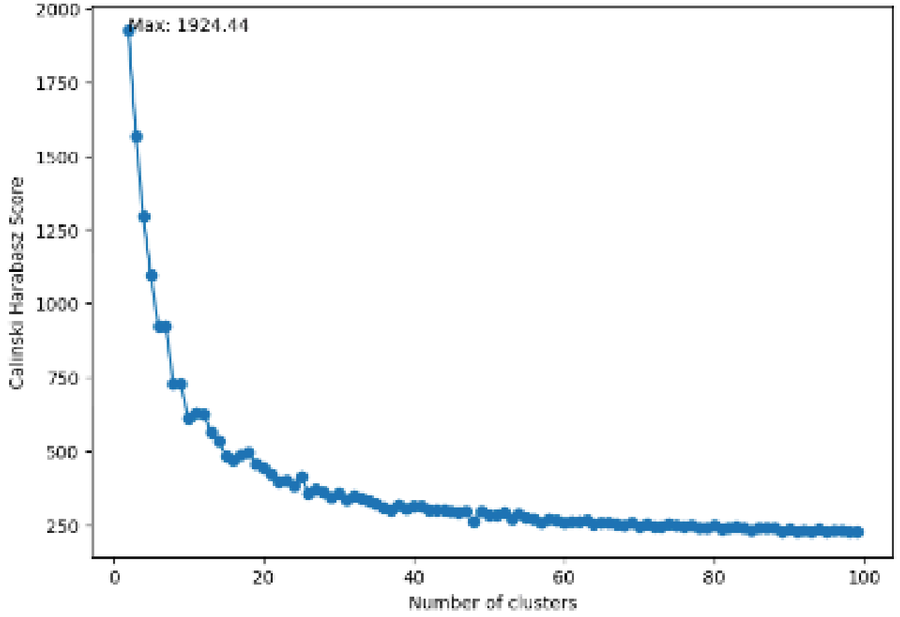


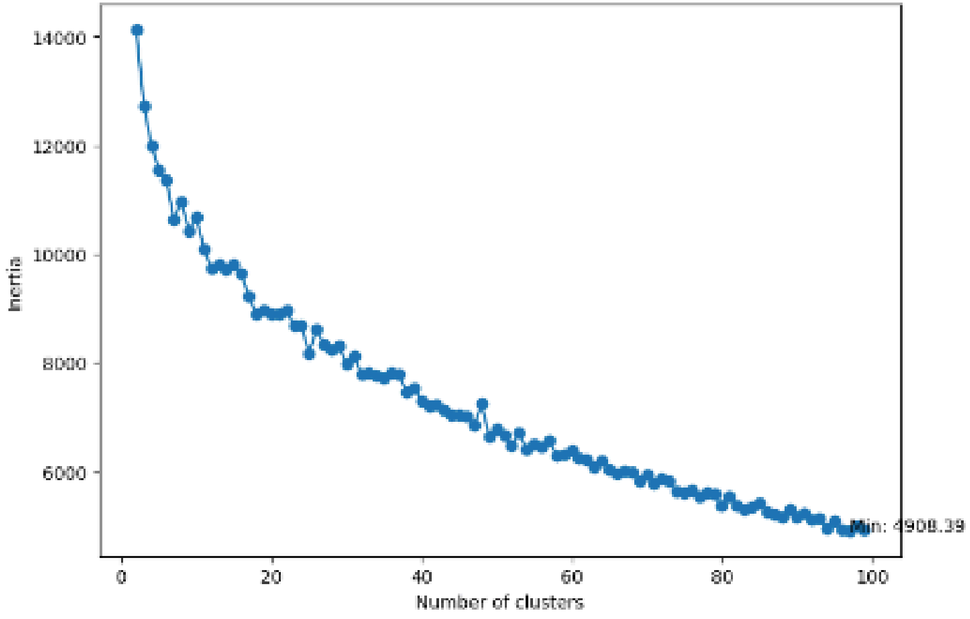
- Many points were not assigned to any cluster (-1 label)
- Similar google places seem to be assigned to separate clusters

