BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INFORMATION SYSTEMS
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

THE RETAIL SITE LOCATION DECISION SYSTEM IN BRNO

SYSTÉM PRO ROZHODOVÁNÍ O UMÍSTĚNÍ MALOOBCHODNÍCH PRODEJEN V BRNĚ

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR **OLEKSANDR TURYTSIA**
AUTOR PRÁCE

SUPERVISOR **Ing. JIŘÍ HYNEK, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2024

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Bachelor's Thesis Assignment

153877

Institut:	Department of Information Systems (DIFS)
Student:	**Turytsia Oleksandr**
Programme:	Information Technology
Title:	**The Retail Site Location Decision System in Brno**
Category:	Information Systems
Academic year:	2023/24

Assignment:

1. Study Retail Site Location Decision Process. Analyze current approaches.
2. Study GIS and data processing for GIS.
3. Analyze available open data of Brno datahub portal. Analyze the problem of Retail Site Location Decision Process in Brno.
4. Based on the results of the analysis, design system for Retail Site Location Decision Process in Brno which would recommend suitable areas.
5. Implement the designed system.
6. Evaluate the implemented solution. Test its usability.

Literature:

- Huff, D. L. (1964). Defining and estimating a trading area. *Journal of marketing*, *28*(3), 34-38.
- Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *The annals of mathematical statistics*, 832-837.
- Roig-Tierno, N., Baviera-Puig, A., Buitrago-Vera, J., & Mas-Verdu, F. (2013). The retail site location decision process using GIS and the analytical hierarchy process. *Applied Geography*, *40*, 191-198.
- Saaty, T. L. (1988). *What is the analytic hierarchy process?* (pp. 109-121). Springer Berlin Heidelberg.

Requirements for the semestral defence:
Items 1 to 4.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor:	**Hynek Jiří, Ing., Ph.D.**
Head of Department:	Kolář Dušan, doc. Dr. Ing.
Beginning of work:	1.11.2023
Submission deadline:	16.5.2024
Approval date:	30.10.2023

Abstract

Location plays a key role in the success of a business. No amount of property features such as building, decorating, or price can overcome the negative impact of a poor location. A strategically positioned business not only reduces financial risks but also enhances the likelihood of achieving success. This work aims to develop a system that implements a methodology to assist retailers in making informed location decisions. The system was evaluated with the data provided by the City of Brno.

Abstrakt

Lokalita má klíčový význam pro úspěch podnikání. Žádné vlastnosti nemovitosti, například budova, vybavení nebo cena, nemohou překonat negativní dopad špatné polohy. Strategicky dobře umístěný podnik nejen snižuje finanční rizika, ale také zvyšuje pravděpodobnost dosažení úspěchu. Cílem této práce je vyvinout systém, který implementuje metodologii pomáhající maloobchodníkům při informovaném rozhodování o umístění prodejny. Systém byl vyhodnocen na základě dat poskytnutých městem Brnem.

Keywords

Location information, business success, property features, site evaluation, subjective requirements, location-based decision-making, decision support system

Klíčová slova

Informace o lokalitě, obchodní úspěch, vlastnosti nemovitosti, hodnocení lokality, subjektivní požadavky, rozhodování na základě lokality, systém podpory rozhodování

Reference

TURYTSIA, Oleksandr. *The Retail Site Location Decision System in Brno*. Brno, 2024. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Jiří Hynek, Ph.D.

The Retail Site Location Decision System in Brno

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Ing. Jiří Hynek Ph.D.

.....
Oleksandr Turytsia
May 9, 2024

Acknowledgements

I would like to thank Ing. Jiří Hynek Ph.D. for the great support and help in writing this Bachelor.

Contents

Chapter 1

Introduction

What are the three most important factors in selling real estate? Location, location, and location. This is applicable not only in real estate but also in retail business [1]. Deciding where to locate business has always been a problem that people continuously try to solve worldwide. Throughout the time, most retailers would make a decision based on personal experience and instinct, regarding the process very much as an "art". People would mainly use very subjective techniques. Some of them are no more than "hunches" based upon experience [10].

In the retail environment, businesses are surrounded by an enormous amount of data and variables that can influence their success. As information systems evolved, research procedures became more sophisticated. For retailers, this presented a challenge: without using location decision procedures to improve objectivity, they risked falling behind businesses that adopted such methodologies [10]. Retailers must carefully select and coordinate these tools to ensure they complement each other and provide a comprehensive view of the decision at hand. Otherwise, they risk making false decisions or mistakes.

A solution is to build a system to aid retailers in making informed location decisions. Such a system could utilize one of the procedures, which are designed to assist retailers in the decision-making process, particularly in identifying optimal business locations. This thesis adopts one notable methodology outlined in the journal [16]. This procedure enables users to analyze multiple datasets, utilize GIS features for location selection, and input their preferences into the system, which makes the procedure flexible and suitable for every retailer who chooses to utilize it [16]. Such a system should minimize the amount of work that needs to be done by retailers to analyse and locate the best possible site based on the provided data in any region.

Chapter 2 dives deep into the theory of retail site location assessment procedure. It is necessary to have a clear understanding in order to build a solid system. Chapter 3 introduces *Geographic Information System* (GIS), its methods and the technologies behind it because the final solution partially implements them. Chapter 4 describes target audience for the system, its use cases and functional requirements. It also dives into existing solutions that are available on the internet. Chapter 5 focuses on design, outlining the visual and functional aspects of the system based on the requirements established in chapter 4. Chapter 6 dives into the implementation of the system, detailing the technologies employed, design patterns utilized, and the steps taken to achieve the desired outcome. Chapter 7 outlines the testing procedures applied on the system, strategies for mitigating performance issues, and guidance on utilizing the system with real-world data.

Chapter 2

Retail Site Decision Process

The retail sector is currently in a phase of transformation due to various factors, including the rise in consumer mobility, the prevalence of e-commerce, shifting household demographics, and market saturation. With that, there were evolving business strategies to align with emerging trends. Since the late 1970s, there has been a noticeable decline in the growth of hypermarkets, with small and medium-sized supermarkets gaining preference. This shift in consumer behaviour has forced small retailers to adopt more strategic approaches when selecting store locations, determining store size, and offering services [14].

This section aims to conduct a concise analysis of the algorithms and methodologies mentioned in research [16]. It aims to examine their strengths and weaknesses and explore alternative options available on a global scale. The section is structured so that each subsection explains a specific step within the research (Figure 2.1).

The methodology implies two critical concepts based on spatial dispersion: *Geo-demand* and *Geo-competition*. Geo-demand is the location of potential customers. Geo-competition is the location of business competitors [3]. Each concept is going to contain data points that can be outlined on a map within separate layers. First layer will contain density of the customers on a map, the second layer should contain estimated trading areas of the competitors [16]. Once these two layers are identified, the third layer can be obtained by their joint analysis [3]. The third layer should reveal areas where commercial service is poor, and population density is high. These areas are then considered good for outlets—at this point, a retailer can determine potential sites for his business.

In the next step decision-maker must provide attributes for each potential location, which he can then compare against each other using 1-9 scales, for instance, attribute A is 9 times more important for him than attribute B.

Once it is done, *Analytic Hierarchy Process* (AHP) is applied in order to evaluate all the attributes on each site and output locations with their rating. The location that contains the greatest value of the rating is considered to be the most desired. The consistency of the output fully depends on the user [16].

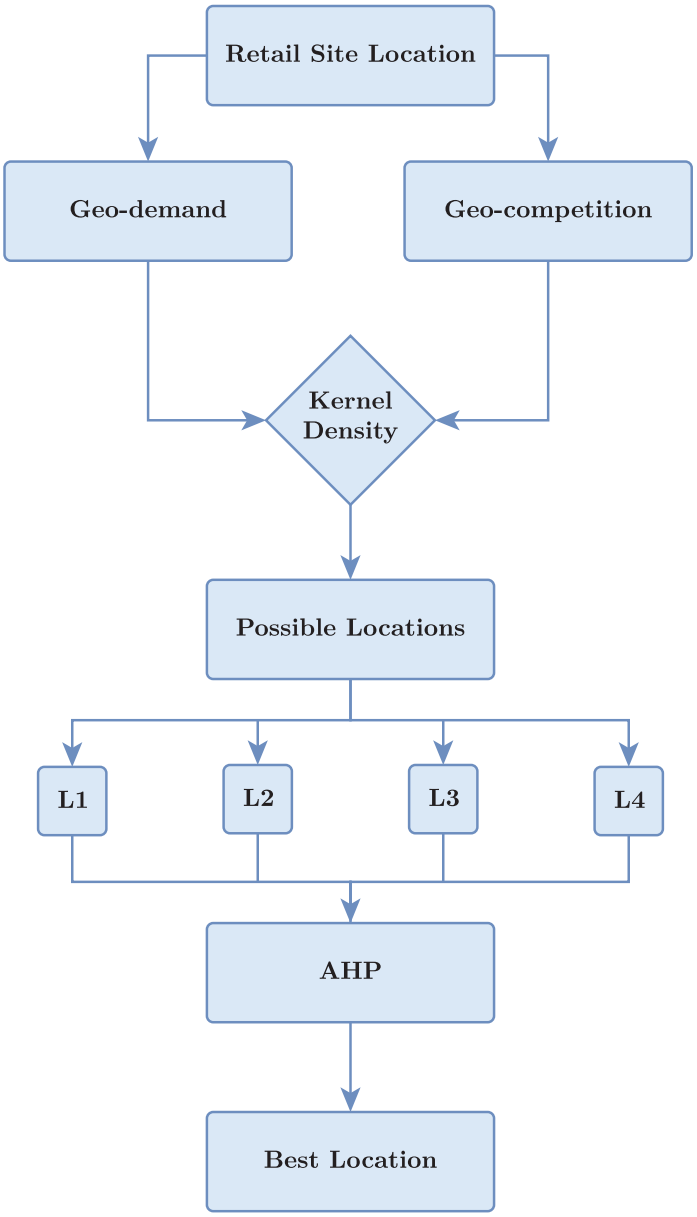


Figure 2.1: Process flow diagram (Adapted from [16])

2.1 Identifying Geo-demand

Geo-demand can be defined as the location of potential customers who purchase a product or service in a specific market (Figure 2.2). According to [16], individuals who live in the area can be viewed as potential customers. This is due to the possibility that individuals might express interest in various markets without certainty regarding their preferences. The data for geo-demand can be acquired from the local city database.

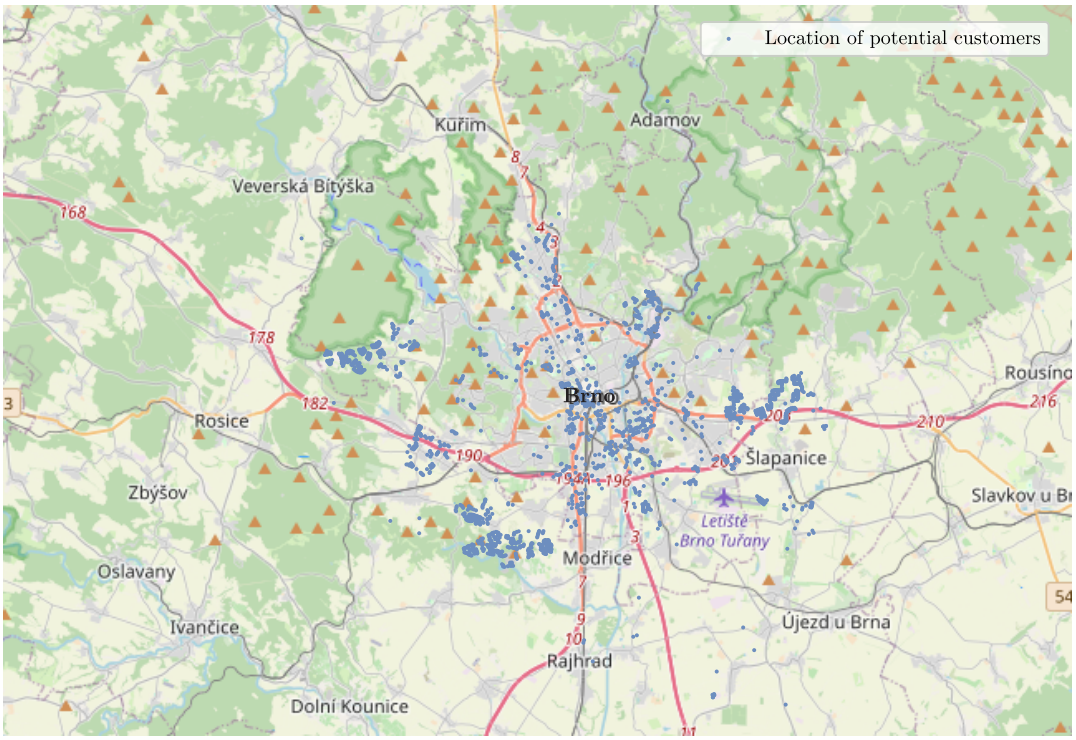


Figure 2.2: Example of geo-demand in Brno. Each blue point represents the location of potential customers.

2.2 Identifying Geo-competition

Geo-competition is the location of the competitors of a business and the delineation of their trade areas in a particular market. Trade area can be defined as the geographic area in which a retailer attracts customers [3, 16].

Geo-competition is more complicated than the previous step [16]. In the last decade, people have created tons of procedures and algorithms that can help to identify trading areas. The procedure for this thesis targets theoretical method, specifically the probabilistic model invented by David L. Huff in 1964 to identify geo-competition, but for completeness of analysis, this section will briefly describe the development of theoretical methods to understand their strengths, weaknesses and challenges [16].

2.2.1 Development of Theoretical Methods

Theoretical methods for defining trading areas involve developing conceptual frameworks or models that explain the spatial interactions and patterns associated with trade or business activities. These methods are based on theoretical principles about how certain factors influence the formation of trading areas [14].

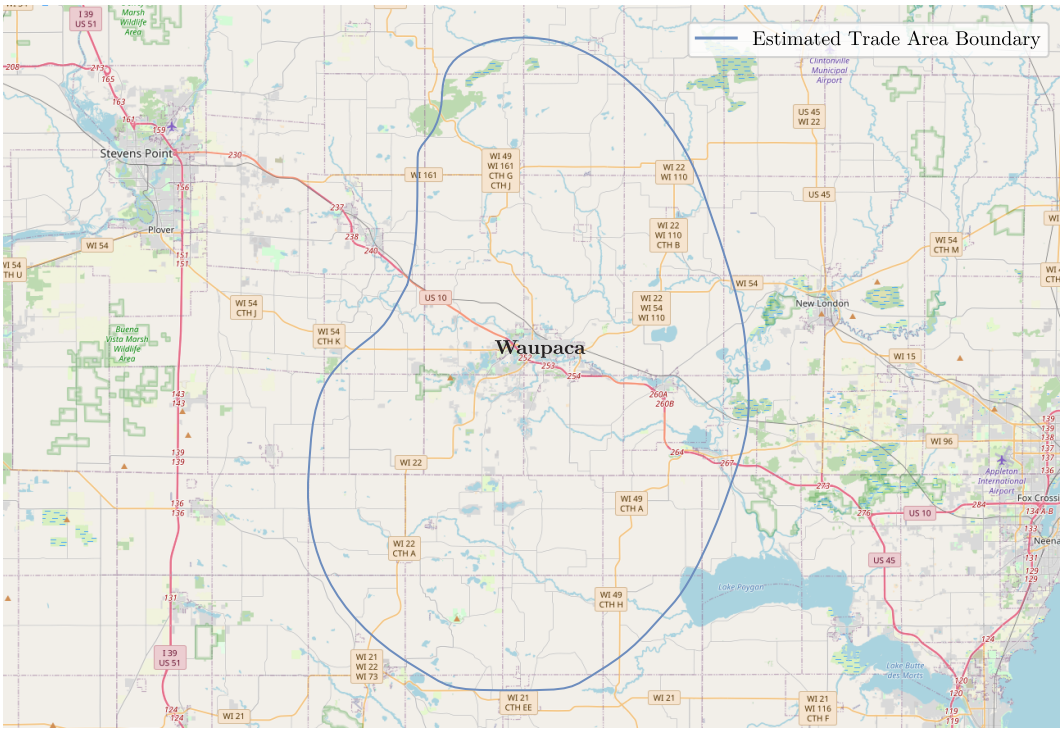


Figure 2.3: This is a map of Waupaca, Wisconsin, USA. The map shows the location of a town, Waupaca, along with surrounding communities. Based on community populations and their distribution, it is possible to draw a simple trade area using the concepts of Reilly's Law (Adopted from [15])

- **Gravity Models**—theoretical frameworks borrowed from physics. They are derived from the laws of Newtonian physics, based on the balance between the store attractivity and the distance to the potential customers [14].
- **Probabilistic Models**—models that are based on the likelihood of customers visiting a certain location within a specific geographical region [12].

J. Reilly's Model

One of the first studies that developed theoretical methods was done by William J. Reilly who formalized a number of empirical observations concerning consumer shopping movements between cities [12].

The idea is that people prefer shopping in bigger communities, but the distance and time it takes to get there affect their willingness to do so in a particular city. In simpler terms, people tend to choose shorter travel distances when they can. Moreover, larger communities are more appealing to shoppers because they usually have a greater variety of products and services (Figure 2.3) [24].

The retail gravitation model has the following structure:

$$\frac{B_a}{B_b} = \left(\frac{P_a}{P_b}\right)\left(\frac{D_b}{D_a}\right)^2 \tag{2.1}$$

- B_a = the proportion of the retail business from an intermediate town attracted by city A (Size or buying power of City A's trade area)
- B_b = the proportion of the retail business from an intermediate town attracted by city B (Size or buying power of City B's trade area)
- P_a = the population of city A
- P_b = the population of city B
- D_a = the distance from the intermediate town to city A (Euclidean distance)
- D_b = the distance from the intermediate town to city B (Euclidean distance)

P. D. Converse's Model

P. D. Converse made a significant modification of Reilly's original formula which makes it possible to calculate the approximate point between two competing cities where the trading influence of each was equal [12]. This means it is possible to outline a city's retail trading area by calculating and connecting the breaking points between that city and each of its competitors in the region (Figure 2.4) [12].

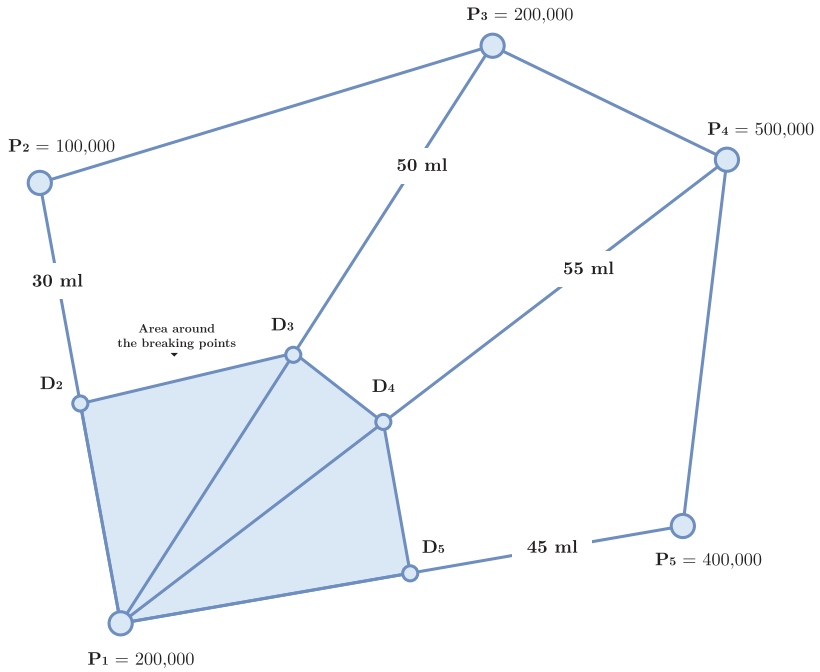


Figure 2.4: Example of calculated breaking points D (D_1 - D_5) between the cities P (P_1 - P_5) by David L. Huff 2.3 (Adapted from [12])

The breaking point formula derived by Converse:

$$D_b = \frac{D_{ab}}{1 + \sqrt{\frac{P_a}{P_b}}} \tag{2.2}$$

- D_b = the breaking point between city A and city B in miles from B
- D_{ab} = the distance separating city A from city B (Euclidean distance)
- P_a = the population of city A
- P_b = the population of city B

Calculation of breaking points [12]:

$$\begin{aligned}
D_2 &= \frac{D_{12}}{1 + \sqrt{\frac{P_1}{P_2}}} = \frac{30}{1 + \sqrt{\frac{200000}{100000}}} = 12.4 \text{ ml} \\
D_3 &= \frac{D_{13}}{1 + \sqrt{\frac{P_1}{P_3}}} = \frac{50}{1 + \sqrt{\frac{200000}{200000}}} = 25 \text{ ml} \\
D_4 &= \frac{D_{14}}{1 + \sqrt{\frac{P_1}{P_4}}} = \frac{55}{1 + \sqrt{\frac{200000}{500000}}} = 33 \text{ ml} \\
D_5 &= \frac{D_{15}}{1 + \sqrt{\frac{P_1}{P_5}}} = \frac{45}{1 + \sqrt{\frac{200000}{400000}}} = 26.3 \text{ ml}
\end{aligned} \tag{2.3}$$

Many analysts, including David L. Huff, noted 3 important issues with P. D. Converse's formula. The first issue was that it was impossible to estimate the area around the breaking point (Figure 2.4). Calculating the area around breaking points (points D) poses a challenge due to the dynamic nature of trade zones. The problem lies in determining precise boundaries for retail trading areas where the influence of competing cities is equal.

