IT UNIVERSITY OF COPENHAGEN

Manipulation of Geospatial Data: a Dashboard Within Hungary's Geographical Scope

Gergo Gyori

(gegy@itu.dk)

Department of Data Science, IT University of Copenhagen

Bachelor Project

Course code: BIBAPRO1PE

Supervisor: Eleni Tzirita Zacharatou

15th of May, 2023

Table of content

1. Abstract

This study aims to develop a geospatial dashboard that enables users to identify geospatial areas based on specified constraints. It addresses two key knowledge gaps. Firstly, there is a lack of a geospatial dashboard in Hungary that focuses on the intersection areas of multiple multipolygon layers. Developing such a user-friendly dashboard would allow users to identify geographical areas based on their specified constraints. Secondly, traditional life satisfaction studies have overlooked the perspective of the user, neglecting their preferences and value propositions. By incorporating user-focused surveys with both open-ended and closed-ended questions, this research aims to bridge this gap and provide a deeper understanding of user perceptions in the context of place satisfaction. Various geospatial data manipulation techniques, including buffer, unary union, and simplification are used to achieve accurate intersection areas of multiple multipolygon layers. The findings highlight the high computational cost associated with data manipulation techniques.

2. Introduction

Traditional place satisfaction studies have long been used to identify the important attributes that contribute to resident satisfaction and value propositions within a particular place. However, these studies have often overlooked the perspective of the user, or resident, when exploring perceptions of a place. Instead, they have relied on organisations, service providers, local governments, community groups, or businesses as the foundation for evaluating place attributes [37]. This approach may not fully capture the value propositions that are most important to residents when assigning value in the context of place.

Therefore, in the present study, I aim to fill the gap in traditional place satisfaction studies by exploring the perceptions of residents when evaluating the value propositions of an ideal place. By incorporating the needs and preferences of the users when evaluating the value of a place, I build a dashboard that is user-focused when exploring ideal places to live and to go on vacation. This results in a more accurate understanding of what makes a place desirable for those who use it, and leads to the creation of a better-designed, more user-centred dashboard that meets the needs and preferences of the people.

The overall purpose of this paper is to create a user-friendly dashboard that allows users to define multiple constraints in order to identify ideal places. I examine efficient methods for managing and handling geospatial data, while also analysing the effects of object simplification on the resulting areas. The dashboard serves as a tool to provide an effective identification of intersecting geospatial layers. I use geospatial manipulation techniques and geospatial analysis, conducted in Python [44]. Geospatial manipulation techniques are conducted with the help of packages such as Geopandas (GPD) [13] and Shapely [10]. Geospatial analysis is conducted by the use of GPD, Shapely and Memory Profiler [20]. To the author's knowledge, there is no such dashboard that has been developed in Hungary, thus, the present study focuses on utilising Hungary as a case study to develop and implement the proposed dashboard.

The paper is structured as follows. First, I present the “Background” section, which acquaints the reader with the packages used and relevant terms along with their explanations. Second, I introduce the “Methodology” section, providing details on the survey, data preparation and the dashboard. Third, in the “Results” section, I present the findings of the analysis structured around three main parts: (a) survey, (b) data preparation and (c) dashboard. Fourth, I put forth a “Discussion” section that explores practical applications of the dashboard. Fifth, I include a “Limitations” section centred around the potential sources of inaccuracy that may affect the results generated by the dashboard. Finally, the paper concludes with recommendations for improving the intersection calculation process.

3. Background

I used two Python packages to analyse the different methods for computing the intersection of multiple layers. OSMnx played a crucial role in obtaining a comprehensive list of various amenities and environmental layers, such as lakes, rivers, and forest areas. This package utilises key-value tags [38] to capture specific features and attributes associated with map elements.

By leveraging the power of key-value pairs, I used queries to retrieve objects representing areas that were tagged by users. These tags provided valuable information about the characteristics and properties of the identified areas, facilitating the subsequent analysis and intersection calculations.