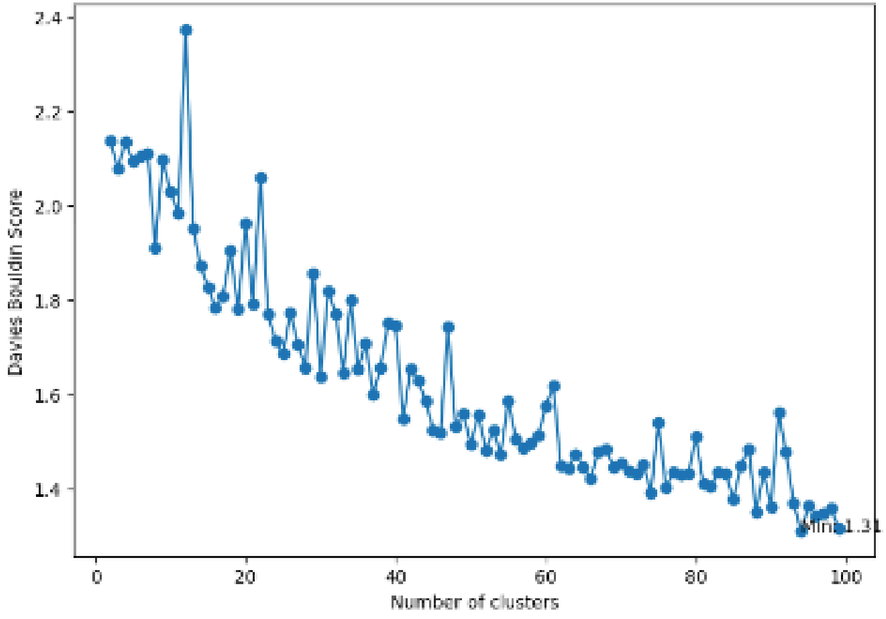
KMeans

We were not satisfied with the results of HDBSCAN, so we decided to try one more algorithm called **KMeans**. Initially we did not know how many clusters of google places points we wanted to achieve so we performed a broad analysis with different score measures.