The second problem arises, as pointed out by Huff when defining the retail trading areas for multiple shopping zones within a specific geographical region. The resulting overlapping boundaries are inconsistent with the main goal of the formula (Figure 2.5) [12].

The third issue is that the formula does not include any parameter that would indicate the type of shopping trip (grocery, furniture, etc). So many analysts have assumed that it is logical that such an exponent will vary, depending on the type of shopping trip [12].

Huff Model

As a solution to the problems described in 2.2.1, Huff presented a probabilistic model:

$$P_{ij} = \frac{\frac{S_j}{T_{ij}^{\lambda}}}{\sum_{j=1}^{n} \frac{S_j}{T_{ij}^{\lambda}}} \tag{2.4}$$

- P_{ij} = the probability of a consumer at a given point of origin i travelling to a particular shopping centre j
- S_j = the size of a shopping centre j (measured in terms of the square footage of selling area)

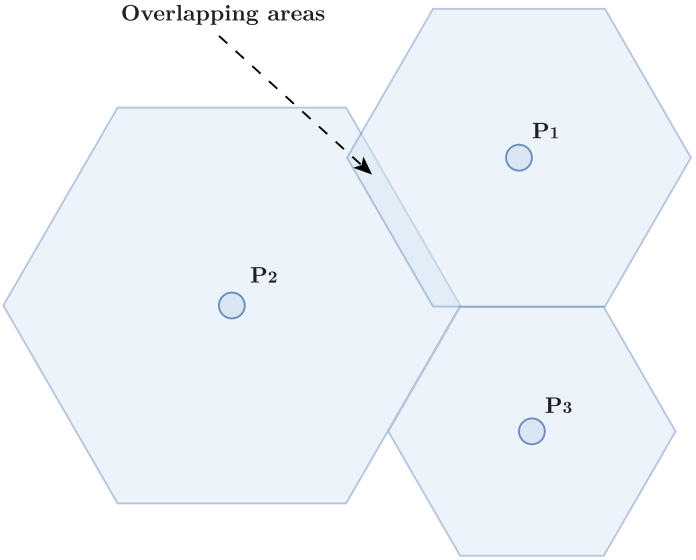


Figure 2.5: Example of overlapping areas problem. Let's consider more than two cities: P_1, P_2 and P_3. The problem becomes visible when calculating trading area boundaries between more than two cities [12] (Adopted from [12]).

- T_{ij} = the travel time involved in getting from a consumer's travel base i to a given shopping centre j
- λ = a parameter which is to be estimated empirically to reflect the effect of travel time on various kinds of shopping trips [12]. In simple terms, the distance decay factor in Huff's model represents how the influence or attractiveness of a retail location decreases as a customer moves farther away from it.

Huff defined a trading area in his research as a "geographically delineated region, containing potential customers for whom there exists a probability greater than zero of their purchasing a given class of products or services offered for sale by a particular firm or by a particular agglomeration of firms" (Figure 2.6). He stated that when consumers face multiple similar options, they often find it challenging to pick just one. The choices are so similar that it's hard to distinguish between them. As a result, consumers might end up choosing somewhat randomly, relying on their instincts [13]. Secondly, a consumer is uncertain whether the expected store will fulfil his shopping expectations [13].

When it comes to evaluating locations using this model it is important to note that it is heavily dependent on data. Its quality plays a huge role in predicting trading areas accurately. The retailer would need not only the area of the stores in the region but also to conduct a survey in order to properly estimate parameter λ, which is called distance-decay factor, and its purpose is to correct the resulting trading area [12]. For instance, based on Huff's distance-decay factor calculated in 1964, we can assume that customers were willing to travel further in order to buy furniture and travel less to do trips that involved clothing purchases, therefore it is important to keep this parameter up-to-date because in different cities, categories or time intervals it may change [12]. However, even though the formula is considered to be quite complete, many analysts claim that the equation provided by Huff is limited due to the fact that it utilizes the area of the store only as a variable which has not been found to have a great influence on drawing power, it is rather an explaining factor

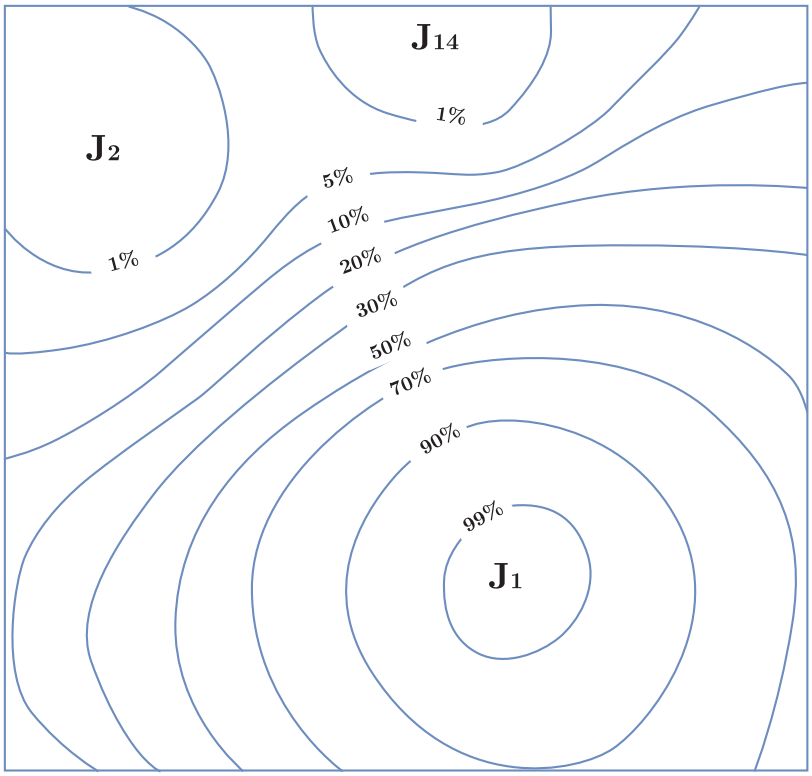


Figure 2.6: A depiction of a retail trade area using probability contours P_{ij}. J_1, J_2, J_{14} represent stores (Adapted from [12]).

of the drawing power [20]. It is worth noting that Huff had explained that "mathematical models are not infallible. They are, by necessity, simplified constructs of some aspect of reality. It is impossible for such constructs to include all the possible factors that may have a bearing on a particular problem. Therefore decision-makers should be aware that there are variables other than those specified in the model that affect the sales of a retail firm", therefore as a consequence **human judgement plays an important role** as well [13].

2.2.2 Huff Model Calibration

The Huff model introduces a crucial parameter known as the "distance decay parameter" (λ), which represents the rate at which the probability of choosing a destination decreases with increasing distance. Usually, this parameter varies from 1.5 to 2 [11]. Estimating λ is called "Huff model calibration" [12].

The calibration of the Huff Model involves the estimation of model parameters to accurately reflect the observed consumer behaviour within a study area. The process typically relies on real-world data obtained through household surveys, which capture actual shopping preferences of residents in various subareas. The objective is to determine the frequency with which residents patronize different stores within each subarea. The calibration process is essential for the model to generate meaningful estimates that align with observed patronage data [11].

The process of gathering the required data for model calibration involves a series of steps outlined below [11]:

- Delineate the study area.
- Divide the study area into subareas.
- Conduct a survey of households within each subarea to determine the frequency at which consumers patronize stores within the study area and apply *Ordinary Least Squares* (OLS) regression to estimate model parameters [11].

To apply survey data and estimate the parameters of the Huff Model, a regression analysis is performed. The Huff Model is transformed using a log-centering approach, rendering it linear in its parameters [11].

The general form of the Huff Model can be defined as follows [11]:

$$P_{ij} = \frac{(\prod_{h=1}^{H} A_{hj}^{\gamma_h}) D_{ij}^{\lambda}}{\sum_{j=1}^{n} (\prod_{h=1}^{H} A_{hj}^{\gamma_h}) D_{ij}^{\lambda}} \tag{2.5}$$

- P_{ij} = Probability that a consumer in geographic area i visits facility j.
- A_{hj} = Measure of the h_{th} characteristic ($h = 1, 2, ..., H$) reflecting the attraction of facility j.
- γ = Parameter representing the sensitivity of P_{ij} associated with the attraction variable h.
- D_{ij}^{λ} = Measure of accessibility of facility j to a consumer located at i.
- λ = Parameter representing the sensitivity of P_{ij} with respect to accessibility.
- n = Number of facilities.

This model can be transformed into a linear form in the parameters by applying the following transformation [11]:

$$\log(\frac{P_{ij}}{\tilde{P}_i}) = \sum_{h=1}^{H} \gamma^h \log(\frac{A_{hj}}{\tilde{A}_j}) + \gamma \log(\frac{D_{ij}}{\tilde{D}_i}) \tag{2.6}$$

$\tilde{P}_i$, $\tilde{A}_j$, $\tilde{D}_i$ are geometric means of P_{ij}, A_{hj} and D_{ij}.

Delineating the Study Area

The initial and crucial step in obtaining the right data is to clearly define the study area. This is extremely important because the size and boundaries of the study area influence the type of data collected and the conclusions drawn from the analysis. The study area can be seen as an „island“ where buyers and sellers interact for exchanges. Ideally, most transactions should happen within this defined area. There should be minimal trade from people outside the area, and residents within the area should engage in limited trade outside of it [11].

Dividing the Study Area into Subareas

After establishing the study area, it's important to divide it into smaller sections to reduce geographic distortions (Figure 2.7). One effective method is to create a grid with cells of equal size, allowing for the collection of census data within each cell [11].

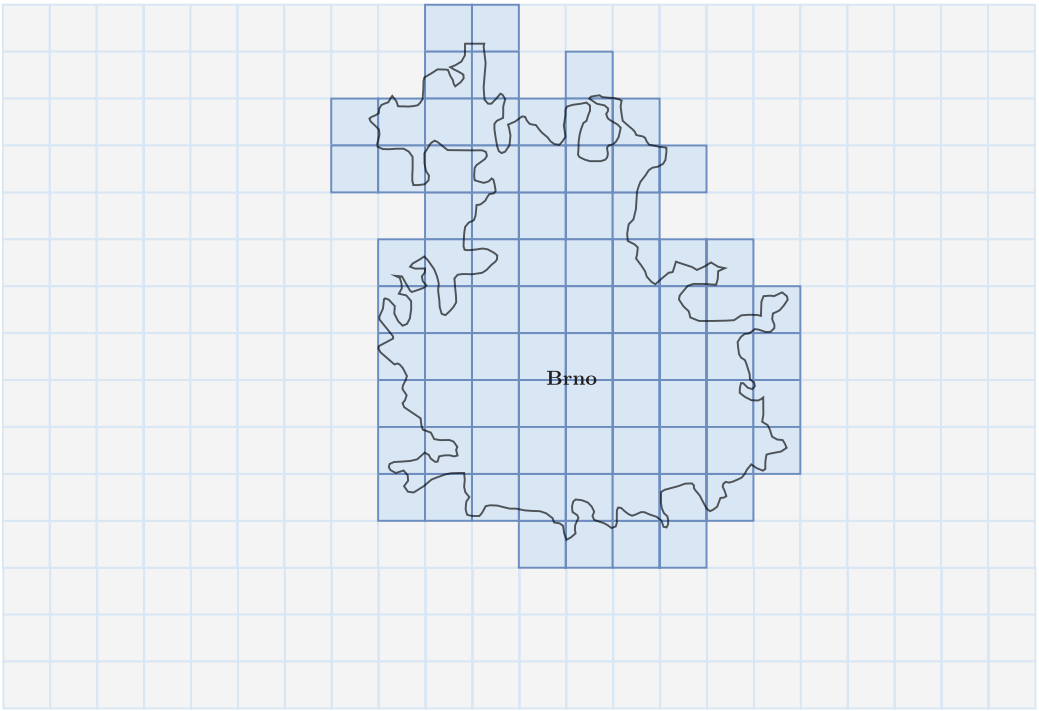


Figure 2.7: Subdivided study area. As the study area, a 15-minute driving time was used. Highlighted sections are the areas of interest (Adapted from [11]).

Household Surveys

Residents are chosen at random from various subareas, with surveys conducted through telephone or personal interviews. The survey aims to delve into consumer preferences, covering aspects such as where residents purchase food for home preparation, the frequency of visits to specific stores within a set number of shopping trips, individual preferences for each store, and the residents' awareness of promotional activities or marketing programs run by the stores [11].

Typical questions that are often asked of respondents are listed below:

- At which stores do you normally purchase food items prepared at home?
- Out of 10 shopping trips, how often do you go to each store?
- What do you particularly like about each of these stores?
- Are you aware of any promotional activities by any of these stores or other marketing programs?

The collected data may require transformation to make it suitable for regression analysis. This includes creating variables that are appropriate for the Huff Model, such as the logarithm of shopping frequency and other variables for categorical responses (e.g., store preferences). The transformed data should accurately represent the variables used in the model [11].

The linear form of the Huff Model (Equation 2.6) is used in a regression framework to estimate its parameters. The purpose of using linear regression is to find the coefficients that best fit the observed data [11].

Linear Regression

Linear regression is a statistical method used to model the relationship between a dependent variable (also called the response or outcome variable) and one or more independent variables (predictors or explanatory variables). It assumes a linear relationship between the independent variables and the dependent variable [9].

In the context of the Huff Model calibration, linear regression serves as a powerful tool to estimate the model's parameters, providing insights into the factors influencing consumer choices in retail settings [11].

The general form of a simple linear regression equation with one independent variable is [9]:

$$y = \beta_1 x_1 + ... + \beta_p x_p \tag{2.7}$$

The goal of linear regression is to estimate the values of $\beta_0 ... \beta_p$ that minimize the sum of squared differences between the observed and predicted values of y. This process is often referred to as "ordinary least squares" (OLS) regression [9].

Example of a simple linear regression (Figure 2.8):

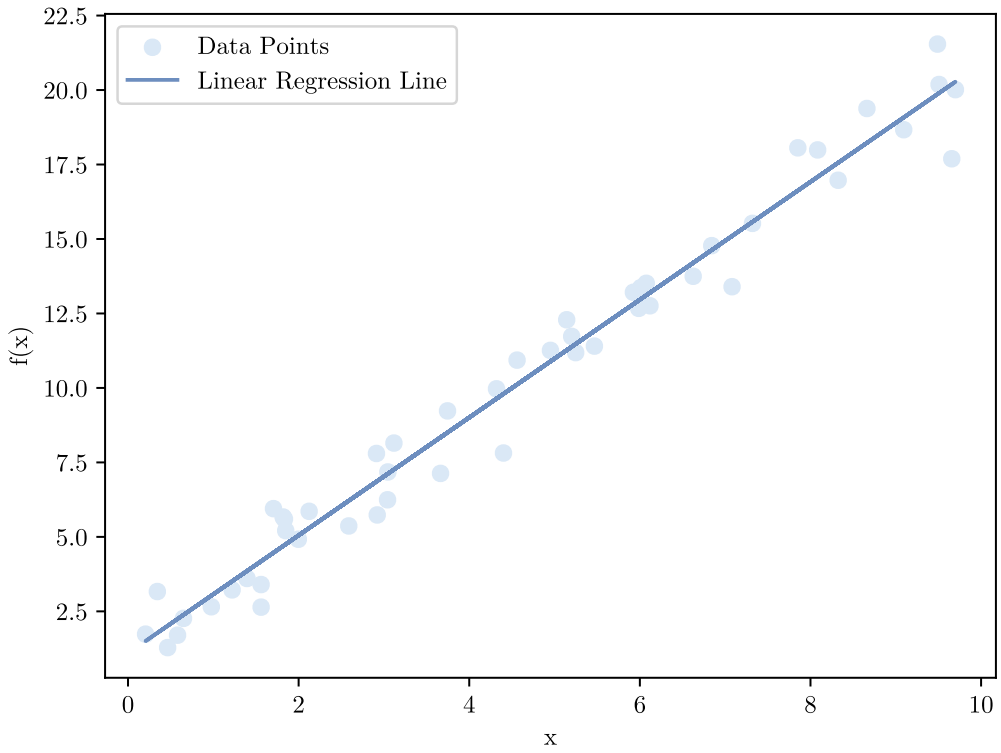


Figure 2.8: A collection of data points that represent the relationship between our dependent variable and one independent variable. The goal of this regression model is to find a line, represented by the function f, that best fits the pattern formed by these points [9].

If the data points are clustered around a straight line, it suggests that a linear relationship might be a good approximation for our data. This alignment indicates that linear regression could be a suitable method to model the association between these variables [9].

2.2.3 Improved Huff-based Models

This section will present examples of enhanced Huff-based models. Even though this thesis follows Huff model, it is still crucial to gain insights into potential enhancements or motivation for improving the system and understand the challenges.

Many analysts have attempted to address limitations in the model by introducing additional variables that influence the trading area [13].

Image Inputs to a Probabilistic Model

One of the studies made in this direction by Thomas J. Stanley and Murphy A. Sewall who assumed that customer's perception of the stores is a multidimensional phenomenon, therefore it is important to include more factors that affect trading area in the equation [20].

As a foundation, Thomas J. Stanley laid the research made in 1971 by economist Philip Kotler, who came up with the term "image factor", or more commonly "brand image", which he used in order to describe shoppers' preference for visiting a specific store. According to him, this particular behaviour was affected by the image, which is made up of such elements as the brand's reputation, values, personality, and associations [20]. These factors can be represented using extra parameters like the distance-decay factor in the original model. They can also be estimated by conducting an extra survey [20].

The enhanced model with image parameters has the following structure:

$$P_{ij} = \frac{\frac{S_j^{\lambda_s} \cdot D_{ij}^{\lambda_D}}{T_{ij}^{\lambda_t}}}{\sum_{j=1}^{n} \frac{S_j^{\lambda_s} \cdot D_{ij}^{\lambda_D}}{T_{ij}^{\lambda_t}}} \tag{2.8}$$

- P_{ij} = the probability of a consumer at a given point of origin i travelling to a particular shopping centre j
- S_j the size of a shopping centre j (measured in terms of the square footage of the selling area)
- λ_S = The sensitivity of changes in shopping probability to changes in selling area
- T_{ij} = the travel time involved in getting from a consumer's travel base i to a given shopping centre j
- λ_t = The sensitivity of changes in shopping probability to changes in travel time
- D_{ij} = The measure of an image between an "ideal" supermarket chain for consumers in area i and the chain represented in the market area by supermarket j
- λ_D = The sensitivity of changes in shopping probability to changes in store image

As a last step, he compared the output of the original formula and concluded that based on its result, the modified formula appears to be more accurate in terms of reality, but **less practical due to the necessity of an extra survey** [20]. The extra survey itself is not an issue in the context of a model because it was quite common for the retailers to conduct them, but rather an extra layer in an already complex formula.

In addition, authors of the next model enhancement from the section 2.2.3 claimed that these surveys are labour-intensive and time-consuming in terms of a business [23].

Use of Social Media Data

Social media data present fresh possibilities for gaining insights into consumer behaviour and outlining user territories. Obtaining this data is often easier compared to surveys, as it reflects the actions of a large number of users over extended periods. However, it's crucial to acknowledge that social media data has limitations—it only captures the activities users share online and doesn't fully represent all their real-world actions [23].

The study focused on the social app "Sina Weibo" launched in 2009, which is among the largest social media platforms in China. Similar to Twitter, it enables users to share their location at interesting places. This location data can be utilized to define real-world trade areas [23].

A model that includes sensitive parameters, which are estimated using social media data, has the following structure:

$$P_{ij} = \frac{\frac{A_j^{\alpha}}{D_{ij}^{\lambda}}}{\sum_{j=1}^{n} \frac{A_{ij}^{\alpha}}{D_j^{\lambda}}} \tag{2.9}$$

- A_j = the attractiveness of facility or retail agglomeration j
- D_{ij} = the distance between i and j
- α = sensitive parameter associated with attractiveness A. It determines the impact of the attractiveness of a location on the probability of interaction
- λ = sensitive parameter associated with distance D. It determines the impact of distance on the probability of interaction

This method simplifies the retailer's task by eliminating the necessity of conducting surveys to determine the distance-decay factor [23]. However, the availability of data is crucial and, unfortunately, the necessary data for implementing this modified method is currently lacking in many countries, particularly in Europe.