Another package is GPD is a package for working with geospatial data in a tabular format. It extends the functionality of Pandas [43] to allow for easy manipulation of geospatial dataframes, including performing spatial joins, buffering, and other spatial operations. Some of the GPD functions rely on Shapely. Shapely is a package for working with and manipulating geometric objects in Python. It provides a range of functions for constructing and manipulating geometric shapes, including points, lines, and polygons.

In addition to Python packages, I retrieved the different statistical data from various sources. The climate data was gained from KLIMADAT [14] which focuses on planning, information technology, and monitoring development related to Water Management and the impacts of climate change.

3.1 Terminology

Point: It represents a specific location in space. It has no size, shape, or dimension. In coordinate geometry, a point is represented by an ordered pair of numbers (x, y) that correspond to its position on a two-dimensional plane.

LineString: It is a sequence of two or more points that defines a line segment or a curve. It is a basic geometric object used to represent a path or a linear feature in a two-dimensional plane or a three-dimensional space. A LineString consists of a series of connected straight line segments that form a continuous path.

Polygon: It is a two-dimensional geometric shape that is made up of a finite number of line segments and enclosed by a closed path. It is a closed figure with three or more straight sides that form a closed loop. Each line segment of the polygon is called a side, and each endpoint where two sides meet is called a vertex.

Multipolygon (MP): It has multiple objects: polygons, points and linestrings. They are grouped together to form a single object. A MP is represented as a collection of objects, each with its own set of vertices. It is used to represent complex geographic features. The objects are not overlapping each other, in this report it is used to cluster the overlapping objects.

Intersection: The term "intersection" pertains to a mathematical procedure employed in the field of geospatial analysis to identify and extract shared or common regions between different geographic objects. Specifically, in the context of this report, the intersection method is applied to MP objects.

Vertex (vertices): It is a point where two or more straight lines or edges of a polygon or polyhedron meet. In other words, a vertex is a point where an angle is formed by the intersection of two or more lines or edges

Layer: In this report a layer is used to describe a MP, where the objects have no overlap. A layer can be a collection of multiple objects or one MP object.

Buffer: It is a function [13] that provides an approximate representation of the set of points that fall within a specified distance from the given geometric object.

Unary union (UU): It is a function in the Shapely library that allows for the merging or combination of multiple geometric objects into a single object. It takes a collection of geometric objects as input and returns a single object representing the union of all the input objects. This function is useful for consolidating overlapping geometries into a unified representation, or eliminating duplicated objects.

Explode method: In the context of MP objects, the term "explode" refers to a function that decomposes or splits a MP into its constituent elements. When a MP contains multiple objects such as polygons, points, and linestrings grouped together, the explode function separates them and returns a list of individual objects.

Dataframe (DF) : A DF is a table-like data structure that can be changed in size and contains rows and columns. It can store data of any type. The data is stored in a tabular format with each row representing a record and each column representing a field. DF is used to store the data of each individual object. In this paper DF is referred to as GPD DF.

GRID: GRID is data refers to a spatially gridded dataset, which means that the data is organised into a regular grid of cells or pixels, with each cell representing a specific area or location on the Earth's surface. GRID data is commonly used in geographic information systems (GIS) and remote sensing applications, as it allows for efficient data storage and processing, as well as spatial analysis and visualisation.
Projection: It refers to transforming the Earth's curved surface onto a flat plane. It involves converting geographic coordinates into a projected coordinate system, enabling accurate spatial measurements and analysis. Projections are used to create maps and visualisations, minimising distortions based on purpose and area of interest. By applying a projection, geospatial data can accurately represent spatial relationships and facilitate precise measurements and analysis.
Ground truth: The ground truth comprised merged layers obtained from queries that focused on similar amenities or nature-related objects, consisting exclusively of polygon objects. I removed point and linestring objects from the dataset because they cannot adequately represent areas due to their lack of physical extent. No additional manipulation was performed on the ground truth data, except for projecting it into the EPSG:23700 coordinate system.
Accuracy assessment: It is a process used to evaluate the reliability and precision of geospatial data or the results obtained from geospatial analysis. It involves comparing the generated data or analysis results with a reference dataset to measure the level of agreement or discrepancy between them. In the context of geospatial data analysis, accuracy assessment aims to quantify the quality of the data by examining how closely it aligns with the true or known values. It helps to assess the degree of error or uncertainty present in the data and provides insights into the overall accuracy and validity of the analysis.