KMeans

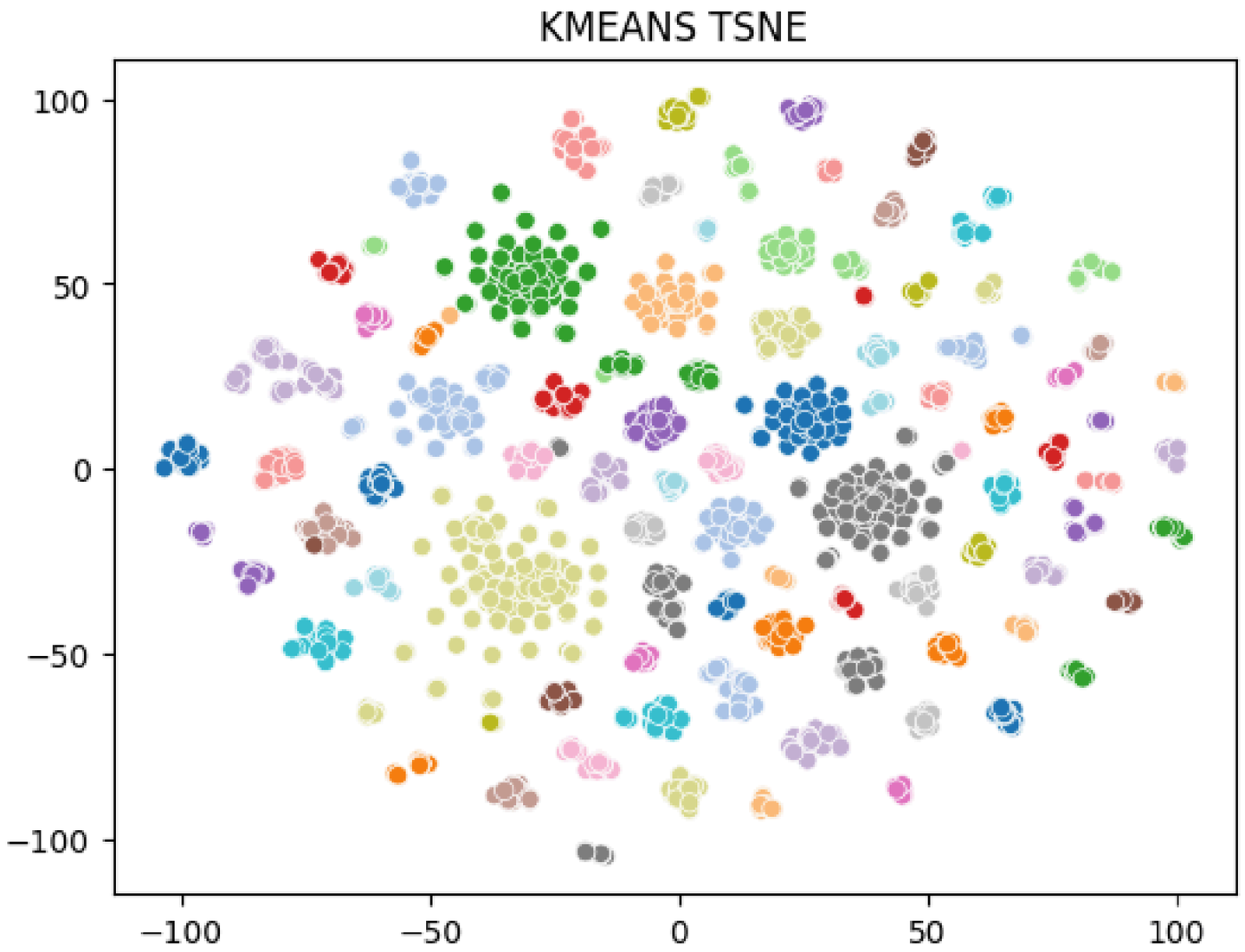
```
Optimal number of clusters according to Silhouette Score: 97
Optimal number of clusters according to Calinski Harabasz Score: 2
Optimal number of clusters according to Inertia: 97
Optimal number of clusters according to Davies Bouldin Score: 94
```