2.3 Determining The Possible Locations

The third step of the process is to match information resulting from the analysis of geo-demand and geo-competition in order to obtain the area where a population has a poor range of commercial services. In order to achieve this *Kernel Density Estimation* can be applied [16].

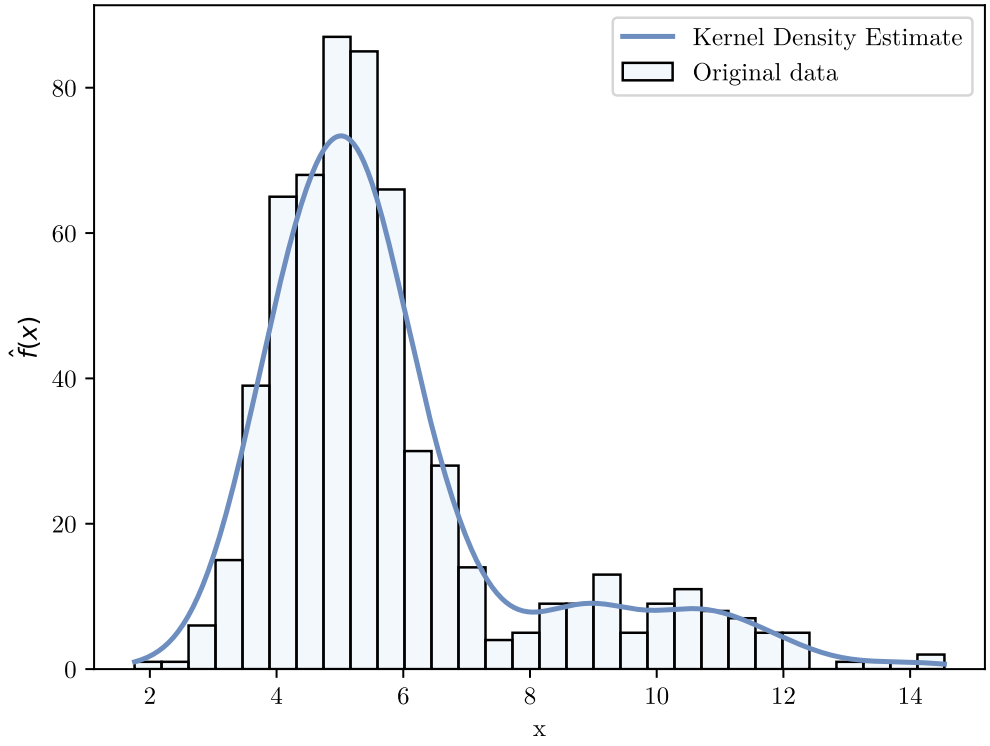


Figure 2.9: Example of kernel density estimation $\hat{f}(x)$ of random data. As $K(x,t)$ was used Gaussian Kernel (Adapted from [25]).

2.3.1 Kernel Density Estimation

Kernel density estimation (KDE) is a non-parametric technique for density estimation in which a known density function (the kernel) is averaged across the observed data points to create a smooth approximation (Figure 2.9) [25].

$$\hat{f}(t) = \frac{1}{n}\sum_{i=1}^{n} K(x_i, t) \tag{2.10}$$

- $\hat{f}(x)$ = Kernel estimate of original unknown probability distribution function $f(x)$
- x_i = An independent and identically distributed sample of unknown probability distribution function $f(x)$
- n = Number of observations
- $K(x_i, t)$ = Kernel function, such as Gaussian Kernel (Normal Distribution),

In essence, the objective of kernel density estimation is to compute the density of points within a specified area based on the distances between the points. This calculation is contingent on the assumption that all points carry equal weight. Nevertheless, it is possible

to use distinct weights to individual points, enabling the prioritization of specific points over others [16].

According to [16], the pixel was adopted as a unit of analysis. A pixel is a square on a digital map that represents a specific area and is assigned a value linked to the features within that space. The process of subdividing the map into those pixels was described in section 2.2.2.

For each square on the map, a circular area is created using the square's centre as the circle's centre. The data points within this circle can have different weights to these, considering their distance from the centre of the square. Simply put, points closer to the centre have more influence, while those farther away carry less weight. This concept can be expressed as follows [16]:

$$L_j = \sum_{i \in C_j} \frac{3}{\pi r^2} \left(1 - \frac{d_{ij}^2}{r^2}\right)^2 \tag{2.11}$$

- L_j = Estimated density of a pixel
- d_{ij} = Distance between points i and j
- r = Width of the window or search radius, which determines the degree of smoothing
- C_j = Set consists of the i points whose distances from the centroid of a pixel are less than the established radius of the circle ($d_{ij} < r$)

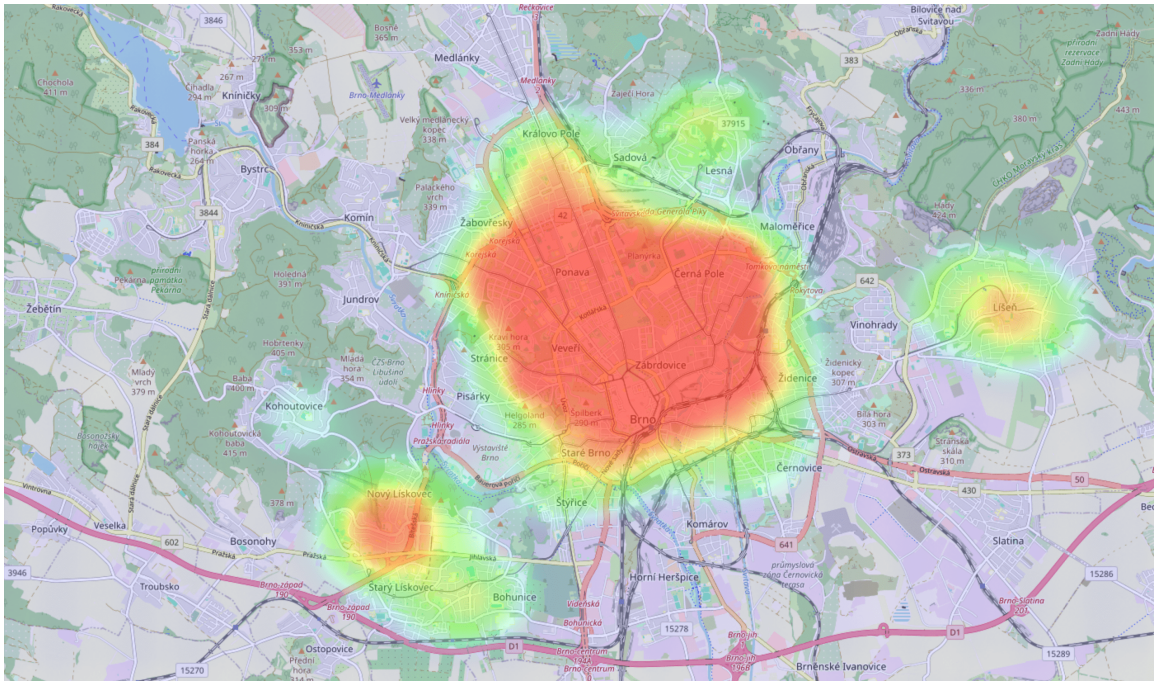


Figure 2.10: Example of kernel density estimation on map. Hot spots represent areas with potential customers and a high range of commercial services. This figure was made using a dataset with population and the first 35 candy stores from the dataset with retail outlets of Brno.

2.3.2 Selecting Possible Locations

To identify optimal retail locations in areas with a lack of services, businesses often employ advanced mapping techniques that integrate demographic data and service coverage [16]. A crucial aspect is to create heatmaps using kernel density, which provides a visual representation of customer density and service gaps across different regions.

Once the heatmap is estimated, it becomes possible to pinpoint regions with customers and a poor range of services [16] (Figure 2.11).

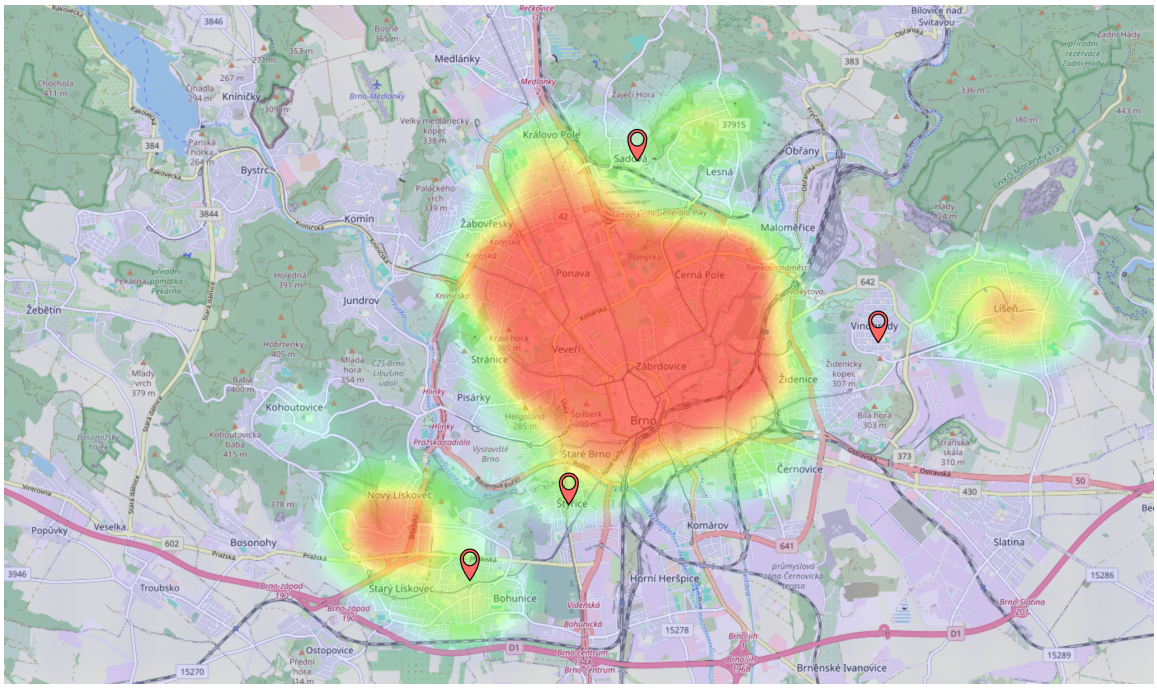


Figure 2.11: Example of selecting possible locations after the kernel density estimation. Here, the same estimated kernel density was used from figure 2.10.

2.4 Multi-criteria Decision Analysis

Areas with higher concentrations of potential clients were identified through the kernel density analysis, the next step involves evaluating and rating potential sites based on their geographic and physical characteristics. This rating process aims to prioritize locations that contain favourable attributes for establishing new commercial establishments—this is where multi-criteria decision analysis comes into play [16].

Multi-Criteria Decision Analysis (MCDA) is a systematic approach employed in decision-making processes that involve the evaluation of multiple alternatives. This analytical method addresses the complexity of decision problems by considering various criteria simultaneously, providing a structured framework for assessing and comparing potential options [5].

In many decision-making scenarios, particularly those involving site selection, decision-makers are faced with multiple factors that influence the outcome, such as sales floor area, accessibility, potential market, and distance to competition [16]. These factors are often referred to as criteria.

In order to help the retailer evaluate multiple criteria, there is a variety of methods, such as PROMTHEE, ELECTRE, TOPSIS, AHP or Hybrid methodologies [26]. According to [16], the AHP method was selected.

2.4.1 Analytic Hierarchy Process

The Analytic Hierarchy Process (AHP) is considered to be the most popular among the others [22]. The method was proposed by Saaty as a pairwise comparison-based methodology [17].

Hierarchical Structure of AHP

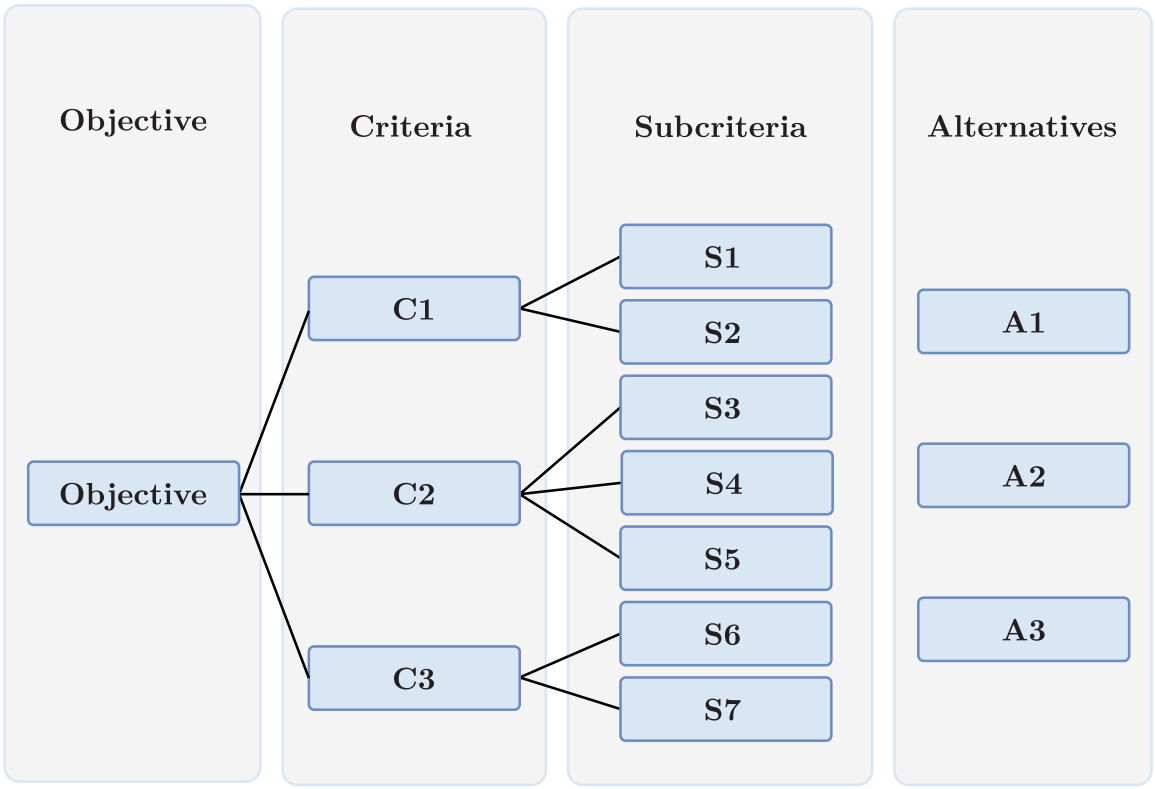


Figure 2.12: Hierarchical model (Adopted from [16])

What all multi-criteria methods have in common is that most decisions can be improved by decomposing the overall evaluation of alternatives into evaluations on a number of criteria relevant to the problem [5]. The AHP method takes advantage of that by structuring decision problems into a comprehensive hierarchy which consists of several levels [22] (Figure 2.12).

The first level of the hierarchy is a goal of the process, sub-levels are constructed of criteria that can affect the choice. The last level of the hierarchy contains the alternatives, which are, at the end of the process, assessed.

Application

To illustrate this method, let's consider the following example.

Suppose there are three projects: Project A, Project B and Project C. Using an analytical hierarchical process it is possible to identify the relative priority of each project. The goal is the project. Let's assume there are three criteria that drive the choice of project: duration, cost, and expected quality (In reality, there may be many more such criteria).

Let's use a 1-9 scale to compare criteria, define the significance of other attributes and put it into the table (Table 2.1):

	Duration	Cost	Quality
Duration	1	0.333	0.200
Cost	3	1	0.333
Quality	5	3	1

Table 2.1: Criteria that are compared in pairs.

Now let's calculate the sum of each column and divide the value of each cell by the sum of the values of the corresponding column (Table 2.2).

	Duration	Cost	Quality
Duration	0.111	0.077	0.130
Cost	0.333	0.231	0.217
Quality	0.556	0.692	0.652

Table 2.2: Each value from previous table 2.1 divided by sum of each column.

By calculating the average values of the rows, it is possible to find the specific weight of each of the criteria (2.3).

Duration	Cost	Quality
0.106	0.261	0.633

Table 2.3: Weights of each of the criteria.

Let's assume that each of the projects has the following attributes (Table 2.4):

Taking each of the estimates with the specific weight of the criterion found earlier and adding them up in a project-by-project manner, we get:

$$\text{Project A} = 0.106 \cdot 5 + 0.261 \cdot 7 + 0.633 \cdot 3 = 4.256$$

$$\text{Project B} = 0.106 \cdot 7 + 0.261 \cdot 5 + 0.633\dot{3} = 6.054$$

$$\text{Project C} = 0.106 \cdot 3 + 0.261 \cdot 7 + 0.633 \cdot 5 = 4.690$$

Obviously, **Project B** will be selected.

Limitations

The user is able to define his judgements on a 1-9 scale, so rather than prescribing a "correct" decision, the AHP helps decision makers find the decision that best suits their goal and their understanding of the problem. The method faces the disadvantage of uncertainty and inconsistency in judgment and ranking criteria, which means the decision-maker may not know how to judge specific criteria, which leads to another flaw—it is possible for a rank reversal to occur [22]. In order to detect this flaw, it is possible to calculate consistency,

	Project A	Project B	Project C
Duration	5	3	7
Cost	7	5	3
Quality	3	7	5

Table 2.4: Project attributes.

which is estimated using the consistency ratio, which reflects how consistent the judgements are relative to large samples of purely random judgments. If the consistency ratio exceeds 10%, the judgments are considered untrustworthy [18].

Important Criteria for Site Selection Process

According to [16], there are four main criteria: establishment, location, demographic factors and competition (Figure 2.13):

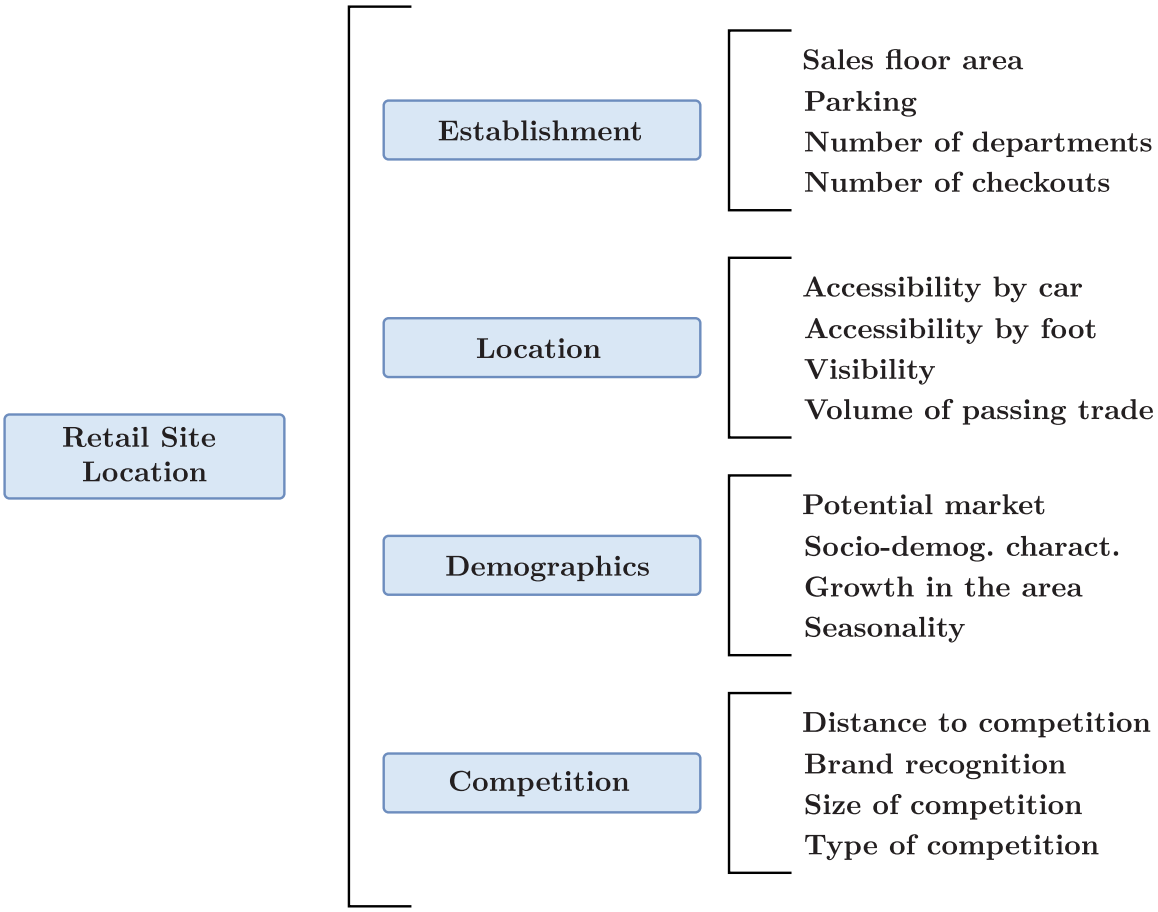


Figure 2.13: Factors that affect the success of a supermarket (Adopted from [16])

1. **Establishment**—refers to the features of the property itself. These features encompass the size of the sales floor area, whether there is parking, the number of product departments, and the available number of checkouts [16].