4. Methodology

During the research process, I utilised ChatGPT [34] to enhance the quality of the writing.

4.1 Survey

The present study is based on a partially free-form survey, where open-ended questions provide respondents with the opportunity to answer in their own words, without being constrained by a predefined set of response options [1, 9]. This format enables respondents to express their needs and preferences freely, without predetermined ideas. The information gained from open-ended questions can then be used to generate new categories, explore unexpected findings of needs and preferences, and gain a deeper understanding of the factors that influence place satisfaction. The other half of the survey is built upon a more traditional approach present in place satisfaction studies, where closed-ended survey questions provide respondents with a predetermined set of options or answer choices to choose from.

The aim of the survey is to determine the user needs and preferences of ideal places prior to the development process. The close-ended part of the survey consisted of a 5-point scale questionnaire that measured the importance of different amenities and weather conditions. Even though I used a 5-point scale to measure the importance of the criteria, grouped the responses into two major categories. The criteria marked as "neither important nor unimportant", "important", or "very important" were grouped together as "being used," while those marked as "not important" and "unimportant" were merged into a single category "not being used". Therefore, this part of the survey is in a close-ended categorical response format, where the responses are limited to a 5-point scale and then are grouped into two major categories, "being used" and "not being used.

For easier understanding, I divided the survey into two parts, with the same questions in each. The first part focused on criteria for choosing a place to live, while the second part focused on criteria for choosing a vacation destination. Both sections included a hard-coded list of different types of predefined amenities based on previous knowledge [42], such as shop, gym, access to highway, educational institution, workplace, access to public transport, park, forest, and lake or river. Additionally, I included climate factors, such as average temperature and air quality.

An important aspect of the survey is that users were also able to describe their needs and preferences in free text. This allows me to create a dashboard that is a better-designed and more user-centred platform. I collected and classified the main points from the free text responses provided by the users in my survey. To accommodate responses in Hungarian, I translated them into English for classification purposes.

This process involved pairing the collected points with corresponding data sources available to me. For instance, mentions of terms such as "coffee shop" and "coffee" were classified under the new category of "cafe,"; references to "Tabak shop" were classified as "cigarette"; responses containing terms like "bars", "bar" and "pub" were classified as "pub"; the response "No wind at all" was classified under the category of "wind"; responses, such as “Not too rainy” were classified as “precipitation”; mentions of terms like "football field" and "football" were classified as "sport"; “tourist attractions”, “sightseeing spots”, “hiking areas”, “waterfall” were classified as “spots”, and finally references to "hospital", "healthcare" and "doctor" were classified as "healthcare" to capture the relevant information provided by the respondents. Therefore, based on the information added in free text, in total, I identified the following 16 categories: cafe, pub, wind, precipitation, sport, healthcare, gas station, playground, cost of living, salary, spots, beach, swimming pool, clubs, concerts, crime rate and restaurant.

However, due to time constraints and difficulty of finding relevant dataset, responses indicating the crime rate, availability of concerts, clubs, gym, sport, swimming pool, spots were left unclassified. Preferences related to beaches were disregarded due to the lack of beaches in Hungary.