Decision: 90 clusters

KMeans

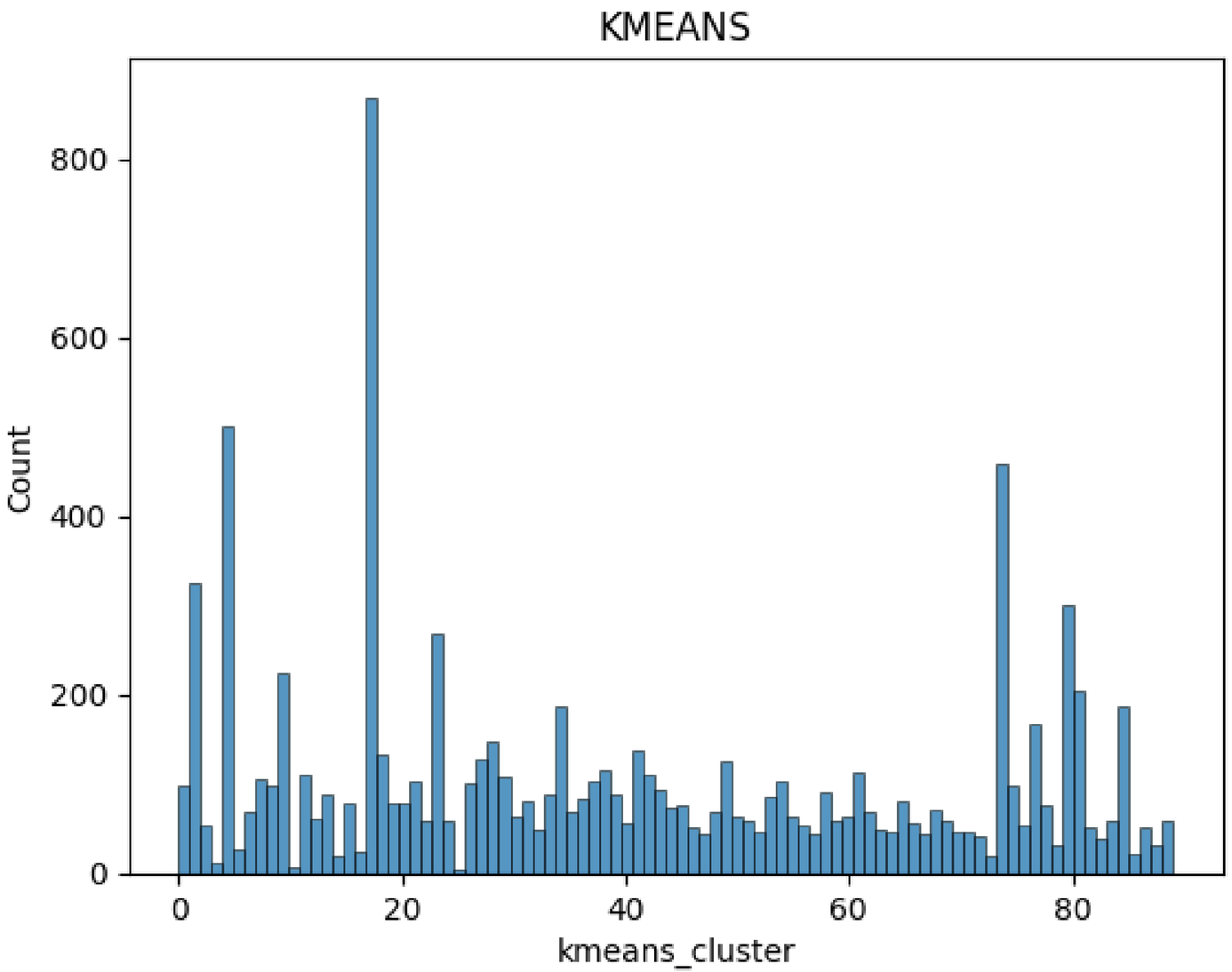
Concerns:

- close points generally were assigned to the same cluster but because of the high number of clusters, they can be very small in terms of the number of members