2. **Location**—covers all aspects related to where the store is situated. This includes how easy it is to reach by car or on foot, its visibility to potential clients, and the amount of passing trade in the surrounding area [16].

3. **Demographics**—is about understanding the characteristics of the people living in the trade area of the new retail site. This involves considering the total population in the trade area, the specific types of clients based on factors like purchasing power and family structure, predictions for growth in the surrounding area, and variations in sales throughout the year [16].

4. **Competition**—focuses on the features of other establishments that offer similar services. This takes into account the distance to competitors, how well-known their brand is in the area, the size of their sales floor area, and the type of commercial strategy they employ [16].

The decision-making process involved a diverse group of retail site location and marketing experts, individuals from both academic and professional backgrounds. Each expert participated in interviews where they assessed and rated various criteria crucial for retail site selection (Figure 2.13). The comparison matrix derived from these evaluations served as a foundation for determining the significance of each criterion. During the analysis, scores with a consistency ratio exceeding 10% were excluded to maintain the reliability of the results [16].

The next step involved extracting eigenvectors associated with each matrix, representing the expert's individual judgments. These eigenvectors were then aggregated using the arithmetic mean technique, resulting in a set of weights associated with each criterion. These weights, reflecting the experts' consensus on the importance of each criterion, were subsequently used to rank the sub-criteria [16].

According to the insights (Table 2.5) gathered from the consulted experts, the critical factors influencing a supermarket's success are primarily the volume of passing trade (17.44%), store visibility (14.62%), proximity to competitors (14.49%), the potential market within the trade area (9.75%), accessibility by car (9.71%), and accessibility by foot (9.17%). The collective impact of these six sub-criteria accounts for more than 75% of a supermarket's success.

Ranking	Criteria	Subcriteria	Score
1	Location (0.509)	Volume of passing trade (0.342)	17.44%
2	Location (0.509)	Visibility (0.287)	14.62%
3	Competition (0.245)	Distance to competition (0.591)	14.49%
4	Demographics (0.188)	Potential market (TA) (0.519)	9.75%
5	Location (0.509)	Accessibility by car (0.191)	9.71%
6	Location (0.509)	Accessibility by foot (0.180)	9.17%
7	Competition (0.245)	Brand recognition (0.227)	5.56%
8	Demographics (0.188)	Seasonality (0.250)	4.69%
9	Establishment (0.057)	Number of departments (0.575)	3.30%
10	Competition (0.245)	Type of competition (0.128)	3.13%
11	Demographics (0.188)	Growth in the area (0.147)	2.75%
12	Demographics (0.188)	Socio-demographic characteristics (0.085)	1.60%
13	Competition (0.245)	Size of competition (0.055)	1.35%
14	Establishment (0.057)	Sales floor area (0.179)	1.03%
15	Establishment (0.057)	Parking (0.154)	0.88%
16	Establishment (0.057)	Number of checkouts (0.092)	0.53%

Table 2.5: Ranking of the subcriteria that determine the success of a supermarket (Adopted from [16])

Chapter 3

Geographical Information Systems

In today's rapidly advancing technological world, the integration of Geographical Information Systems (GIS) has become crucial in various fields, including retail location decision systems. GIS offers a powerful toolset for capturing, analyzing, and visualizing spatial data, providing valuable insights that aid decision-making processes. This section explores the base concepts and applications of GIS [8].

3.1 Geographic Information System

A Geographic Information System (GIS) is an informational system that analyzes and displays geographically referenced information which is attached to a unique location. It is a useful tool for individuals and businesses to grasp how things are arranged in space. It helps compare the locations of various elements to uncover their connections. For instance, on a single map, you could have one layer showing potential retail locations and another layer indicating factors like customer demographics and competitor locations. This kind of map could assist in deciding the best spots for opening new stores, making the process of choosing retail locations more informed and effective [7, 16].

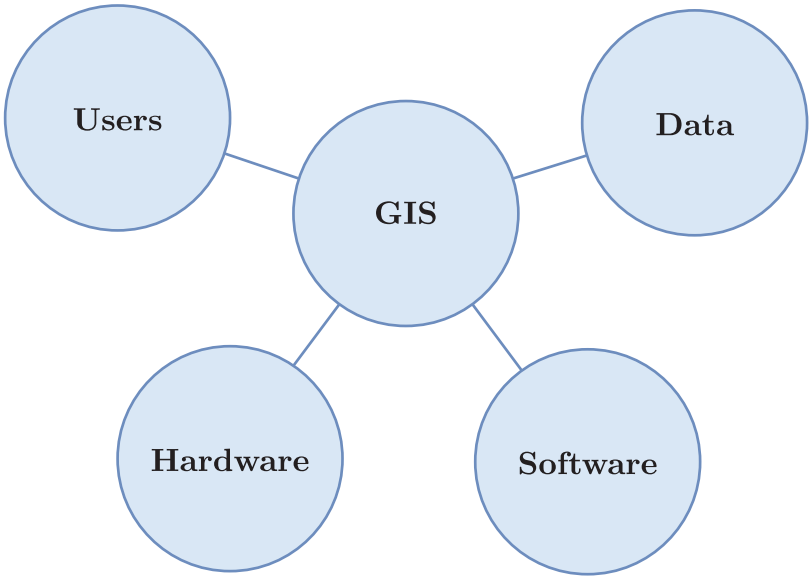


Figure 3.1: A depiction of GIS components [7]

GIS system includes such components as:

- **Hardware**—GIS relies on a variety of hardware components, including computers, servers, and data collection devices. These devices serve as the backbone for processing and managing spatial information [7].
- **Software**—specialized programs form the software component of GIS, providing the tools and interfaces necessary for manipulating, analyzing, and visualizing spatial data. These software applications enable users to perform tasks ranging from simple map creation to complex spatial analyses [7].
- **Data**—heart of any GIS system. Spatial data includes information with a location paired with relevant attributes. The data component forms a comprehensive database of spatial information that serves as the foundation for GIS analysis [7].
- **Users**—GIS involves a diverse set of users with distinct roles. Data processors engage in collecting, inputting, and managing spatial data. GIS managers oversee the system's overall operation and coordinate its use within an organization. Recipients of spatial information, which could be decision-makers or end-users, benefit from the insights and visualizations provided by GIS to inform their decision-making processes [7].

3.2 Data Formats

GIS applications may include cartographic data, photographic data, digital data, or data in spreadsheets [7] (Figure 3.2).

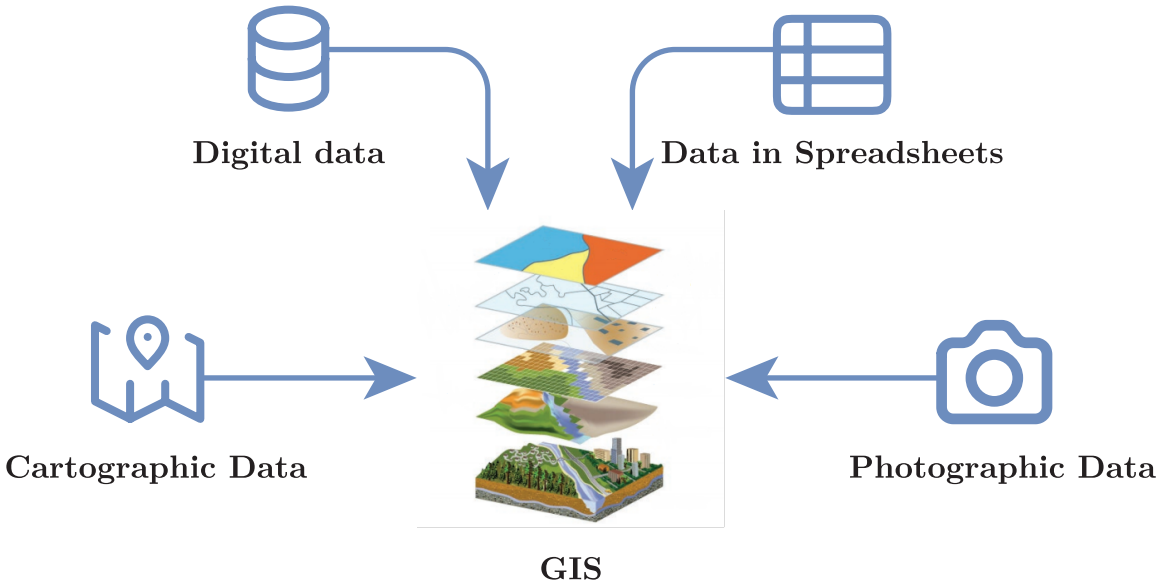


Figure 3.2: A depiction of GIS data formats [7]

- **Cartographic Data**—refers to information typically found on maps. It includes features such as roads, rivers, political boundaries, and other geographic elements represented in a graphical form.

- **Photographic Data**—involves images captured by various sources, including satellite imagery, aerial photography, or ground-based photographs. These images provide a visual representation of the Earth's surface.
- **Digital Data**—refers to information stored in a digital format, which can include vector data (points, lines, polygons) and raster data (grids of pixels representing surfaces).
- **Data in Spreadsheets**—includes information in table form, which is in rows and columns.

3.2.1 Spatial Relationships

GIS technology serves as a powerful tool for visualizing spatial relationships and depicting linear networks, enhancing understanding of geographical features.

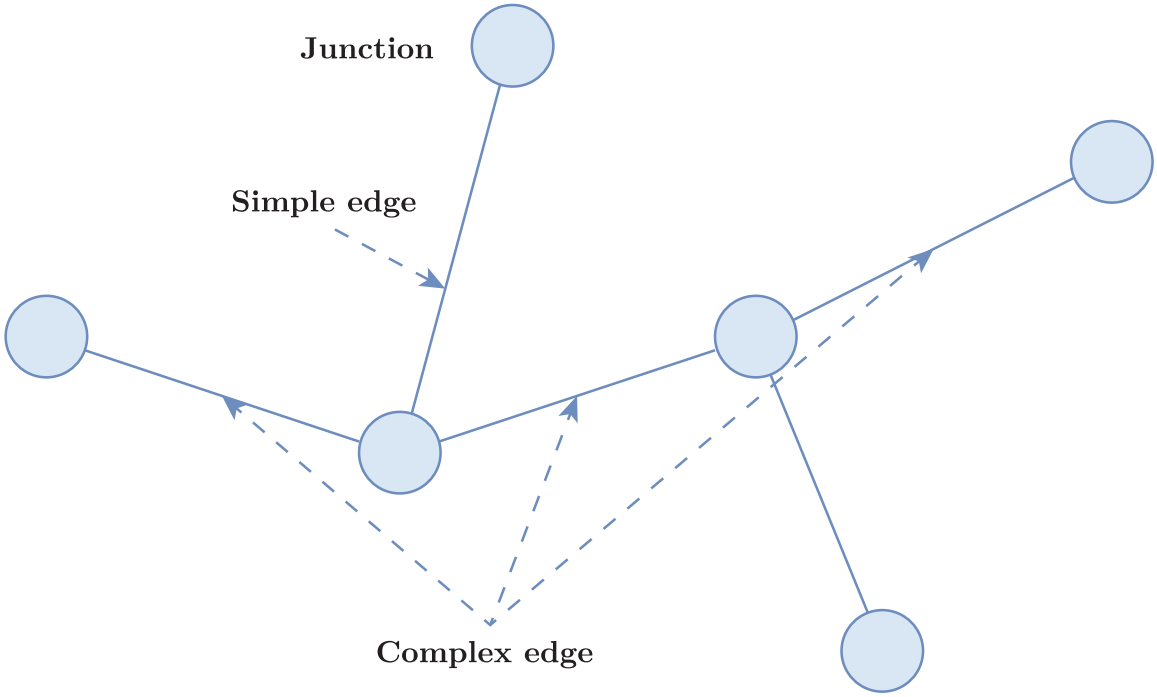


Figure 3.3: A depiction of the linear networks

Linear networks, also known as geometric networks, find representation in GIS through elements like roads, rivers, and utility grids (Figure 3.3).

There are two kinds of edges:

- **Simple Edges**—are straight lines connecting two neighbouring junctions. Resources go in at one end and come out at the other.
- **Complex Edges**—involve a network of connected lines with two or more junctions. Resources move from one end to the other, but they can also be diverted along the edge without needing to split the entire edge feature.

Junctions indicate the positions where edges either meet or terminate. A junction can link two or more edges, enabling the smooth transfer of a commodity (such as traffic or water) from one edge to another.

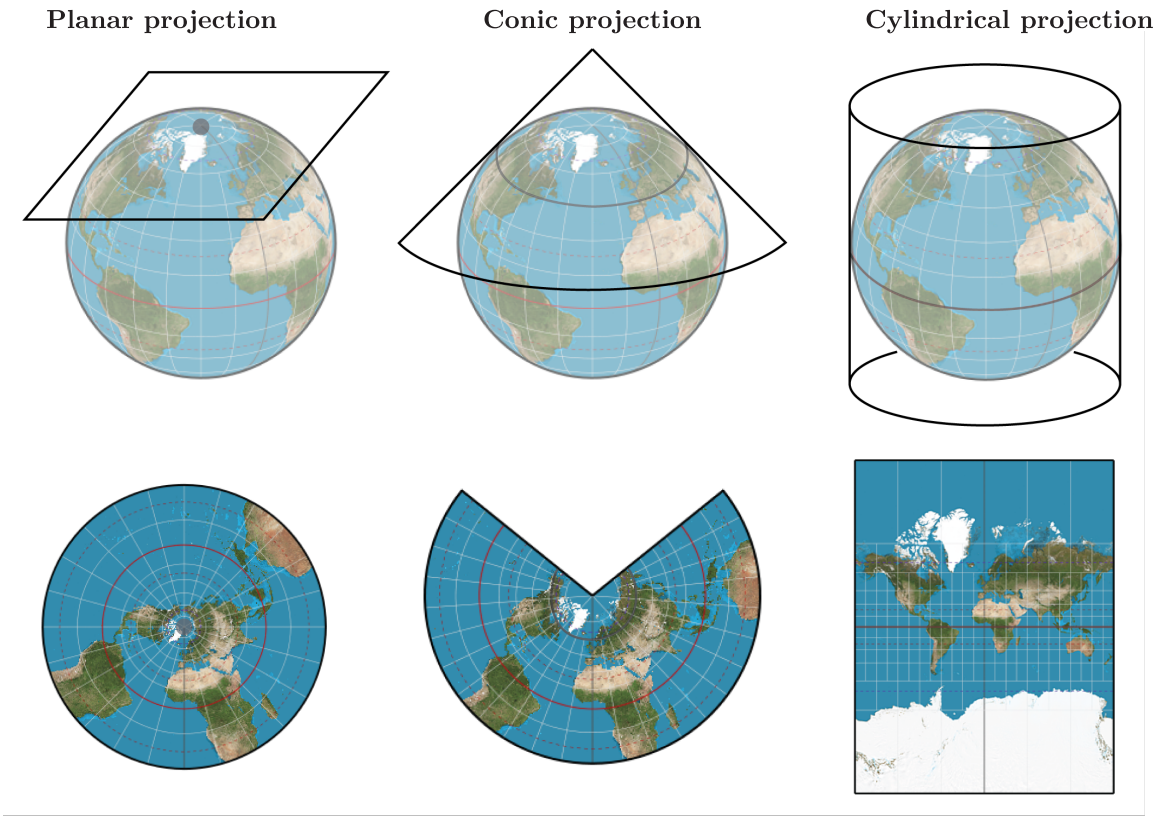


Figure 3.4: A depiction of different map projections (Adopted from [2])

The manipulation of data often becomes necessary in GIS due to variations in map projections. A projection, the method of transforming information from the Earth's curved surface to a flat medium like paper or a computer screen, introduces distortions (Figure 3.4). Different projections prioritize either maintaining accurate sizes or shapes of geographical features, but achieving both simultaneously is impractical.

3.2.2 Representation of Geospatial Data

Digitalizing real-world geospatial data is essential for working with and storing it on a computer. The utilization of basic geometric shapes is a common method to describe objects in the real world because they are suitable to be stored in database systems [19]. These types of data are commonly referred to as *spatial data types*, covering categories like point, line, and region. Additionally, more complex types such as partitions and graphs (networks) can be included. Spatial data types serve as a foundational abstraction, enabling the modelling of the geometric structure, relationships, properties, and operations with objects in space [19].

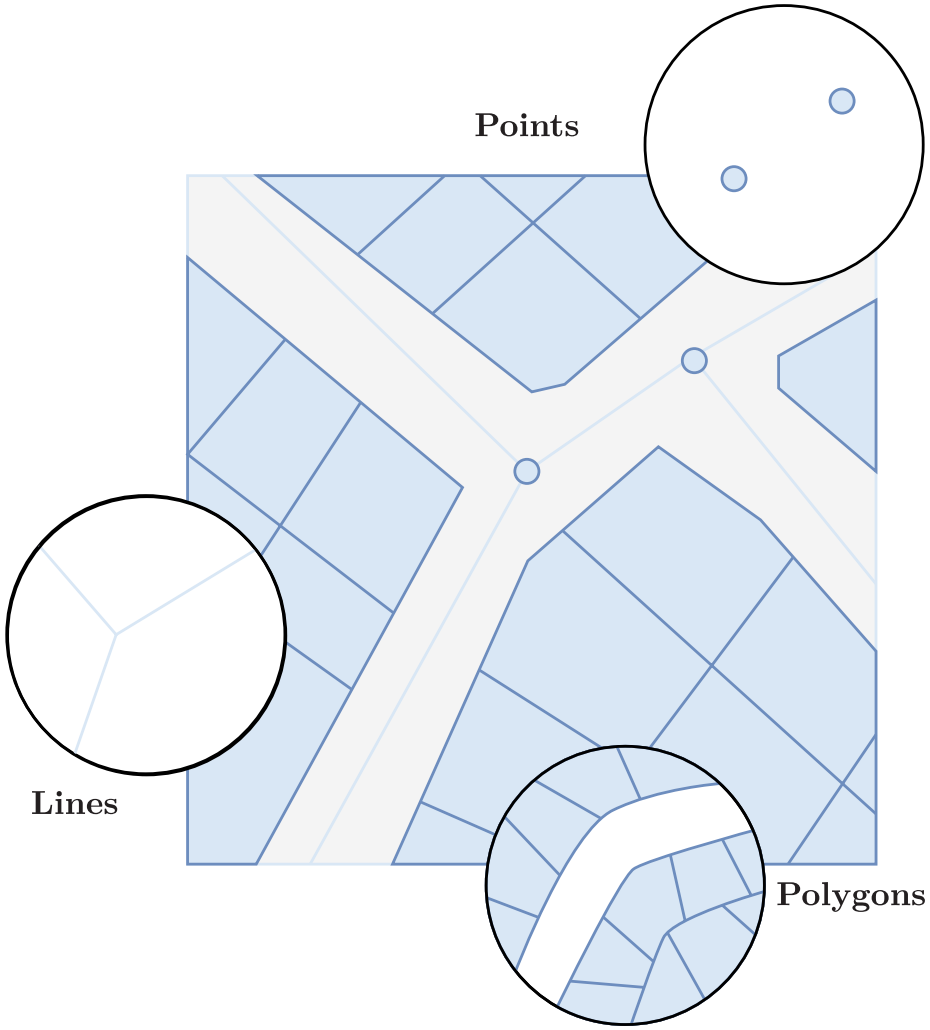


Figure 3.5: A depiction of points, lines and polygons on a map (Adopted from [21])

- **Point**—a point is a basic geometric entity that represents a single, precise location in space. It has no length, width, or area; it simply denotes a specific coordinate on a map (x, y) or in a three-dimensional space (x, y, z) (Figure 3.5). Points can define the location of a city on a map or the position of a landmark.

- **Line**—a line is a geometric object that extends infinitely in both directions, characterized by length but no width or area. In spatial data, a line is often used to represent linear features connecting two or more points (Figure 3.5). With a line, it is possible to define a road on a map, a river on a terrain model, or a flight path connecting two airports.

- **Area**—an area, also known as a polygon, is a two-dimensional geometric shape with a defined boundary. It encloses a space and has both length and width. Areas are used to represent the extent of geographical features (Figure 3.5). The area can define the boundary of a park on a map, the shape of a lake, or the footprint of a building on a site plan.

3.2.3 Storing Geospatial Data

GeoJSON is a widely used format for representing geospatial data and provides a simple and lightweight structure for encoding different types of geometry. In this section, we will explore how point features can be stored in GeoJSON, offering a clear way to describe geographic entities [4].

Here is a simple example of the GeoJSON format (Listing 3.1).