In the final dashboard, I included the following 3 categories with respective subcategories, which incorporated both predefined categories and free-text generated categories from survey responses: (a) amenities like supermarket, restaurant, pub, gas station, tobacco shop, cafe, playground, healthcare and educational institutions; as well as access to highway and public transportation; (b) aspects related to “Nature”, such as river, pond and lake, as well as green areas, such as forest, park and mountain; and (c) “Weather Statistics”, such as average daily minimum temperature, rainy days per year, average precipitation in millimetre/month, number of summer days (daily maximum temperature above 25°C), average wind speed per day, average maximum wind gust speed and average household price.

4.2 Data Preparation

The methodology involved the extraction of various amenities and areas from OSMnx queries using key and tag pairs. While OSMnx is a free and open source software, discrepancies in data quality are expected [15, 16, 23]. The different layers were created by merging various queries. For instance, to obtain GPS coordinates of schools, three different key-tag pairs are used, namely “amenity=school”, “building=school”, “education=school”.

4.2.1 Buffer

The OSMnx query results were initially retrieved in the EPSG:4326 (WGS 84) coordinate system. To ensure the precise buffer of objects within the layer, it is necessary to convert the layer into the EPSG:23700 projection system, which is known to provide an accuracy of approximately ±1 metres within the area of Hungary [24]. Following the projection, the value of the GPD buffer method corresponds to metres in distance. Figure 1 (left) shows the visible result polygons of 4 objects after being buffered by 20 metres.

4.2.2 Unary union

To determine the most efficient approach for expanding different layers and minimising RAM usage, I employed the Memory Profiler package. Throughout the measurement process, I utilised two data frames: one containing individual objects (DF-AO) and the other containing a multi-polygon object (DF-MPO). I employed three methods: Buffer Objects (BO-M), MP Buffer (MPB-M), and Exploded MP Buffer (EMPB-M).
BO-M involved expanding all individual objects in DF-AO and generating a final layer using the UU method. MPB-M focused on buffering DF-MPO, while EMPB-M utilised the explode method in DF-MPO to separate objects into individual parts. The expansion process resulted in a final layer using the UU method.
In Figure 1 (right), I observe the merging of object 2, 3, and 4 into a single object after applying the UU operation. However, object 1 remains distinct as it does not overlap with any other areas.