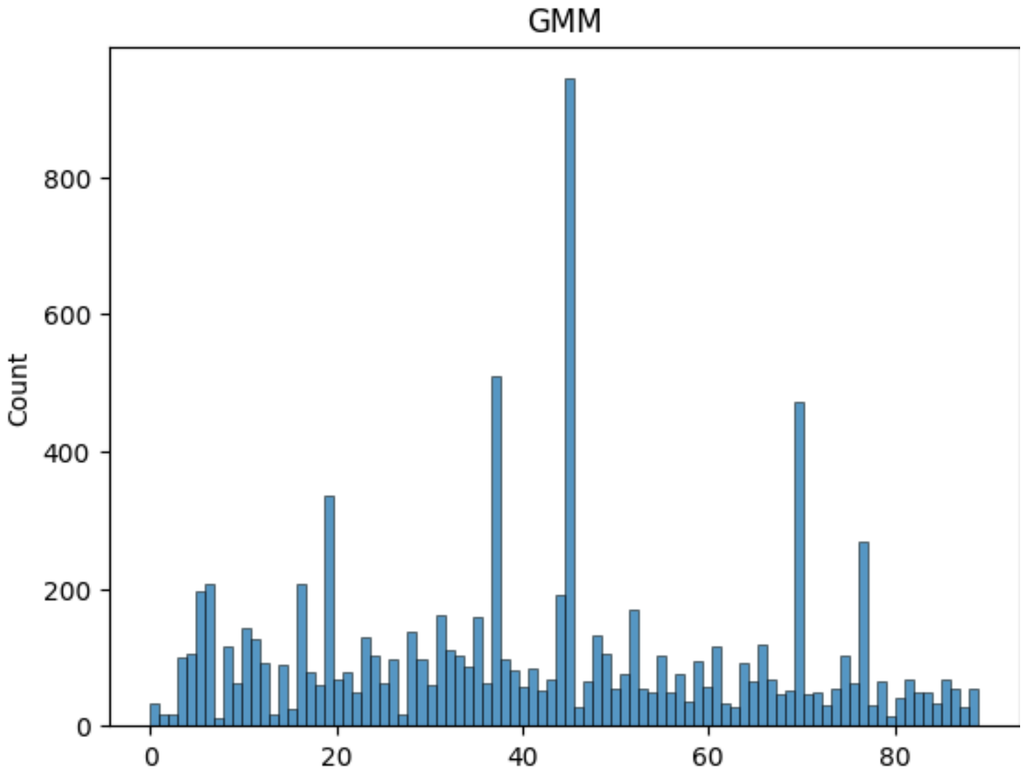
KMeans

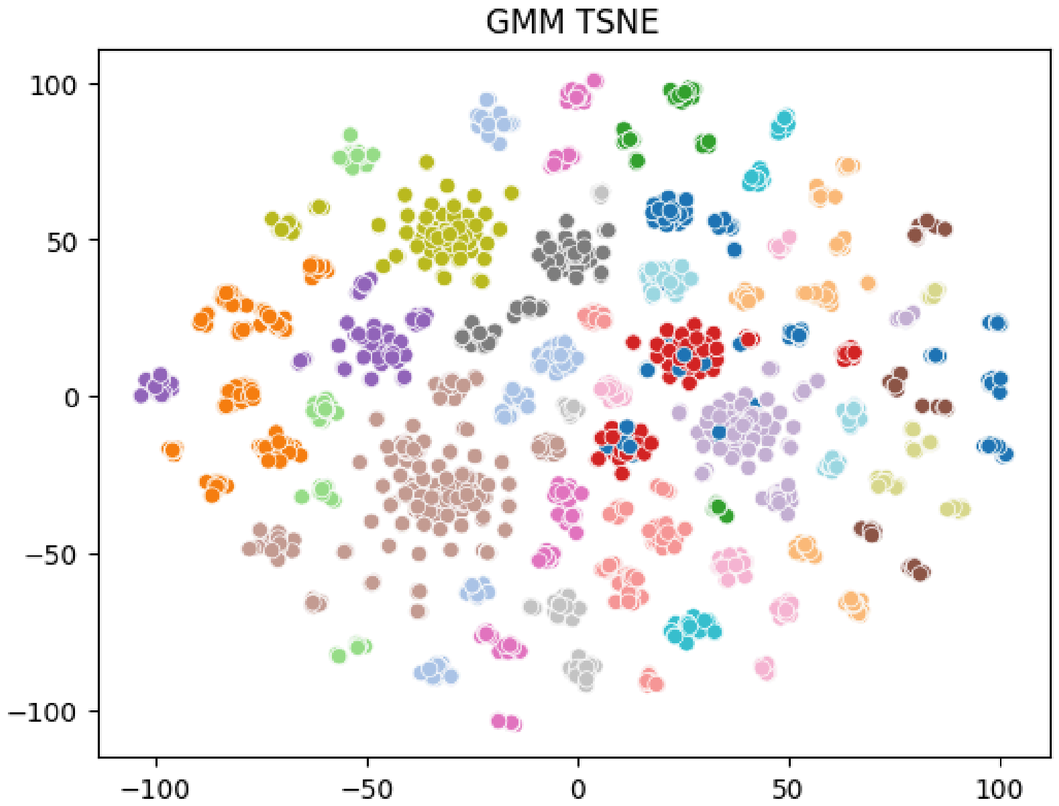
- Most of the clusters contain **50 - 100 google points**



Gaussian Mixture Model

After running multiple experiments with the data, we decided to use Gaussian Mixture Model as the final one. As we already saw on visualisations, the right amount of **clusters** for this case should oscilate about **90** and that's the number that we selected.