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [49.06, 13.23]
  },
  "properties": {
    "name": "Sample Point",
    "description": "A brief description of the point feature"
  }
}
```

Listing 3.1: Example of a GeoJSON point feature.

GeoJSON supports several types of geometric objects, allowing it to represent various spatial features. According to [4], the primary geometric object types in GeoJSON are:

- **Point**—represents a single geographic point and is defined by its coordinates, which are typically given as $(longitude, latitude)$ or (x, y) depending on the coordinate reference system.

- **LineString**—represents a sequence of two or more geographic points connected by straight line segments.

- **Polygon**—represents a closed, non-self-intersecting ring of coordinates.

- **MultiPoint**—represents a collection of points.

- **MultiLineString**—represents a collection of LineString geometries.
- **MultiPolygon**—represents a collection of Polygon geometries.
- **GeometryCollection**—represents a collection of heterogeneous geometries.
- **Feature**—is a fundamental building block in GeoJSON. It represents a geographic feature and consists of two main components: a geometry object, representing the spatial aspect and a properties object, containing additional non-spatial attributes or metadata.
- **FeatureCollection**—represents a collection of Feature geometries.

3.3 GIS Data Processing

Geo-data processing involves working with spatial data by manipulating, analyzing, and transforming it. GIS offers a variety of tools to assist in these tasks, including geocoding, distance analysis, spatial joint, and spatial trend analysis [8].

- **Geocoding**—a technique employed to pinpoint the location of objects or individuals globally. This process involves converting a postal address, which serves as an implicit geographic reference, into explicit spatial coordinates (Latitude and Longitude). It translates address information into precise geographical coordinates, facilitating accurate mapping and location identification [8].
- **Distance Analysis**—a tool employed to determine the nearest objects to a specific point, compute the area of a polygon, and generate buffer zones around these objects.
- **Spatial Joint**—serves the purpose of merging attribute data from two distinct datasets, relying on their spatial relationships. In this process, features from a "join" dataset are paired with features from a "target" dataset, determined by their spatial proximity or intersection. The attributes of these matched features are amalgamated, resulting in a new dataset that incorporates attributes from both of the original datasets.
- **Spatial Trend Analysis**—is a tool that entails overlaying two layers of objects, emphasizing statistical indicators derived from their spatial relationship. For example, this tool can be instrumental in identifying areas with high population density but limited access to retail services.

Chapter 4

Analysis

This chapter analyzes the problem of the retail site selection process from the user's point of view, examining it from various angles to understand the challenges and requirements associated with choosing an optimal location. The examination unfolds in three sections: an understanding of the use cases, a technical analysis, and a review of existing tools.

4.1 Target Audience

The primary beneficiaries of this system are non-professional individuals, such as aspiring entrepreneurs, small business owners, or local retailers seeking to expand. These individuals often lack technical expertise in geographic analysis but require informed decisions for choosing the right location for their new outlet.

4.2 Use Cases

In the dynamic world of retail, major players and emerging businesses alike recognize the critical importance of strategic location decisions. Large retail stores or newcomers often use the assistance of specialized experts. Hiring specialists for the retail site decision process involves professionals with backgrounds in market analysis, real estate, and business development. These experts leverage their experience to conduct thorough assessments, providing valuable insights that inform decision-making.

While the expertise of specialists enhances the strategies of larger retailers, the reality is that many small retailers face financial constraints that may limit their ability to use such assistance. Recognizing the unique challenges of resource limitations, small retailers can explore alternative solutions, such as systems, to overcome the challenges of opening new locations.

The most general cases in which such a system could help are, for example, identifying high-potential areas, competitive analysis and location evaluation.

4.2.1 Identifying High-Potential Areas

Identifying high-potential areas for opening new retail outlets is a crucial step in strategic business. This process involves a comprehensive analysis of various demographic factors to pinpoint locations where a retail establishment is likely to thrive. The goal is to maximize the chances of success by aligning the business with the preferences, needs, and behaviours of the target market in a specific geographical area.

To identify high-potential areas, certain requirements must be met. Firstly, there should be the availability of comprehensive data sets pertaining to demographics and consumer behaviour. Access to reliable and up-to-date data sources is crucial as it ensures the accuracy of the analysis.

In addition, implementing advanced analytical tools becomes essential. These tools are necessary for processing and interpreting the vast amount of data involved in the identification process. Geographic Information System (GIS) tools and data visualization platforms play a significant role in contributing to effective decision-making in this context.

4.2.2 Competitive Analysis

Competitive analysis is a systematic process of understanding the competitive landscape within a particular industry. It involves the examination of key players and their strengths and weaknesses. The primary objective is to gain insights to inform strategic decision-making, enabling businesses to position themselves effectively in the market.

For those without a background in market analysis, understanding the competitive landscape can be challenging. In this case, requirements stated in section 4.2.1 are still relevant. Moreover, the system should include data layering, similar to GIS, for displaying different data types.

4.2.3 Location Evaluation

Location evaluation involves analysis of potential locations to determine their suitability for establishing a new retail presence. This multifaceted assessment considers a range of factors that impact the success and viability of a retail operation in a specific geographic area.

Location evaluation poses a significant challenge for retailers because of the location attributes that are usually evaluated individually.

4.3 Functional Requirements

After examining the presented use case examples, I have identified essential functional requirements crucial for developing an effective tool for non-professional users.

4.3.1 User Interface

The system needs a user-friendly interface to cater to non-professional individuals seeking to expand their retail presence. This includes easy navigation through analytical tools and data visualizations without nesting them.

The user interface (UI) is a critical component of the location evaluation system, especially considering the non-professional individuals as the primary users. The UI should prioritize simplicity and intuitiveness to ensure accessibility for users without technical expertise in geographic analysis. A clean and well-organized design is essential, allowing users to navigate seamlessly through various analytical tools and features. Clear and concise menu structures, accompanied by easily understandable icons, should guide users through the process of identifying high-potential areas, conducting competitive analysis, and evaluating potential locations.

Visual elements within the UI, such as maps, should be presented in a way that facilitates easy interpretation of geographic data. The system should provide interactive elements, enabling users to manipulate and explore data effortlessly.

The UI should include contextual help features and tooltips to provide guidance throughout the analysis. Additionally, the system should offer user-friendly wizards or step-by-step workflows to assist users in conducting location evaluations.

4.3.2 Data Integration

The system should provide an integration of custom datasets and their configuration. Different sources may provide information in diverse structures; for this reason, compatibility with various data formats is essential. The system is going to have a clearly defined format for datasets.

Data layering is a feature within the system that allows users to gain an understanding of the complex and interconnected factors.

The system should allow users to overlay different datasets onto a single map, creating informative layers that provide a comprehensive view of relevant information, similar to GIS data layering. For instance, users may overlay competitor locations and demographic data on the same map.

4.4 Available Data

The thesis relied on the data repository `data.brno`[1], a platform that hosts numerous datasets released under open licenses[2]. Some of these datasets contain information crucial to the employed methodology. Understanding its contents is vital as subsequent chapters build upon this foundation.

Additionally, I will use this portal's datasets to test and demonstrate the system, but in this section, I will only analyze relevant datasets and their structure. Their application I will describe in section 7.3.

4.4.1 Data Requirements

As was noted in the section 2.2, the system will use the Huff model, described in section 2.2.1. This model requires such variables as the distance from a customer to a competitor and the size of a competitor. The distance can be calculated if the locations of a customer and a competitor are known.

The datasets that can fulfil these requirements for the system are "Number of people living at the addresses" and "Brno retail research". Both datasets follow the same initial data model, but they both differ in the "properties" attribute:

- **features**—an array containing individual features.
  - **type**—specifies the type of an object. In the case of these datasets, it is always set to "feature".
  - **properties**—key-value pairs containing descriptive information about the feature.

[1] Data of Brno—https://data.brno.cz/
[2] License for the datasets—https://creativecommons.org/licenses/by/4.0/

  - **geometry**—specifies the geometric shape of the feature.
    * **type**—specifies the type of an object. In this model, it is always a "Point" geometry.
    * **coordinates**—array with the longitude and latitude of the feature.

4.4.2 Number of People Living at the Addresses

The first dataset[3] contains information on the number of people living at the addresses. It has the following model:

- **pocet**—number of people
- **cislo_dom**—house number
- **cislo_ori_zn**—reference number and symbol
- **ulice**—street name

Here is an example of this dataset (Listing 4.1):

```
{
    "features": [
        {
            "type": "Feature",
            "properties": {
                "objectid": 1,
                "pocet": 4,
                ...
            },
            "geometry": {
                "type": "Point",
                "coordinates": [ 16.58..., 49.17... ]
            }
        },
        ...
    ]
}
```

Listing 4.1: Example of a "Number of People Living at the Addresses" dataset.

The only properties that are interesting to us are coordinates, latitude, and longitude, which represent the location of a potential customer, which is required in order to calculate the distance between a customer and a competitor.

4.4.3 Brno Retail Research

The second dataset[4] contains information about all the business outlets in Brno, such as area, type and many other less relevant attributes. The dataset has the following model:

- **sluzba_typ**—type of service

[3] Dataset with a number of people living at the addresses—https://arcg.is/1Lfbzb0
[4] Brno retail research—https://arcg.is/0CaaCS

- **plocha**—size of retail outlet (area)

Here is an example of this dataset (Listing 4.2):

```
{
    "features": [
        {
            "type": "Feature",
            "properties": {
                "sluzba_typ": "POH - restaurace",
                "plocha": "do 20",
                ...
            },
            "geometry": {
                "type": "Point",
                "coordinates": [ 16.58..., 49.17... ]
            }
        },
        ...
    ]
}
```

Listing 4.2: Example of a "Brno Retail Research" dataset.

Important properties in this dataset are the coordinates, "plocha" and "sluzba_typ". Plocha represents the area of the outlet, while sluzba_typ defines a category of a business. It will become more important further in the thesis.

4.5 Existing Tools

This section briefly explores existing solutions to the retail site selection problem for users seeking alternative ways to assist in a decision-making process.

Selecting the best retail site involves a combination of market research, data analysis, and location scouting. There isn't a single software or web application that can make this decision for the retailer. However, there are software applications that can provide a user with a set of tools to help with decision-making, such as analyzing market share and competition, targeting new customers or determining new sites. The most recent approaches to solving the retail site selection problem involve GIS.

4.5.1 Geographic Information System

As it was mentioned in Chapter 3, geographic information systems are very powerful tools for spatial analysis, which provide the functionality to capture, store, query, analyze, display and output geographic information (Figure 4.1). However, in order to conduct analysis related to retail site selection, these systems are used in conjunction with other tools and methods such as *decision-making system* (DSS) and *multi-criteria decision-making* (MCDM). These methods are not considered to be a part of GIS, but very often, they are provided as a separate set of extra tools to work with spatial data, and the user must know how to use them correctly.

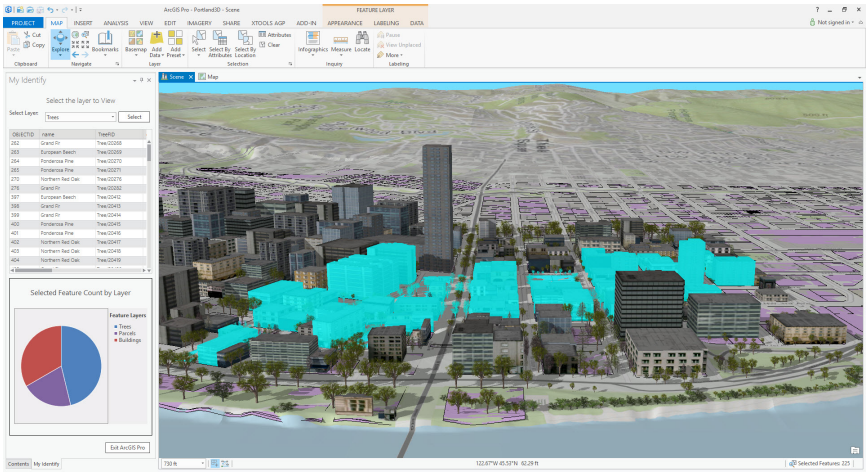


Figure 4.1: ArcGIS user interface

There are also private GIS systems, such as Buxton. They offer a distinct advantage in accessing extensive datasets for comprehensive spatial analysis. Unlike publicly available GIS tools, these private systems often provide proprietary data sources, enabling users to delve deeper into layers of information.

4.6 Conclusion

As it was mentioned in the section 4.5, there isn't a single software or web application that can make decisions for the retailer because the process involves analysis of multiple variables that can have a simultaneous impact on the success of a business. For this reason, site selection is a complex problem that can be tackled using multiple models, methods and procedures, making the process even harder for those who are unaware of them, which is why not anyone can dive straight into GIS. Additionally, the entire decision process may include extra tools that are not directly related to GIS and may not be available in the same application.

From the analysis, it can be seen that most of the requirements are not met by existing solutions. Therefore, it is needed to build a custom one. Such a system should prioritize user-friendly interface and data integration. It should follow a specific procedure, which can also be configurable for an individual's needs.

Creating a system that connects complex analytics with the needs of regular retailers is crucial for making retail decisions easier and available to everyone.

Chapter 5

Design

In this chapter, I will design a system that will fulfil user requirements described in Chapter 4.

5.1 The Idea

As was mentioned in Chapter 1, the system will implement the methodology from the journal [16]. It will be responsible for guiding users through the site selection process if it was conducted manually, but without requiring users to dive into all the technical details.

The system defines the following steps for the user to achieve the results (Figure 5.1).

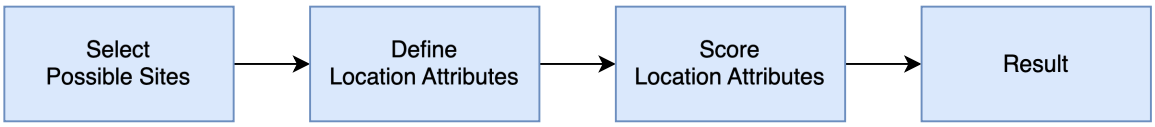


Figure 5.1: System process diagram.

5.1.1 Selecting Possible Sites

The methodology outlines two key concepts: Geo-demand and Geo-competition, both of which were described in Chapter 2. These concepts help to identify areas where there is a demand for a certain type of business, but it's currently lacking. Using a dataset with a number of people living at specific addresses, it becomes possible for the system to identify geo-demand automatically. Something similar is possible to do with geo-competition: the dataset for geo-demand and another one with real outlets and their square areas are going to serve as input for the Huff model to identify trading areas.

Once these concepts are identified, the system can apply kernel density estimation on both geo-demand and geo-competition. Then, by combining these concepts together, it becomes possible to identify areas where many people are not provided with the services of a business. The system will display this information as a heatmap layer, making it easy for the user to select potential locations.

5.1.2 Define Location Attributes

Once potential sites are selected, the next step requires the user to input location attributes, such as visibility, distance to competition or anything else that can described using quan-

titative or qualitative information. The information itself must be provided by the user as well.

5.1.3 Compare Location Attributes

When the user has defined the attributes, in the next step, he has to estimate them on a scale of 1 to 9, which is then used as input for the AHP method.

5.1.4 Result

The AHP method will output the list of potential sites that will contain ratings calculated based on the user's input.

5.2 System Architecture

The system for informed decisions in the site selection process will be developed as a full-stack web application (Figure 5.2).

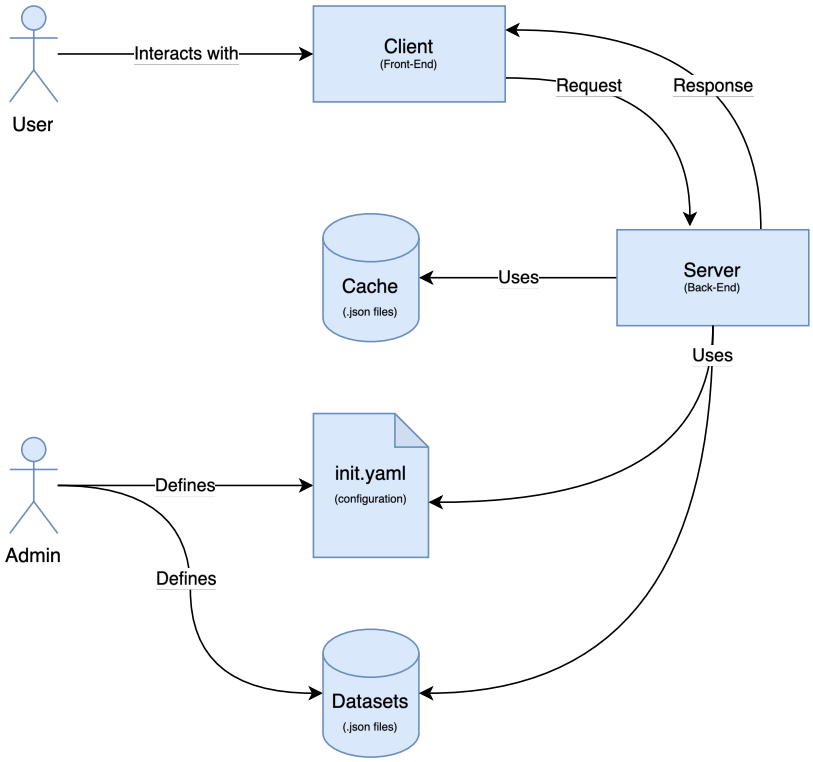


Figure 5.2: System architecture

User represents an individual that needs to utilize this system and its site selection process. All the user's actions are controlled by the client. The client meanwhile communicates with the configured server to handle certain regions on a map and to evaluate input provided by a user from the client.

Admin is an individual who configures the server with the region and related datasets. Both individuals can be the same person.

5.2.1 Front-End Architecture

The front-end should allow users to interact with spatial data and provide user inputs. It will have a dynamic map displaying various data layers and components presenting information relevant to each stage of the decision-making process, such as possible location selection, site attributes definition etc.

Process Control

Each step of the process will be managed client-side rather than on the server. This approach is deliberate because there is no need to store user-related data, sessions, or other sensitive information on the back-end. Instead, the back-end primarily handles pre-defined data utilized by these client-side processes.

Abstractions

The client application will contain several crucial abstractions for the functionality of the system: Marker, Attribute and Map. All of them will be described in Chapter 6.

5.2.2 Back-End Architecture

The back-end will serve as the central engine driving the system, offering a way to configure and adapt to different cities and datasets associated with specific categories of products. This flexibility enables the creation of multiple instances of the application for diverse cities and product categories defined within the back-end.

Configuration

Utilizing configuration information, the server can set up map positions and estimate potential high-demand areas based on the type of business and datasets provided by the user.

In order to analyze data by the system, the datasets will have to follow a certain format. For this reason, a user who sets the system up will have to use some parsing tool to convert the original dataset into the required one. This process will be described in detail in section 7.3.

Endpoints

To provide all the functionalities, the back-end will expose various endpoints accessible for specific actions, each triggered at distinct stages controlled by the clients.

5.3 UI Prototypes

The core component is a map. It allows users to interact with the system and control the entire flow of the decision-making process. The map will provide retailers with useful insights for the area based on their input that can be analyzed. Another important component will be a modal window where the user can configure the importance of location attributes. The design also presents prototypes of the system's main components.

5.3.1 Map

The central hub of the system is the map page, it is expected to be the most frequently accessed feature (Figure 5.3). This page contains various components, such as a menu, a panel with location information and controls to work with the map. The sidebar menu includes a toolkit, offering users a range of tools to manipulate and engage with data, for example they can explore data layers.

Moreover, an informative panel is integrated into the left corner of a map, presenting location attributes for users to analyze and make comparisons. This feature enhances the user's ability to obtain insights from the data presented on the map.

Additionally, the map serves as a dynamic platform where all procedure steps are executed and visually represented. To guide users through each step, hints are placed, providing informative cues and a smooth navigation experience. This comprehensive approach ensures that non-technical users can effortlessly engage with the system, analyze data, and make informed decisions with ease.

5.3.2 User Inputs

This module acquires user input in the form of a rating ranging from 1 to 9. This rating reflects the assessment of specific attributes related to potential locations. The purpose is to customize the behaviour of a procedure based on individual user preferences and evaluations (Figure 5.4). Input windows like this will pop up on the page with an interactive map during certain steps of a process. Users will be able to manually open them and rerun location evaluation if needed.

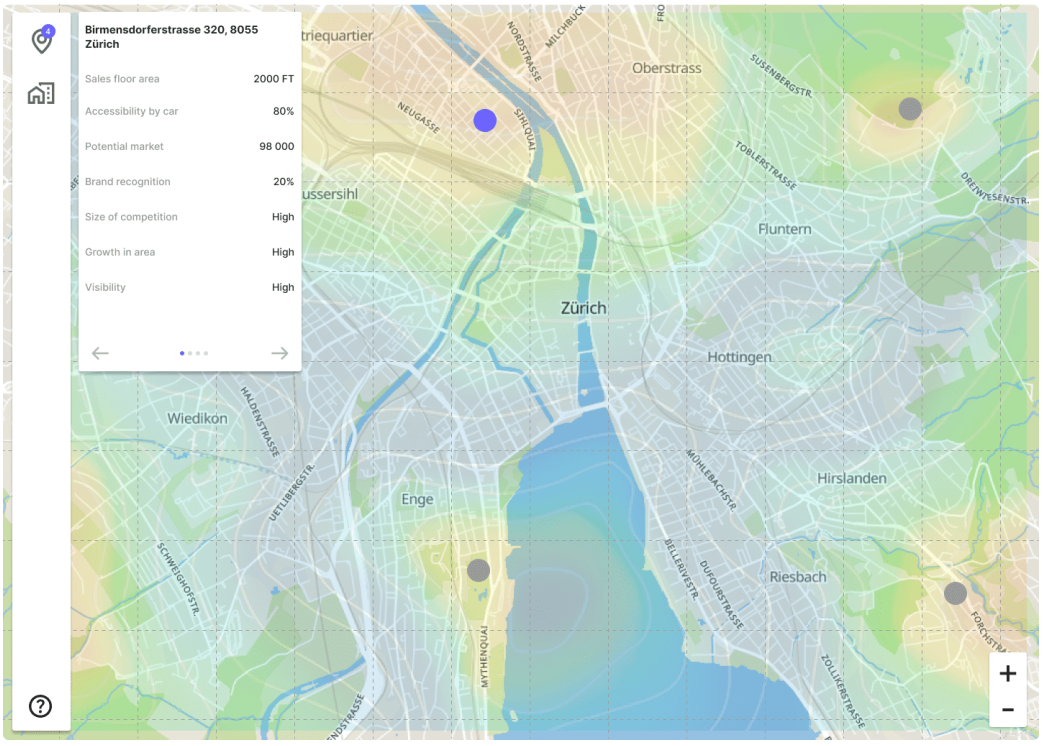


Figure 5.3: Prototype of a map component.

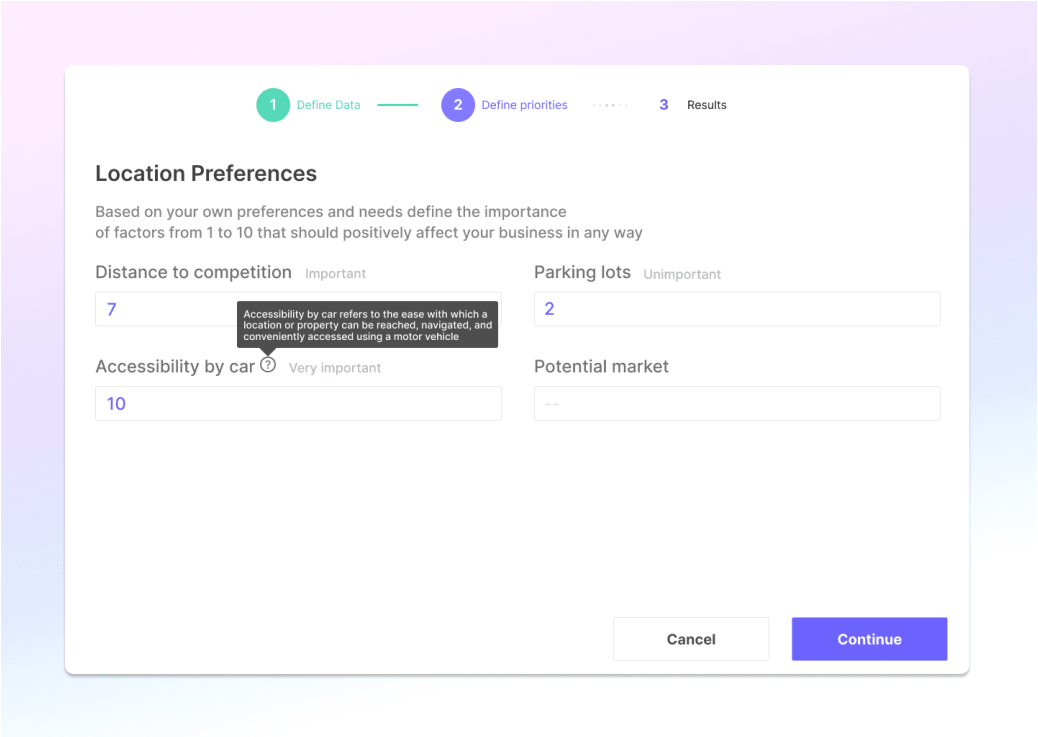


Figure 5.4: Prototype of a window with inputs.

Chapter 6

Implementation

As it was noted in section 5.2, a system with a site selection process is developed as a full-stack web application. This chapter describes which technologies were used and how each component of the system was implemented.

6.1 Used Technologies

The project involves building a simple application consisting of separate front-end and back-end parts. One main criterion that all the chosen technologies follow is the simplicity and extensibility for future contributors and potential users.

6.1.1 Front-End

- JavaScript is currently the leading programming language when it comes to the development of web applications. For this project specifically was chosen TypeScript[1], a superset of JavaScript, because of its strong typing capabilities, enhancing code quality and also for providing advanced features that support object-oriented programming (OOP).
- One of the goals of a front-end application is to allow users to interact with the map. For this reason, leaflet[2] library was used. It is a lightweight mapping library that provides interactive and customizable maps with straightforward API and modular design that makes it easy to extend with additional functionalities using OOP, accommodating future enhancements to the mapping capabilities of the application.
- In order to be able to communicate with a back-end, axios[3] library was used. It provides an intuitive API that simplifies data retrieval and manipulation.
- A core of the front-end architecture is React[4]—library for building user interfaces. Its component-based structure facilitates modularity and reusability, ensuring a robust and scalable frontend for future developers and users.

[1] Typescript—https://www.typescriptlang.org/
[2] Leaflet—https://leafletjs.com/
[3] Axios—https://axios-http.com/docs/intro
[4] React—https://react.dev/

6.1.2 Back-End

- When it comes to back-end development there is wide variety of languages that can be used. Each of them has its advantages and disadvantages in different development stages. When it comes to simplicity and extensibility, Python[5] is a good option. It is known for its simplicity and serves as the primary back-end programming language. Its extensive ecosystem of libraries and frameworks eases development and integration with other technologies.
- FastAPI[6], a modern web framework for building REST APIs with Python, was chosen for its exceptional performance and intuitive design. FastAPI offers interactive documentation generation and built-in validation capabilities, empowering future contributors to efficiently extend and enhance the back-end functionality.

6.1.3 Common technologies

Docker[7] is among the technologies leveraged, offering flexibility for both local application setup and deployment on servers.

6.2 Front-End Implementation

This section describes the implementation of the classes mentioned in section 5.2.1 that are part of the client's architecture (Figure 6.1).

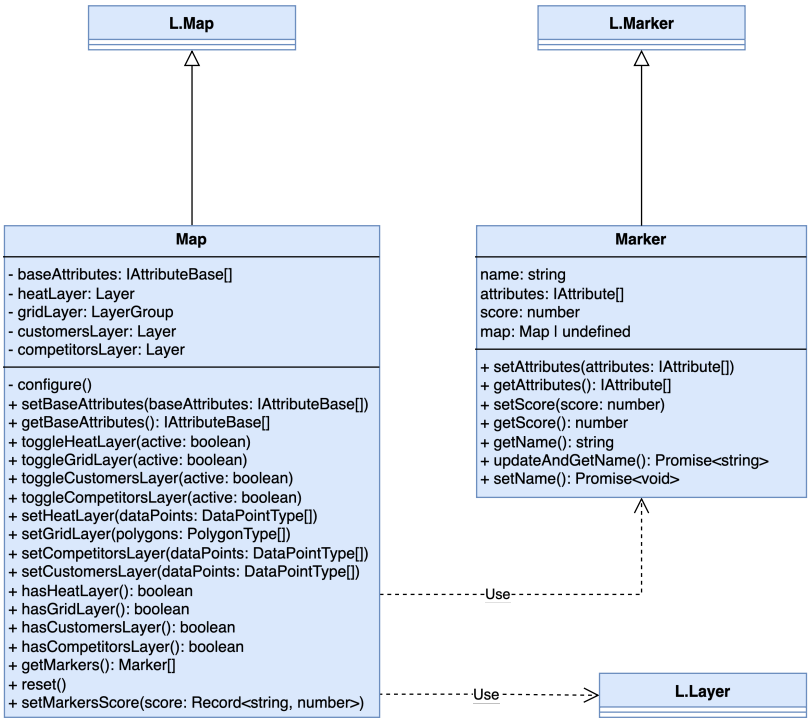


Figure 6.1: Class diagram in client architecture.

[5]Python—https://www.python.org/
[6]FastApi—https://fastapi.tiangolo.com/
[7]Docker—https://www.docker.com/

6.2.1 Marker

Marker class represents a location on the map, it is extended Leaflet's Marker to effortlessly work with Leaflet's Map while incorporating additional functionalities and information related to the system's objectives. For instance, it can store essential details such as location names based on their longitude and latitude or specific attributes associated with the location. Encapsulating such data within a Marker class minimizes unnecessary overhead with react's state management.

Here is an example of the object represented by marker class:

```
{
    name: "Brno, Kuldova 767/18, Czechia",
    attributes: [...],
    score: 83.54,
    latlng: [...],
}
```

6.2.2 Attribute

Attribute abstraction encapsulates the features or characteristics of a location as defined by the user. Essentially, it is a simple list of objects, each representing a distinct attribute associated with the location.

Here is an example of such an attribute:

```
{
    key: "Number of parking places",
    value: 14,
    maxValue: 30
}
```

6.2.3 Map

Extending Leaflet's Map, this class represents the state of the Map react component within the system. It encompasses the user-defined attribute scheme for all locations, automatically added markers, and associated scores. Additionally, the Map class exposes an interface for interacting with data layers crucial for the site selection process. These layers include high-demand areas, grid layers, and customer and competitor locations. Furthermore, the class offers methods for managing and manipulating these layers.

6.3 UI Components

There is a single primary UI component, which is a map and a couple of secondary ones, such as a container with the current step of the process information or a container with map information.

6.3.1 Interactive Map

`Map` is the main component in the entire front-end application (Figure 6.2). It contains its state, controls each step of the site selection process and works with all the architectural entities mentioned in section 5.2.1.

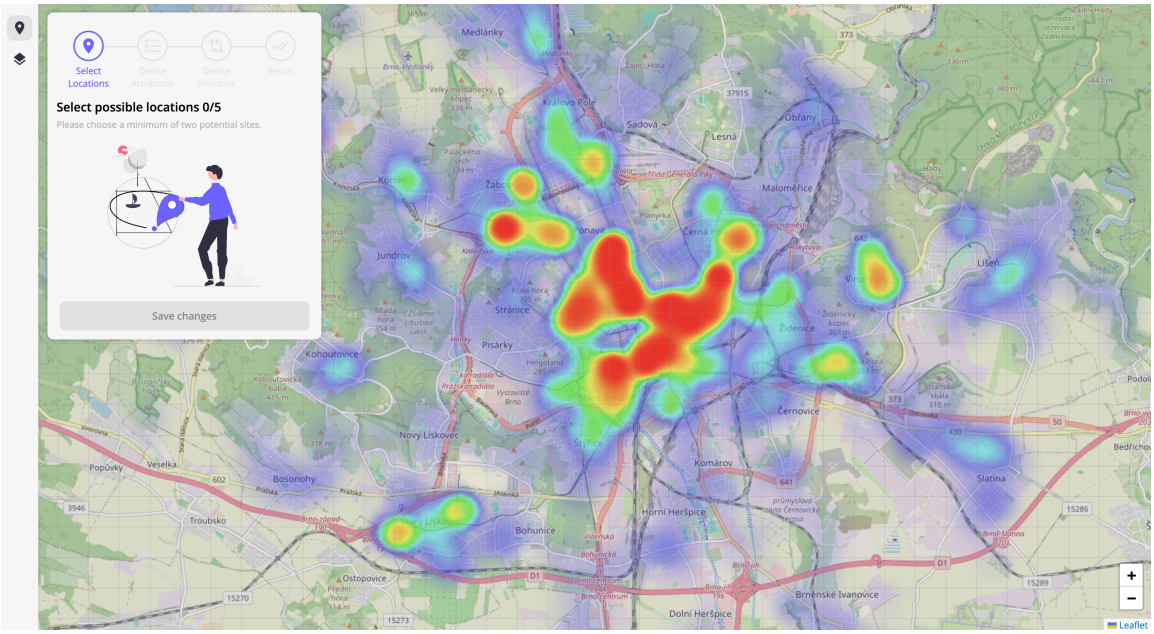


Figure 6.2: Map

The interactive map component itself is created and initialized using a customized `Map` class described in section 6.2.3. It is built-in inside of a custom hook `useMap` to provide a more natural way of interacting with it in React. For example, this hook exposes such methods as layer data fetching and layer toggling provided by the `Map`. Moreover, hook implements the state of a map to initiate re-render and reflect its changes in the `Map` component.

In order to track steps of the site selection process, map utilizes `enum SystemStatus` that contains such values as `SelectLocations`, `DefineAttributes`, `ScoreAttributes` and `Result`. Each value represents the step of a process, and based on this value, the map can display step-related information to the users and allow them to do certain actions that were described in section 5.1.

While progressing through each step of the process, all step-related information is displayed within the `StepInfo` component, nested within a `MapInfo` component. Additionally, `Map` itself provides `MapContext` to access all the data from child components like `StepInfo` or `Layers`.

6.3.2 Map Information

The `MapInfo` serves as a top-left wrapper component responsible for generating nested components based on the current URL the user is accessing. Currently, there are two components that can be displayed within a `MapInfo`: `StepInfo` and `Layers`. Users can select these components from the left menu.

6.3.3 Step Information

The `StepInfo` component is a React component generated within a `MapInfo` component on a map (Figure 6.3). It presents information relevant to each step and includes buttons enabling users to advance to the next step or return to the previous one. When transitioning

between steps, the `Map` component guarantees that all step-related data is preserved unless the user explicitly removes it.

To indicate the current step to the user, a `StepsContainer` component is employed as a wrapper for `StepInfo`. This container generates step-by-step components for each step, highlighting the current step accordingly. The state for this component is held by the `Map` and is accessed from its context.

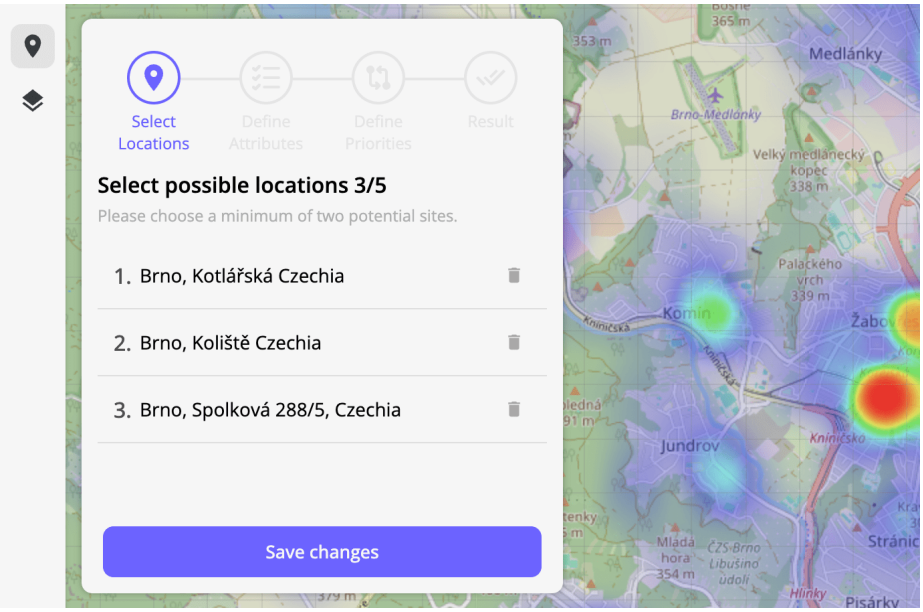


Figure 6.3: Step information component

6.3.4 Layers

The `Layers` component enables users to toggle the data layers sourced from the configuration file on the back-end. It provides information regarding the dataset currently in use and includes toggles for all available data layers (Figure 6.4).

This component is accessible during any step of the process, it can provide extra information for the users when selecting possible locations.

6.3.5 Modal Windows

Modal windows are commonly utilized in this application to gather user input during various phases of the process, such as `DefineAttributes` and `ScoreAttributes`. Modal windows are effective for containing user inputs due to their ability to temporarily suspend the main workflow, thereby focusing user attention solely on the task at hand.

Define Attributes Modal Window

The `DefineAttributesModal` serves as a modal component employed within the `DefineAttributes` step, where users are required to define a minimum of 2 features for potential outlets selected in the preceding `SelectLocations` step (Figure 6.5).

Within this modal window, users can define various attributes available to them. This implies specifying the attribute's name, value, and maximum value, which is essential for

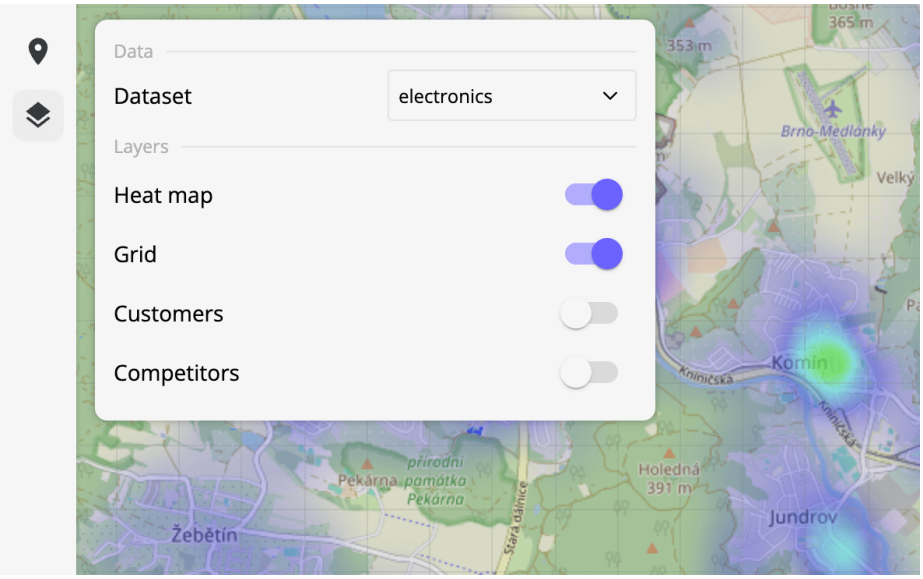


Figure 6.4: Layers component

data normalization. While the default input format for all values is numeric, users have the flexibility to switch between quantitative and qualitative inputs if necessary.

Define Score Modal Window

Once the attributes are defined, they need to be compared. Utilized within the `ScoreAttributes` step, the `DefinePrioritiesModal` component eases user interaction by enabling the comparison of attributes through sliders (Figure 6.6). This modal window offers users a straightforward method to evaluate and prioritize attributes relative to each other.

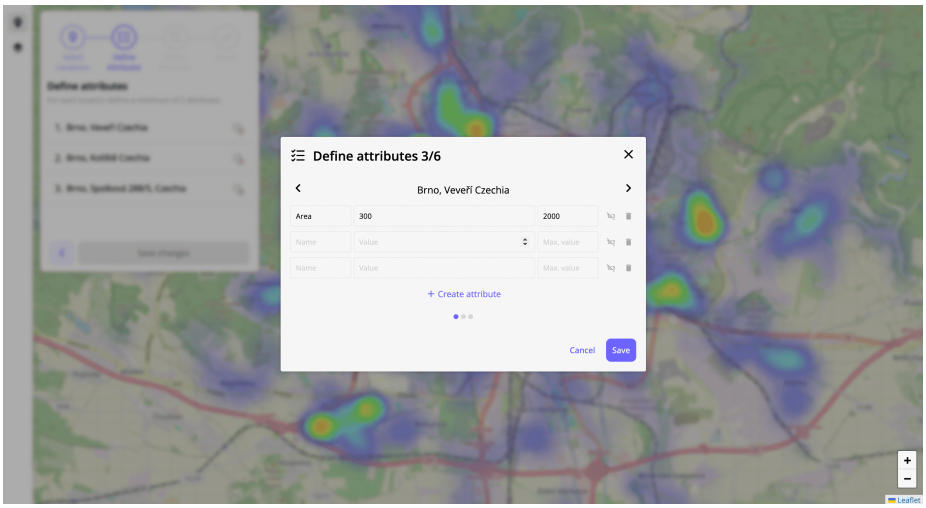


Figure 6.5: Modal window to define attributes

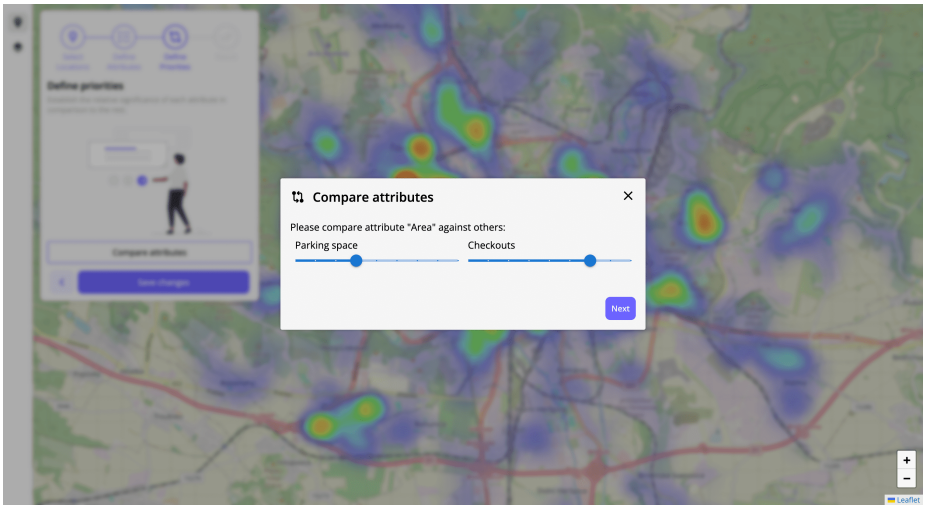


Figure 6.6: Modal window to compare attributes

6.4 Server Implementation

The server functions as a standalone REST API application within the overall system architecture, developed in Python with the assistance of the FastAPI library. It can be configured with various regions or datasets, including customer data and competitor information, via a configuration file. The server implements the methodology described in Chapter 2, and it exposes endpoints that expose specific information related to the current step of the process, which is controlled by the front-end client, as previously discussed in section 5.2.1.

6.4.1 Configuration

The configuration file for this application must be named `init.yaml`, and it must be located in the root folder of the server. In order for the server application to process this file, it must follow the structure:

- **area**—The name of a region depicted on a map. It serves as the key for retrieving the path graph of the region prior to server initialization. The utilization of this graph will be described in Section 6.4.3.
- **customers**—relative path to a dataset containing information about the number of individuals at specific addresses.
- **competitors**—a key-value structure of competitor datasets, where each key serves as the identifier for the dataset that users will see when choosing the business type.
  - **path**—relative path to the server's folder pointing to a dataset that contains information about a certain category of competitors.

– **distanceDecay**—distance decay factor. This exponent was described in Section 2.2.2. This property is optional, and the value is by default set to 1.75[8] implicitly, but it can vary for every business category in different regions.

Here is an example of a configuration file (Listing 6.1). In this example, the distance decay factor was selected randomly.