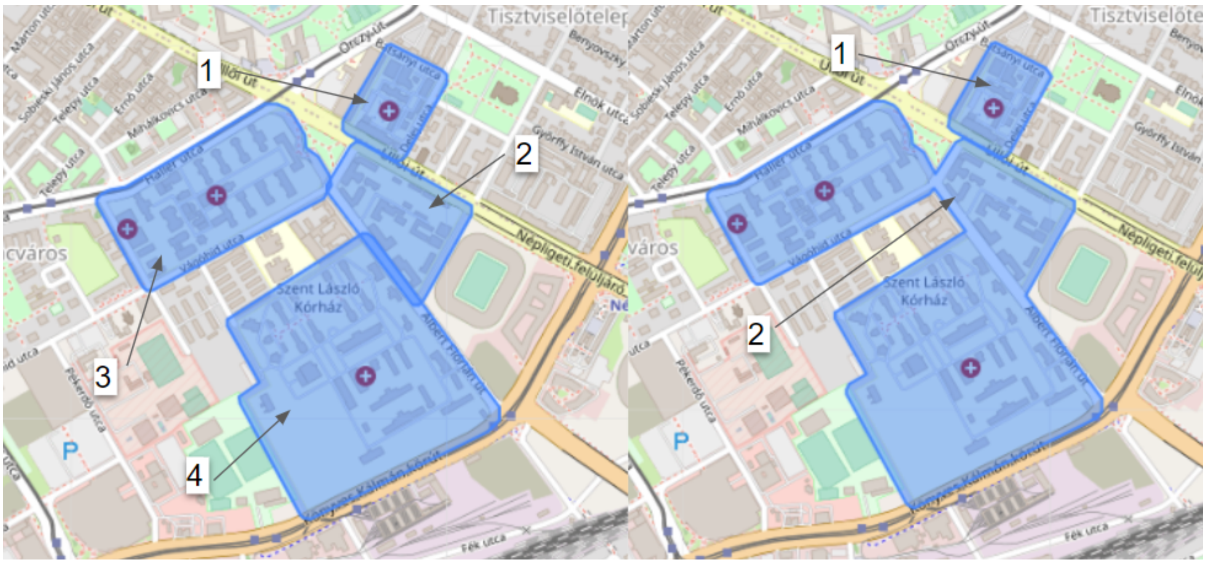

Figure 1: Each object was buffered by 20 metres. On the left picture, overlapping areas are visible. After the UU, overlapping areas disappear and the four individual objects merge into two objects.

I conducted a performance evaluation of the method on the healthcare, park, and public transport layers using buffer distances of 10 and 1000 metres. In order to assess the effectiveness of the buffer process, I systematically varied the buffer values across a range of

1, 10, 20, 50, 100, 200, 500, and 1000 metres on the park, pub, and restaurant layers. For each value within this range, I measured the time taken to expand the objects and perform the UU method.

4.2.3 Simplification

The dashboard's main feature is finding intersections among multiple layers. With a focus on performance, I selected the two layers containing the highest number of objects to measure the intersection method. To reduce the time of the calculation, I utilised the GPD "simplify" [10] method for reducing the point count in the layer. The simplify() function in GeoSeries generates a simplified version of each geometry in the series. As depicted in Figure 2, the original object (in grey) was simplified by factors of 0.2 and 0.5, resulting in the visible blue object. The tolerance value represents the level of simplification. It employs the Douglas-Peucker algorithm [22] to segment the original line or polygon into smaller parts, connecting them with straight lines. Points closer to the straight line than the specified tolerance are removed while preserving the original endpoints.

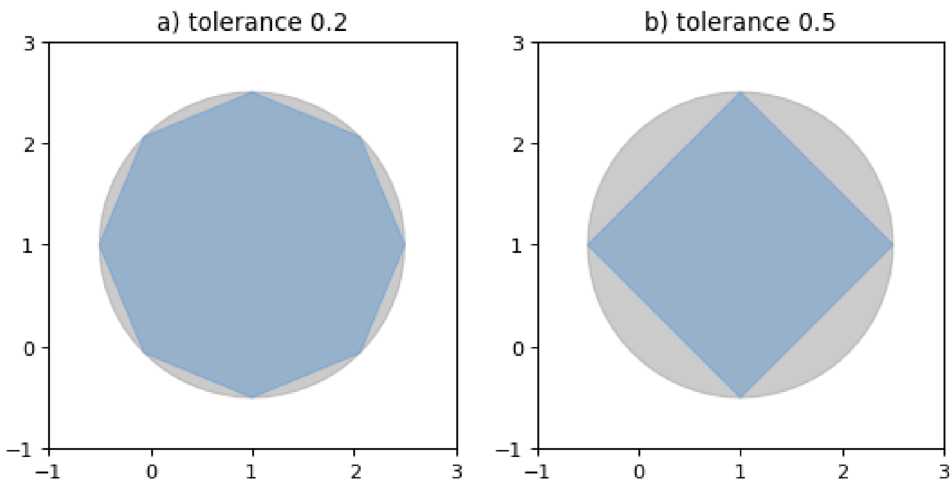

Figure 2. Simplification of a circular polygon using a tolerance of 0.2 (left) and 0.5 (right). [10]

To evaluate the impact of different simplification values on the intersection calculation, I selected the "water" and "forest" layers as they represent the largest layers. I retained only polygon objects in the layer to ensure accuracy. I loaded and projected the layers into EPSG:23700 and then performed a union operation to eliminate overlapping areas resulting from multiple tag queries. I applied a specified radius to buffer the objects, resulting in the creation of a MP object. Subsequently, I simplified the MP object using values of 0.1, 0.2,

0.5, 1, 2, 5, and 10 metres. Finally, the intersection was checked five times in a for loop, and the average was calculated to assess the effectiveness of each simplification value.
The measurements were conducted on a computer with an Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz processor, 8.00 GB of RAM, and a 64-bit operating system with an x64-based processor.