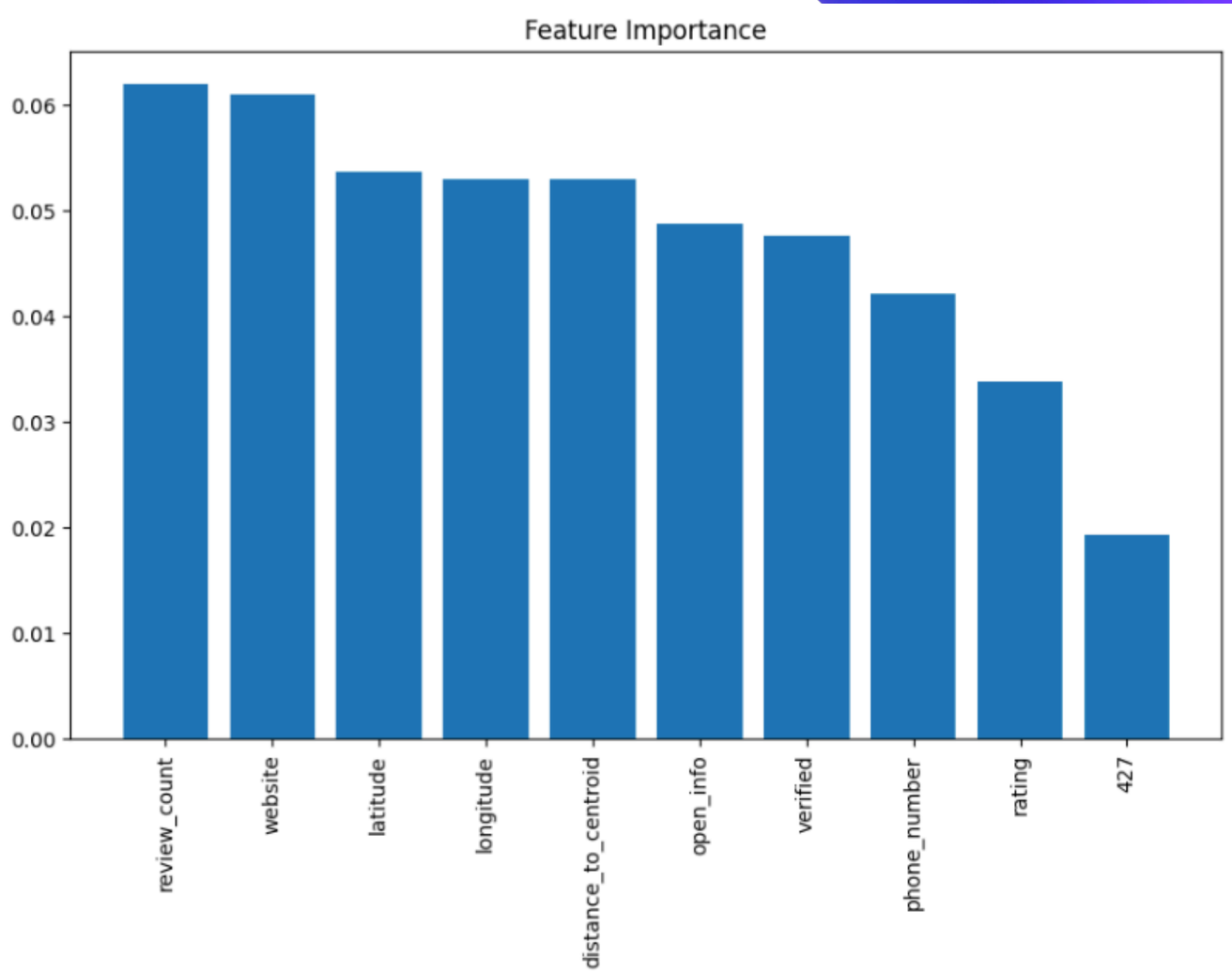


Interpretation

Feature impact

We wanted to test what features impacted the clustering process the most. In order to achieve that we utilized knowledge gained during the previous project about **supervised learning techniques.**

Column labelled as **427** is a column that represents the **type of the business**, along with 300 other column that were **not** included on the plot.