```
area: "Brno, Czech Republic"
customers: "./database/customers.json"
competitors:
  electronics:
    path: "./database/competitors-electronics.json"
    distanceDecay: 1.5
  grocery:
    path: "./database/competitors-grocery.json"
    distanceDecay: 2
  toys:
    distanceDecay: 2.5
    path: "./database/competitors-toys.json"
  ...
```

Listing 6.1: Example configuration file.

It is preferable for all datasets to correspond to the same geographic region. After completing the configuration process, the application can be initiated.

6.4.2 Dataset Format

As was mentioned in section 4.3.2, datasets that the user wants to integrate into this system must follow a specific format (Listing 6.2).

```
[
    [latitude_1, longitude_1, data_1],
    [latitude_2, longitude_2, data_2],
    ...,
    [latitude_n, longitude_n, data_n]
]
```

Listing 6.2: Data format definition.

- **latitude**—latitude position of a point.
- **longitude**—longitude position of a point.
- **data**—associated information or data related to the corresponding latitude and longitude coordinates. For example, the number of customers or the size of a competitor.

Here is a simple example of the dataset with a number of people at specific addresses (Listing 6.3).

[8] ArcGIS is a GIS system that has already implemented the Huff model. According to the ArcGIS documentation, the distance decay factor can range from 1.5 to 2—https://pro.arcgis.com/en/pro-app/latest/tool-reference/business-analyst/understanding-huff-model.htm

```
[
    [49.177, 16.581, 4],
    [49.164, 16.578, 5],
    ...,
    [49.166, 16.583, 49]
]
```

Listing 6.3: Example data with the list of latitude, longitude, and associated data that represent number of people.

6.4.3 High Competitive Areas Calculation

This section describes an implementation for determining regions with high competition. The methodology for this calculation was introduced in Section 2.3.

The competitive areas calculation was implemented as a function `get_geocompetition` located in the `/server/scripts/get_geocompetition.py` file (Figure 6.7).

The `get_geocompetition` function is called whenever the server is launched in order to start the evaluation. It is important to mention that this function requires a graph provided by the `osmnx` library. It is installed before based on the specified region in the configuration and then used in the function in order to find distances between customers and competitors. It will be described further.

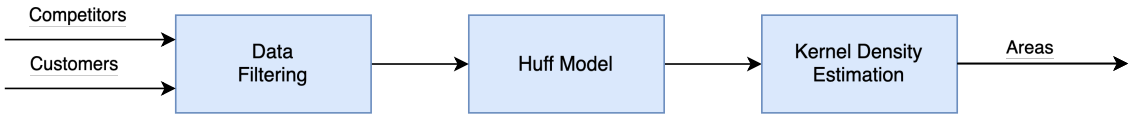


Figure 6.7: Modal window to compare attributes

As an input, the function accepts paths to the customers and the competitors, and the distance decay factor is related to the type of the competitors. Both datasets follow the format defined in section 6.4.2.

Data Filtering

The initial task within the function involves reading and transforming both datasets into `GeoDataFrames`. `GeoDataFrames`, offered by the geopandas library in Python, extend the functionality of pandas `DataFrame` to accommodate spatial data. A `DataFrame` in pandas is a two-dimensional, labelled data structure commonly used in Python for data manipulation and analysis. The data is organized into rows and columns. Each column can have a different data type.

Following this conversion, data undergoes filtering processes, such as the removal of entries that are not located in the specified area. This may happen if the dataset covers a greater area than the server, because the borders which define the area on the server are fetched separately.

Huff Model

Once the data is filtered, the Huff model, which was described in Section 2.2.1, is then applied. In the function, the Huff model was implemented in the following way:

- Iterate through each competitor:
  - Determine the nearest node to the competitor on the graph using method `nearest_nodes` provided by the `osmnx` library based on latitude and longitude coordinates.
  - Iterate through each customer:
    - Identify the nearest node to the customer on the graph using latitude and longitude coordinates.
    - Compute the distance between the current competitor and the current customer utilizing the `shortest_path_length` method from the `networkx` library.
    - Determine the travel time from the current customer to the current competitor.
    - Calculate the probability of the current customer visiting the current competitor and append it to the list alongside other customers.
  - At this stage, a 2D array is formed where each index represents a competitor, and the element value denotes the probability of all customers visiting the competitor at the given index.
- Convert the 2D array into a table where rows correspond to customers and columns correspond to competitors. For each competitor, compute the average probability of customer visits across all stores on the map (Table 6.1).

As a result, an array of averaged probabilities for all customers is obtained.

	Competitor 1	**Competitor 2**	**...**	**Competitor N**
Customer 1	0.08	0.06	...	0.05
Customer 2	0.06	0.05	...	0.07
Customer 3	0.07	0.06	...	0.06
...	...	...	...	...
Customer N	0.09	0.08	...	0.09

Table 6.1: Example of 2D array of probabilities of N customers visiting N outlets.

Kernel Density Estimation

Once the array of averaged probabilities is obtained, it is time to apply kernel density estimation (KDE) to create a distribution of probabilities across the map. The KDE was introduced in the Section 2.3.1. For the implementation, the library `scipy` was used. In order to create a smooth distribution, the `gaussian_kde` was utilized.

Grid-based Optimization

Rather than computing the distance individually between each customer and competitor, I have used a grid-based approach, covering the entire map with cells measuring 500 meters each. By assigning competitors and customers to their respective cells, I simplified the process of calculating distances between cells. Leveraging this grid structure allowed for the efficient reuse of distances, particularly beneficial in scenarios where cells contained multiple competitors and customers.

Dynamic Programming

The initial version of the `get_geocompetition` function had significant potential for dynamic programming to simplify computations, particularly regarding tasks like calculating distances between nodes or finding the nearest node in the graph. I decided to cache the inputs and their corresponding results. This way, if the same input is encountered again in functions like `get_nearest_node` and `get_distance_to_node`, the cached result can be returned directly, avoiding the need for redundant calculations.

Caching Results

The performance of the function for identifying areas with high competition falls short of user expectations. While ideally, it should deliver results within seconds, as demonstrated in Section 7.2, the computations involved are extremely time-consuming. Even with the implementation of dynamic programming and optimization techniques, the fetching time for these computations remains prolonged, thereby impacting user experience negatively.

For this reason, I have decided to cache the results of the `get_geocompetition` function. The cache will be stored in `/server/data` folder for future reuse, and if the user decides to rerun calculations, he is going to have to remove files with the cache. The cache has the same format as for datasets described in Section 6.4.2.

In order to avoid situations in which the very first user needs to wait for computations, I have decided to launch estimation before the server startup so that the initial task of the server is to read all the datasets, perform all the calculations and save the result. In this case, the very first user will get a response instantly because the server is going to use cached results.

6.4.4 Implementation of Analytic Hierarchy Process

This section outlines the implementation of the Analytic Hierarchy Process (AHP), as detailed in Section 2.4.1. The AHP procedure is implemented within the function `ahp_evaluate`, located within the `/server/scripts/ahp.py` file. This function is invoked during the final stage of the process to assess potential locations, considering both location attributes and attribute importance defined by the user.

To implement the Analytic Hierarchy Process, I have utilized the `Compare` method from the `ahpy`[9] library. This method only requires parsing user comparisons into the appropriate format (Listing 6.4).

```
{
    ("Visibility", "Potential Market"): 1/4,
    ("Visibility', "Accessibility by foot"): 4,
    ("Potential Market", "Accessibility by foot"): 9
}
```

Listing 6.4: Example dictionary representing pairwise comparisons in the AHP process.

6.4.5 API

- **/test GET** Endpoint for testing purposes.

[9] AHPy—https://github.com/PhilipGriffith/AHPy

- **/config GET** Retrieves configuration for the front-end application.
- **/customers GET** Returns dataset of customers.
- **/competitors POST** Returns dataset of competitors by category name.
- **/area POST** Calculates and returns areas with a high competition.
- **/result POST** Calculates the resulting list of locations with ratings based on user input.

6.4.6 Documentation

I have decided to utilize Swagger[10] for documenting my endpoints because of its integration with FastAPI. Swagger automatically generates interactive API documentation based on the defined FastAPI endpoints. This ensures that the API documentation stays up-to-date with the codebase without manual intervention, saving time and effort (Figure 6.8).

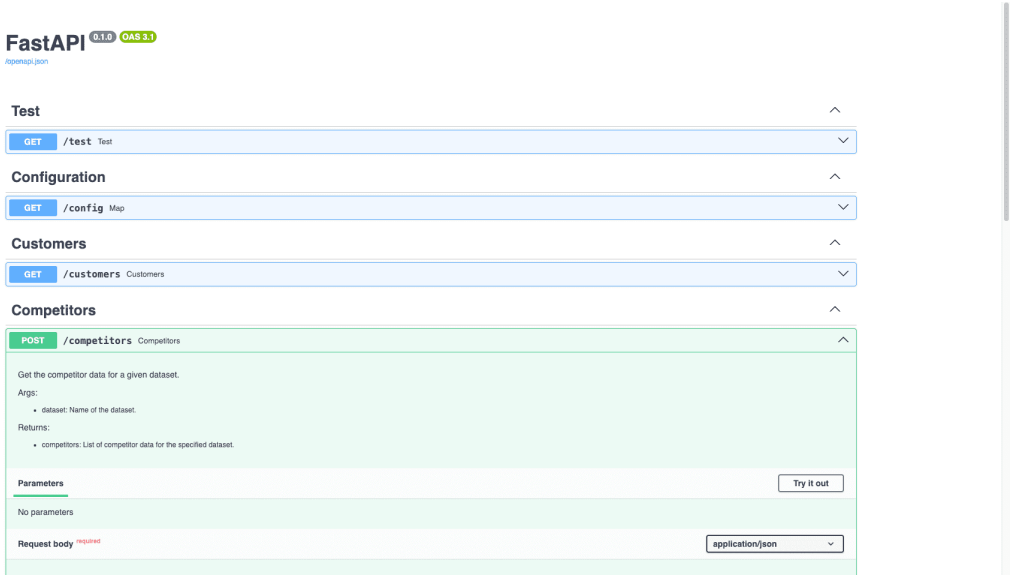


Figure 6.8: Screenshot of swagger documentation for the server.

[10] Swagger—https://swagger.io/

Chapter 7

Testing

Testing is a critical aspect of software development to ensure the reliability, functionality, and performance of the system. This chapter delves into testing strategies employed in this project. It covers unit testing, integration testing and performance testing to validate the correctness and efficiency of the system.

7.1 Unit Testing

Since this system is going to be available as a public repository that people cannot only use but also modify, it is important for such a project to be well-tested to ensure that key features are still working.

Unit testing focuses on testing individual units or components of the application in isolation to verify that they perform as intended. In the context of this project, unit tests were written to validate the functionality of specific components, functions, or classes within the codebase.

7.1.1 Front-End Unit Testing

One of the features that were noted in section 5.2.1 was that the front-end is responsible for managing the state of the system. Therefore, it has several components that must react to system changes. For this reason, unit testing becomes very important to ensure the correctness and reliability of the user interface components and their behaviour. The goal is to validate that the UI components render correctly, respond to user interactions as expected, and maintain their functionality across different scenarios.

Jest and React Testing Library

Jest is a JavaScript testing framework. It provides features for writing and executing tests, including assertions and mocking. React Testing Library is a testing utility for React that supports testing components in a way that reminds how they are used by end users.

Tests

Given that `StepInfo` serves as a primary component that the user is going to interact with very frequently, which is also responsible for presenting the process's state, it becomes necessary to test it.

As it was mentioned in Section 6.3.3, the `StepInfo` component has 4 states, such as `SelectLocations`, `DefineAttributes`, `ScoreAttributes` and `Result`. In order to maintain the functionality of each stage, I have written the following unit tests:

- Test to ensure that `StepContainer` is rendered correctly when default step `SelectLocations` is active. It means that the "Select Locations" step is highlighted, and the image is displayed when no locations are selected. Also, there is a test to check if it is rendered correctly after the user selects several locations.
- Test to ensure that when `DefineAttributes` step is active, it is rendered correctly, and the user can interact with it.
- Test to ensure that when `DefineAttributes` step is active, and the user fills all the necessary inputs, he is allowed to move to the next step. Additionally, there are several tests for invalid inputs. In this case, the test verifies whether the error message is displayed correctly.
- Test to ensure that when `ScoreAttributes` step is active, it is rendered correctly.
- Test to ensure that the final step is rendered correctly.

The tests can be found at `StepInfo.spec.tsx` file. Here is an example of the first test that checks whether components `StepContainer` and `StepInfo` were rendered correctly (Listing 7.1).

```
it('StepContainer renders correctly', () => {

    // Mock the return value of buildMap
    (buildMap as jest.Mock).mockReturnValue({});

    // Mock the implementation of ImageContent
    (ImageContent as jest.Mock).mockImplementation(() => <>Image</>);

    // Render the component under test
    render(
      <StepsContainer>
        <StepInfo />
      </StepsContainer>
    );

    // Assert that the mocked component is rendered correctly
    expect(screen.getByTestId(StepInfoTest.stepContainer)).toBeInTheDocument();

    // Assert if SelectLocations step is active by default
    expect(screen.getByTestId(icons.location)).toHaveClass("current");

    // Assert if default image is displayed
    expect(screen.getByText(/image/i)).toBeInTheDocument();
});
```

Listing 7.1: Unit test for rendering the StepContainer component.

7.1.2 Back-End Unit Testing

Back-end testing for this system is crucial to ensure that the server-side logic functions correctly and handles requests appropriately with different inputs. The back-end is highly configurable; it is a core of the system. It sets up front-end application based on datasets and configuration provided by the user. Therefore, it is necessary to ensure that the system works correctly in case of any errors or unexpected inputs.

Pytest

Pytest is a popular testing framework for Python that offers a simple syntax and powerful features for writing and executing tests. It provides capabilities for assertions, fixtures, parameterized testing, and mocking, making it well-suited for testing web applications built with frameworks like FastAPI.

Tests

To validate the server's functionality, there are several tests to ensure the proper operation of each endpoint. The testing involves simulating a user request and afterwards comparing its output with the expected result. While this process is straightforward for `GET` requests, I have implemented additional tests for `POST` requests with invalid data. This ensures that the server appropriately handles such scenarios.

It is important to mention that I have also created a specific configuration file `init.yaml`, containing generated datasets. This configuration file serves the purpose of generating consistent outputs, which allowed me to compare them with the expected results during testing. Configuration files and datasets for testing can be found in `/server/tests` folder. Additionally, I have added tests to check that the configuration file is processed correctly and that `get_geocompetition` caches results.

The tests can be found in `/server/test.py` file. Here is an example of a unit test for processing configuration file `init.yaml` (Listing 7.2).

```
def test_get_config(monkeypatch):
    monkeypatch.setenv("TESTING", "True")

    datasets = config.competitors.keys()

    expect = {
        "center": get_coordinates(config.area),
        "datasets": list(datasets),
        "grid": get_squares_list()
    }

    response = client.get(Urls.Config.value)
    assert response.status_code == 200
    assert response.json() == json.loads(json.dumps(expect))
```

Listing 7.2: Unit test for retrieving configuration settings.

7.2 Performance Testing

In the context of this project, performance testing is essential due to the complexity of the `get_geocompetition` function that calculates areas with high competition for the provided dataset. The time taken for these calculations can vary significantly based on the size of the datasets provided because of the algorithm's time complexity $O(n^2)$.

7.2.1 Planning and Preparation

In order to measure execution time, `time`[1] library was used.
At the beginning of `get_geocompetition` function, the `time` method was called for the first time, and its result was assigned to `start_of_execution`, at the very end of the same function method `time` was called again and, its result was assigned to `end_of_execution`. Subtracting `start_of_execution` from `end_of_execution` allowed me to determine the execution time.

7.2.2 Testing Process

The following steps outline the testing process:

- Generate different datasets with varying sizes and characteristics to represent realistic scenarios. All the generated datasets that were used in this testing can be found in the folder `/server/tests/performance`.
- Run the application with each dataset and measure the time taken to perform the calculations. Repeat the tests multiple times to ensure consistency and accuracy of results.
- Analyze the performance test results to identify trends (Table 7.1).

7.2.3 Results

Table 7.1 presents the time required to calculate datasets of varying sizes, ranging from 10×10 to 1000×1000 entries (Number of potential customers $\times$ number of competitors).

The dataset size represents the dimensions of the dataset, while the „Entries“ column indicates the total number of data points within each dataset. The „Time (seconds)“ column specifies the duration taken by the application to process and analyze the datasets.

Dataset Size	Entries	Time
10×10	100	1.7 seconds
100×100	10,000	35.38 seconds
1000×1000	1,000,000	908.37 seconds

Table 7.1: Time taken to calculate datasets of varying sizes

7.2.4 Conclusion

In conclusion, the performance testing conducted on the `get_geocompetition` function has provided valuable insights into its efficiency and scalability. The function's execution time

[1] Time library—https://docs.python.org/3/library/time.html

was observed to be influenced significantly by the size of the datasets processed, owing to its time complexity of $O(n^2)$. It is important to note that the function implements several caching and optimization techniques, detailed in Section 6.4.3, which impact its overall performance.

7.3 Testing Application with Real Data

Testing the application with real data is essential to validate its functionality and performance under realistic conditions. This particular test will be used in order to show the potential users and contributors how the system functions.

These are the steps to run and utilize this application effectively:

- Configure the back-end by creating and modifying the `init.yaml` file.
- Enable the application to assess newly incorporated datasets.
- Begin utilizing the application.

7.3.1 Configuration

In configuration, we must specify three crucial elements: geographic area, the dataset containing potential customers, and the datasets of competitors. It's essential that all datasets originate from the same geographic area.

In the context of geographical location, I will use „Brno, Czech Republic.“ As discussed in section 4.4.1, the datasets suitable for meeting the system's data needs include „Number of people living at addresses“ and „Brno retail research“. I have acquired both datasets. However, upon review of section 4.4.1, it's evident that both datasets differ significantly from the specified data format outlined in section 6.4.2. Therefore, it is necessary to parse both datasets to align them with the required format.

Parsing Dataset with Customers

Dataset with customers refers to a dataset with a number of people living at addresses. Using a Python script, the dataset was transformed into a format compatible with the application (Listing 7.3).

```
[
    [49.17789499100007, 16.581255326000075, 4],
    [49.164724902000046, 16.578846962000057, 5],
    ...,
    [49.16660763800007, 16.583760249000022, 49]
]
```

Listing 7.3: Result of a parsed dataset

Here is an example of a Python script that was used (Listing 7.4).