4.2.4 Air quality

The air quality index (AQI) is used to report the average air quality, and the GRID dataset used in this study was obtained from the Hungarian Meteorological Service. The dataset, which covers the period between 2018 and 2022, provides the yearly average of CO, O3, NO2, PM10, and PM2.5 values for each region. The average values were calculated by region for each year. The raw values for the five variables were used to calculate the AQI, and the classification was based on the Hungarian Meteorological Service guidelines [7]. It is worth mentioning that different range classifications can be used by different state actors [4, 5, 6, 7]. The dataset is stored in NetCDF format [3].

4.2.5 Further data preparation

Lakes and rivers were integrated into a single layer, while train and bus stops for public transportation were combined into a unified layer. The dataset of active bus stops used in this study was obtained from Volan, the primary public transport provider for buses in Hungary [27]. In addition, average prices per square metre were projected for various types of villages in Hungary's regions. The prices were originally denominated in Hungarian forints, and converted to euros based on the 2022 average exchange rate. The Hungarian Central Statistical Office was the primary source of the 2022 price data [26, 35].
In the forest layer, all regions with an area less than 5000 square metres were removed from the layer [28]. The grid dataset [36] was used to calculate the average daily means for wind-related options between 2001 and 2022.

4.3 Dashboard

The dashboard [25, 41] , implemented using the open-source Python library Streamlit [8], provides a comprehensive platform for exploring ideal places based on the research outcomes. Streamlit is specifically designed to create customised web applications for machine learning and data science purposes, making it an ideal framework for developing the dashboard.

Within the dashboard, users have the ability to define specific criteria and ranges for various features, enabling them to specify desired proximity and refine their search for suitable areas. The interface presents a comprehensive list of amenities gathered from the survey, supplemented by statistical and weather data for users to investigate further. Additionally, users can set preferences for weather data by defining a range of interest.

One notable feature of the dashboard is the option for users to interactively draw on the map and define their areas of interest. The dashboard calculates and visualises the intersection areas resulting from the combination of multiple user-defined constraints. Two distinct views are available: the simplified view, which focuses solely on the resulting area, and the detailed view, presenting a multimap with overlapping layers representing each constraint. Users can selectively adjust the visibility of different layers according to their specific requirements.

To enhance usability, the dashboard is divided into seven sections, each comprising a collection of subsections related to the main theme. For example, the amenities section encompasses options related to various amenities such as hospitals, playgrounds, and more. Similarly, the nature section offers layers representing mountains, rivers, and forests, among others.

By organising the dashboard into sections and subsections, users can navigate through the interface and easily access the desired information and functionalities based on their specific needs and preferences. The progress bar serves as a visual indicator, providing users with real-time feedback on the ongoing process and enhancing their experience while exploring ideal places.

During the initialization process of the dashboard, data reading is not performed. However, when the user clicks the "submit" button, the code follows a series of coherent steps. Firstly, it examines the layers that are relevant to the user, with the "include" button determining whether a layer should be read or excluded. Then, based on the specified range value, the code reads the corresponding layer. If the user selects the "avoid" option, the code calculates the disparity between the original layer and the layer representing the region of Hungary. After reading the layer and performing any necessary calculations, the code proceeds to verify the intersection of all selected layers.