Recommendation system

Finally we were able to build a recommendation system based on the clustering that was performed. When we choose a place from google maps, the model searches the cluster to which this place is assigned and recommends other places, listing them from closest to furthest.

	name	types	city	country_corrected	gmm_cluster
4536	AB FITNESS	Gym	Dubai	United Arab Emirates	36

name	types	city	country_corrected	gmm_cluster
FITOX Gym	Gym	Dubai	United Arab Emirates	36
Lions Gym	Gym	Dubai	United Arab Emirates	36
Life For You Gym BURDUBAI	Gym, Fitness center	Dubai	United Arab Emirates	36
SPEED FITNESS (BRANCH)	Gym	Dubai	United Arab Emirates	36
ReStart Fitness Center & Gym	Gym, Dietitian, Fitness center, Karate club, K...	Dubai	United Arab Emirates	36

Cluster name: Fitness in Dubai

Recommendation system

name	types	city	country_corrected	gmm_cluster
Tiki Pool	Swimming pool, Outdoor bath, Public swimming p...	Wassenaar	Netherlands	79

	name	types	city	country_corrected	gmm_cluster
4754	Aquariumwarenhuis.nl	Aquarium	The Hague	Netherlands	79
5754	Sea Life Scheveningen	Aquarium, Tourist attraction	The Hague	Netherlands	79
4339	Aquarium House Romberg	Aquarium, Pet store, Aquarium shop, Fish store...	Delft	Netherlands	79
7690	Picasso Aquarium	Aquarium shop	Aalsmeerderbrug	Netherlands	79
1156	Aquariumvereniging Aqua-Verniam.nl	Aquarium	Amstelveen	Netherlands	79

Cluster name: Swimming in the Netherlands

Recommendation system

	name	types	city	country_corrected	gmm_cluster
4520	Police Scotland	Police department	Burntisland	United Kingdom	14

	name	types	city	country_corrected	gmm_cluster
7165	British Transport Police	Police station	Kirkcaldy	United Kingdom	14
2587	Police Scotland Drylaw Police Station	Police station	Edinburgh	United Kingdom	14
5028	Police Scotland Dalgety Bay Police Station	Police department	Dunfermline	United Kingdom	14
8293	Police Scotland Edinburgh	Police department, Police station	Edinburgh	United Kingdom	14
1897	Police Scotland Wester Hailes Police Station	Police department, Police station	Edinburgh	United Kingdom	14

Cluster name: Police in the UK

Recommendations during validations

```
sample = df_validate.sample(1)

sample[['name','types','city','country_corrected','gmm_cluster']]
```

✓ 0.0s Python

	name	types	city	country_corrected	gmm_cluster
1873	Al Fardan Exchange L.L.C. Bada Zayed	Money order service, Currency exchange service...	Abu Dhabi	United Arab Emirates	59

Cluster name: Money in the UAE

	name	types	city	country_corrected	gmm_cluster
5952	Al Hilal Bank - Nad Al Hammar Branch	Bank, ATM, Business banking service, Car finan...	Dubai	United Arab Emirates	59
3113	ADCB Cash & Cheque Deposit ATM - Mirdif City C...	ATM	Dubai	United Arab Emirates	59
4917	Ajman Bank ATM - Westzone Supermarket	ATM	Dubai	United Arab Emirates	59
2956	RAKBANK ATM - Al Naboodah Construction Group A...	ATM	Dubai	United Arab Emirates	59
5722	ADCB Cash & Cheque Deposit ATM - Arabian Centr...	ATM	Dubai	United Arab Emirates	59

Final takeaways

- Number of clusters and other parameters- selection methods
- Crucial role of word embeddings
- Using Pipelines for seamless flow of new data sources
- Further improvements - better UI

Resources and technologies used during the project

scikit-learn.org/stable/tutorial/basic/tutorial.html

Lecture materials from Warsaw University of Technology

https://www.kaggle.com/datasets/azharsaleem/location-intelligence-data-from-google-map

https://www.reddit.com/r/dataisbeautiful/comments/sp027d/20_most_reviewed_places_on_google_maps_oc/

Python, NumPy, pandas, Matplotlib, scikit-learn, sentence_transformers, geopy, scipy

Thank you for reading the report

Project repository

https://github.com/kowalskihubert/Google-Places-Clustering