```
def serialize_customers():
    customers = read_json_file(CUSTOMERS, Customers)

    features = []
    for feature in customers.get("features"):
```

```
        count = feature.get("properties").get("pocet")
        lng = feature.get("geometry").get("coordinates")[0]
        lat = feature.get("geometry").get("coordinates")[1]
        features.append((lat, lng, count))

    write_json_file("customers-formatted.json", features)
```

Listing 7.4: Parser written in python to parse the dataset.

Parsing Datasets with Competitors

The dataset with competitors refers to the dataset of "Brno retail research". It was parsed the same way as the dataset with a number of people, using Python script.

Additionally, it was divided into multiple datasets based on business category `sluzba_typ`, such as grocery stores, pharmacies, or restaurants.

7.3.2 Data Evaluation

Data evaluation involves the system automatically assessing the provided data to pinpoint regions characterized by highly competitive areas, indicating a relatively high trading area. This process activates upon the user launching the back-end application, and its results will be then cached for future reuse. For Brno, the evaluation of all the provided datasets in total utilized 84 threads and took a total of 1971.31 seconds. It's worth mentioning that the results may differ depending on the number of threads available on the system where the evaluation was initiated.

7.3.3 System Launch

Once the evaluation is done, the user can launch the front-end application to start working with the system (Figure 7.1).

7.4 User Testing

As outlined in section 4.1, the target audience for this system includes individuals lacking technical proficiency in retail analysis. Therefore, it is crucial to demonstrate the system to these individuals to gather feedback and drive further enhancements.

7.4.1 Objective

The testing scenario aims to confirm that users can successfully navigate the entire decision-making process facilitated by the system and obtain the desired results.

7.4.2 Participants

Three student participants, aged 19-23, took part in the study, with no level of expertise in retail analysis. Each participant was introduced to the system and provided with the following objective: "Imagine yourself as someone wanting to open a store, restaurant, or cafe and facing challenges in selecting the optimal location. Use this application to fulfil this objective."

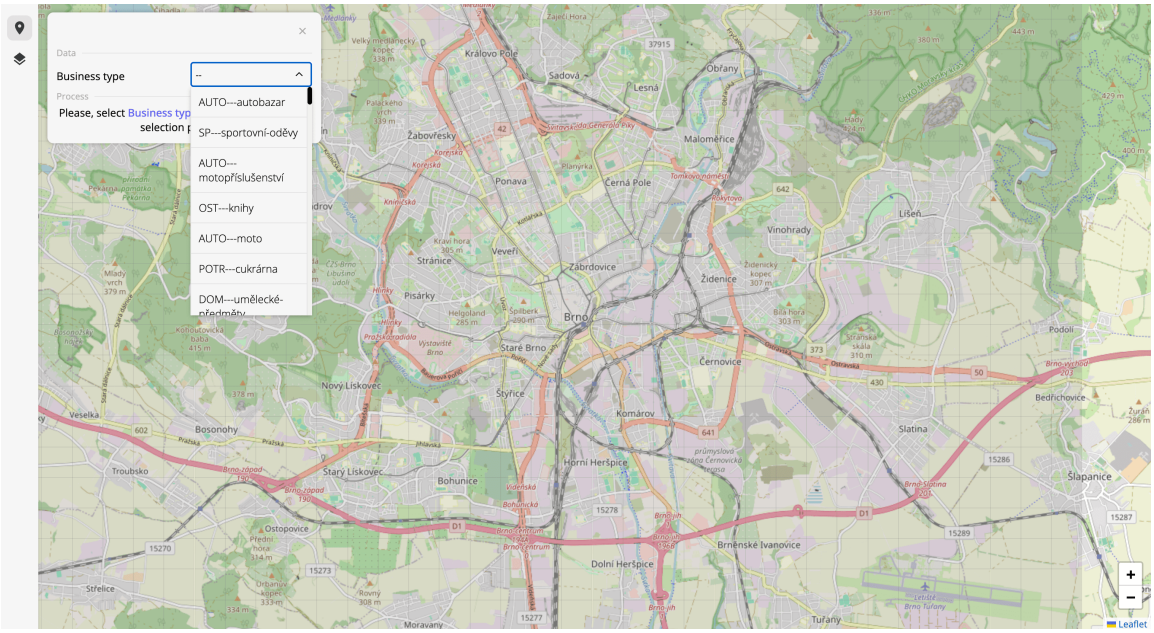


Figure 7.1: Screenshot of a working application.

7.4.3 Testing Environment

As all participants were from Brno, the system was configured with data specific to this location to ensure participants felt familiar with the area.

7.4.4 Gathered Feedback

During the testing sessions, participants often encountered difficulties in understanding the steps of the process or understanding the displayed information. For instance:

- During the site selection stage, two participants were uncertain about the purpose of the heatmap feature and what it represents in the application, and one participant did not realize that locations needed to be selected in this step.
- When defining attributes for the sites, one participant expressed confusion about the types of attributes that could be specified.
- Participants pointed out that the project lacks an introduction, a description of specific UI components detailing their functions and how to interact with them, as well as an explanation of the utilized data.

7.4.5 Improvements

In response to the feedback collected in section 7.4.4, the following enhancements have been implemented in the front-end application:

- Each step now includes a brief description outlining the information presented to the user and the actions required at that particular stage. These descriptions have been incorporated into modal window components and the `StepInfo` component (Figure 7.2).

- When defining attributes, the user will be provided with suggested attributes and an explanation of what could be specified as a value (Figure 7.3).
- I have developed a separate page containing an introduction to the project, its UI components, and the utilized data. Users can also access a demo video demonstrating the entire site selection process (Figure 7.4).

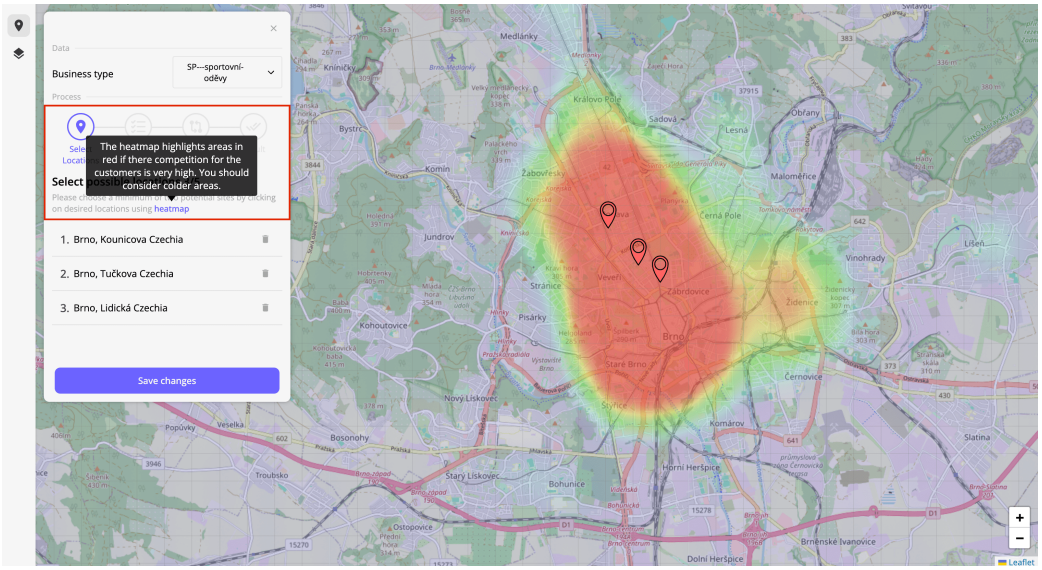


Figure 7.2: Hints and tooltips for the user to navigate.

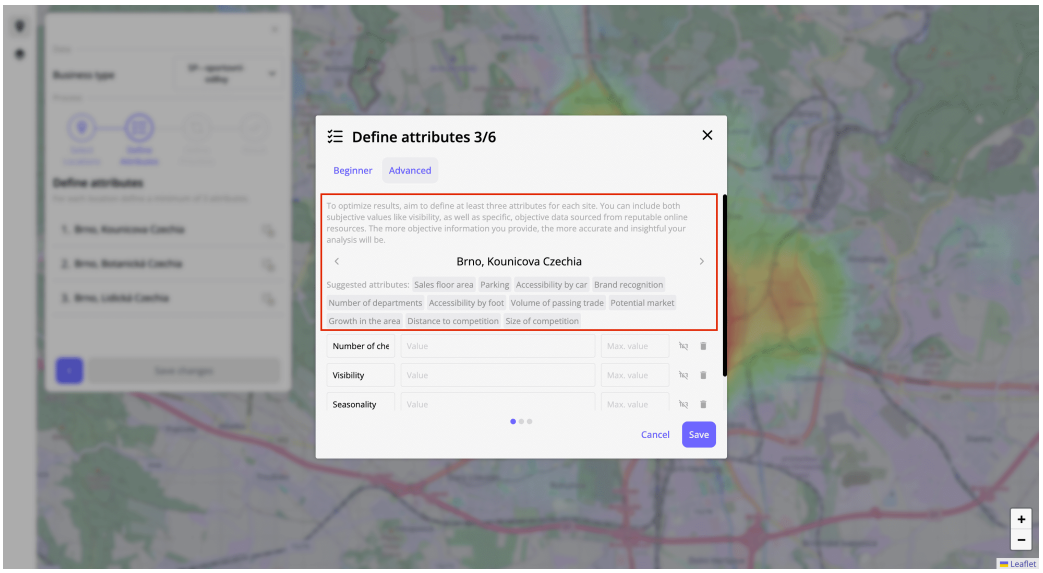
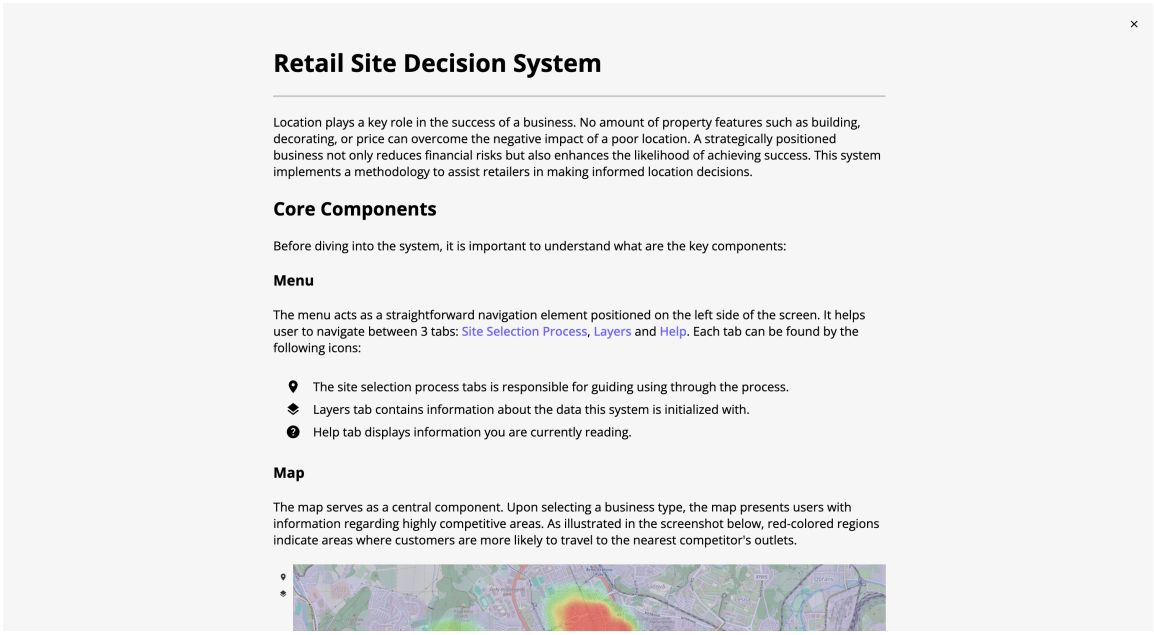


Figure 7.3: Suggested attributes and help for the user when defining attributes.



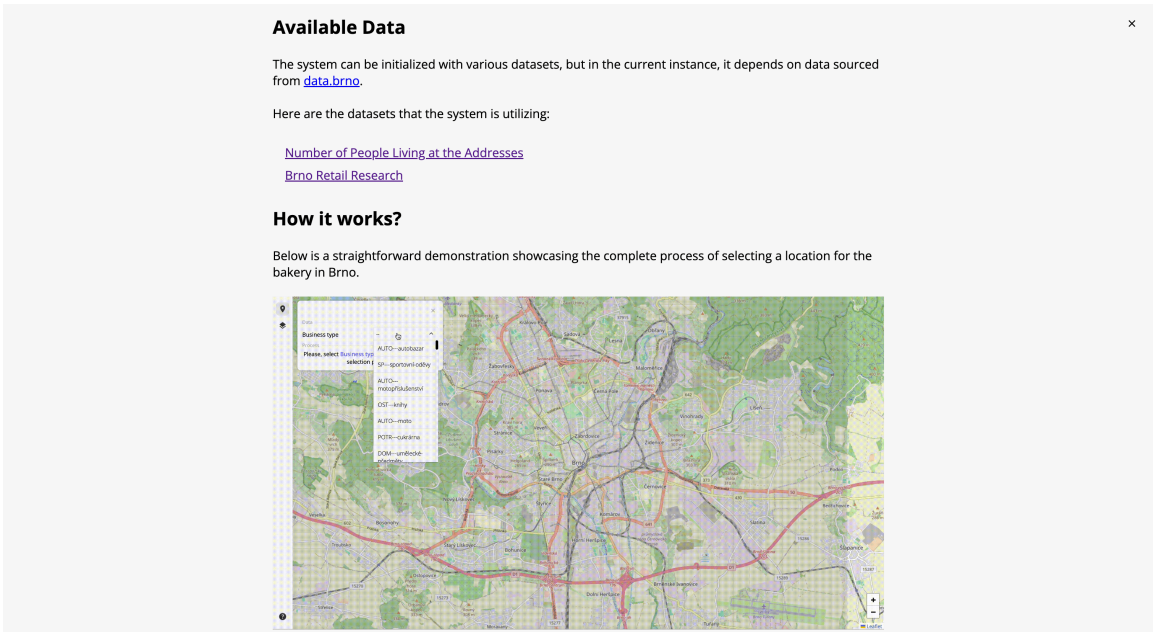


Figure 7.4: Page with introduction to the project and tutorial to use it.

Chapter 8

Conclusion

In this thesis, I addressed the challenge in the retail business: the significance of location in determining success. I began by acknowledging that location prevails in the success of a business. Historically, decisions regarding business locations relied heavily on intuition and subjective assessments, often considered more of an „art“ than a science.

Recognizing the limitations of such subjective approaches, especially in the face of evolving information systems and data analytics, I set out to develop a systematic solution. The aim of this thesis was to empower retailers with a comprehensive methodology for making informed location decisions, utilizing data analysis capabilities and geographic information system (GIS) technologies.

Drawing inspiration from notable methodologies outlined in academic literature, particularly the approach proposed by [16], I developed a system adapted to the needs of retailers. This system was designed to simplify the location decision-making process by integrating multiple datasets, utilizing GIS capabilities, and incorporating user preferences.

Throughout the work on this thesis, I have carefully created a framework for the development and implementation of this system. I have delved into the theoretical foundations of retail location assessment, studied the intricacies of GIS technologies, identified the target audience and functional requirements, studied existing solutions, and designed the interface and functionality of the system.

Having established a solid foundation, I implemented the system using advanced technologies. Extensive testing was conducted to ensure that the system was reliable, performant and compatible with real data.

Looking to the future, I see this system being used by retailers not only in the city of Brno but also in various urban landscapes around the world. By making this tool available in a public repository, I hope to enable retailers of all sizes to thrive in a competitive marketplace.

Furthermore, at the recent Excel Fit 2024 conference, this project stood out among innovative solutions, scoring an impressive 41 points. It caught the attention of attendees, who recognized its potential to transform location decision-making in retail [6].

Bibliography

[1] Anderson, S. P., Goeree, J. K. and Ramer, R. Location, location, location. *Journal of economic theory*. Elsevier. 1997, vol. 77, no. 1, p. 102–127.

[2] Battersby, S. *Map Projections* [https://gistbok.ucgis.org/bok-topics/map-projections]. 2017. DOI: 10.22224/gistbok/2017.2.7. The Geographic Information Science & Technology Body of Knowledge (2nd Quarter 2017 Edition), John P. Wilson (ed.).

[3] Baviera Puig, A., Buitrago Vera, J. and Mas Verdú, F. Trade areas and knowledge-intensive services: the case of a technology centre. *Management decision*. Emerald Group Publishing Limited. 2012, vol. 50, no. 8, p. 1412–1424.

[4] Butler, H., Daly, M., Doyle, A., Gillies, S., Hagen, S. et al. *The geojson format*. 2016.

[5] Durbach, I. N. and Stewart, T. J. Modeling uncertainty in multi-criteria decision analysis. *European journal of operational research*. Elsevier. 2012, vol. 223, no. 1, p. 1–14.

[6] Turytsia, O. *The Retail Site Location Decision System* [online]. Vysoké učení technické v Brně, 2024 [cit. 2024-05-07]. Available at: https://excel.fit.vutbr.cz/submissions/2024/010/10.pdf.

[7] Jeannie Evers, E. E. *GIS (Geographic Information System)* [online]. National Geographic Society, 2023 [cit. 2023-10-12]. Available at: https://education.nationalgeographic.org/resource/geographic-information-system-gis/.

[8] Lille, F. de pharmacie de. *Data processing, analysis and mapping* [online]. sigles, 2018 [cit. 2023-10-12]. Available at: https://www.sigles-sante-environnement.fr/en/study-methodology/data-processing-analysis-and-mapping/.

[9] Gross, J. *Linear regression*. Springer Science & Business Media, 2003.

[10] Hernandez, T. and Bennison, D. The art and science of retail location decisions. *International Journal of Retail & Distribution Management*. MCB UP Ltd. 2000, vol. 28, no. 8, p. 357–367.

[11] Huff, D. and McCALLUM, B. M. Calibrating the huff model using ArcGIS business analyst. *ESRI White Paper*. 2008, p. 1–33.

[12] Huff, D. L. Defining and estimating a trading area. *Journal of marketing*. SAGE Publications Sage CA: Los Angeles, CA. 1964, vol. 28, no. 3, p. 34–38.

[13] Huff, D. L. A programmed solution for approximating an optimum retail location. *Land Economics*. JSTOR. 1966, vol. 42, no. 3, p. 293–303.

[14] Mendes, A. B. and Themido, I. H. Multi-outlet retail site location assessment. *International Transactions in operational research*. Wiley Online Library. 2004, vol. 11, no. 1, p. 1–18.

[15] Pesch, R. and Ryan, B. *Trade Area Analysis* [online]. University of Wisconsin-Madison, 2024-05-04. Available at: https://economicdevelopment.extension.wisc.edu/articles/trade-area-analysis/.

[16] Roig Tierno, N., Baviera Puig, A., Buitrago Vera, J. and Mas Verdu, F. The retail site location decision process using GIS and the analytical hierarchy process. *Applied Geography*. Elsevier. 2013, vol. 40, p. 191–198.

[17] Saaty, T. L. *What is the analytic hierarchy process?* Springer, 1988.

[18] Saaty, T. Decision making for leaders (p. 19). *Pittsburgh: RWS Publications*. 1992.

[19] Schneider, M. Spatial data types: Conceptual foundation for the design and implementation of spatial Database systems and GIS. In: Citeseer. *Proceedings of 6th International Symposium on Spatial Databases*. 1999.

[20] Stanley, T. J. and Sewall, M. A. Image inputs to a probabilistic model: Predicting retail potential. *Journal of Marketing*. SAGE Publications Sage CA: Los Angeles, CA. 1976, vol. 40, no. 3, p. 48–53.

[21] Sümer, S. I., Sümer, E. and Atasever, H. Promoting development through a geographic information system-based Lodging Property Query System (LPQS) for Antalya, Turkey. *Information Development*. Sage Publications Sage UK: London, England. 2016, vol. 32, no. 4, p. 1055–1067.

[22] Taherdoost, H. and Madanchian, M. A Comprehensive Overview of the ELECTRE Method in Multi Criteria Decision-Making. *Journal of Management Science & Engineering Research*. 2023, vol. 6, no. 2.

[23] Wang, Y., Jiang, W., Liu, S., Ye, X. and Wang, T. Evaluating trade areas using social media data with a calibrated huff model. *ISPRS International Journal of Geo-Information*. MDPI. 2016, vol. 5, no. 7, p. 112.

[24] Ryan Pesch, B. R. *Trade Area Analysis* [online]. Community Economic Development, 2023 [cit. 2023-19-12]. Available at: https://economicdevelopment.extension.wisc.edu/articles/trade-area-analysis/.

[25] Węglarczyk, S. Kernel density estimation and its application. In: EDP Sciences. *ITM web of conferences*. 2018, vol. 23, p. 00037.

[26] Yap, J. Y. L., Ho, C. C. and Ting, C.-Y. A systematic review of the applications of multi-criteria decision-making methods in site selection problems. *Built environment project and asset management*. Emerald Publishing Limited. 2019, vol. 9, no. 4, p. 548–563.