Next, the code initiates the intersection check by examining the layers. If only one layer is selected, it is plotted accordingly. However, if two or more layers are selected, the code examines the intersection of the resulting layer with the next chosen layer. This iterative

process continues until there are no more layers remaining to be examined. Once the entire process is completed, the resulting layer is plotted on a Folium map. In the case where the user opts for the "simplified view", multiple layers are displayed on the right side, while the resulting layer is prominently showcased on the left side.
It is worth noting that the code ceases to check intersections if any of the resulting layers yield an empty dataframe, indicating a lack of intersection with the constraints specified by the user. In the context of a double-ended sidebar, each polygon in the layer is associated with a specific value. To refine the layer further, the code applies a filtering process to each polygon based on the minimum and maximum values specified within the given range. Only the polygons that fall within this range are retained within the layer, forming a multipolygon (MP).
Given that the primary feature of the dashboard revolves around this intersection functionality, it is essential to identify an effective method for accurate intersection calculation.

5. Results

The results section presents the findings of the study, highlighting key outcomes and insights. It includes the survey responses on ideal living and vacation destinations, accuracy evaluation of data queries and taggings, performance assessment of different methods, and the impact of simplification on analysis.

5.1 Survey

As of April 27th, 2023, the survey received a total of 31 responses. It is worth noting that from the user interface perspective, this categorization may not be significant as the user has the opportunity to select layers of interest regardless of their importance status. Table 1 shows the number of responses for an ideal place to live, categorising them as "being used" and "not being used." The highest numbers within each category are indicated in bold.

Classified	**Shops**	**Park**	**Forest**	**Public transport**	**Lake or river**	**Workplace**	**School**	**Specific place**	**Access to highway**	**Gym**
Being used	**31**	**28**	**27**	**27**	**26**	**26**	**21**	**20**	**19**	13
Not being used	-	3	4	4	5	4	10	11	11	**18**

Table 1: It presents the results of the classification process, which was based on a 5-point scale, for selecting a place to live.

The holiday preferences show similar results across all categories, except for "specific place, school, gym, and workplace," as Table 2 indicates.

Classified	**Forest**	**Lake or river**	**Park**	**Public transport**	**Shop**	**Access to highway**	**Specific place**	**Gym**	**School**	**Workplace**
Being used	**30**	**30**	**29**	**28**	**21**	**13**	5	2	1	1
Not being used	1	1	2	3	9	18	**26**	**29**	**30**	**30**

Table 2: It shows the results of the classified user responses for the criteria used in selecting a holiday destination.

I collected and classified the free text, capturing up to two mentions of criteria. Table 3 displays the criteria in ascending order based on the number of mentions.

Classified	Restaurant	Bars	Cafe	Hospital	Pub	Sport	Gas station	Clubs	Concerts
Number of mentions	5	5	4	4	4	3	**2**	**2**	**2**

Table 3: Results of the classified users’ free text responses for criteria used in selecting a vacation destination.

The classified data indicates that the remaining text classes have only one mention. Table 4 provides a summary of the results, with a focus on holiday criteria.

Classified	Beach	Restaurant	Swimming pool	Pub	Spots	Mountains
Number of mentions	5	5	4	4	4	3

Table 4: Results of the classified users’ free text responses for criteria used in selecting a vacation destination.

As this paper exclusively focuses on databases related to Hungary, the beach criterion has been excluded from the dashboard.

The respondents in the survey indicated that average air quality and temperature are important factors when choosing a place to live or spend a holiday. Additionally, free text responses revealed that other factors such as wind, precipitation, cost of living, salary, air quality, and crime rate were also marked as important. 65% of the respondents expressed a preference for living far away from given amenities. To accommodate this preference, a function was implemented in the dashboard allowing the user to specify a desired distance from amenities and natural environments such as shops, gyms, highways, educational institutions, workplaces, public transportation, parks, forests, and lakes or rivers.

5.2 Data Preparation

The accuracy evaluation of the different OSMnx taggings and queries was not within the scope of this paper. However, as an example, the number of objects returned by the “amenity=school” query was 3414, while that of “building=school” was 3391. The intersection of the two layers was 2200 objects, indicating that 3227 objects had an amenity key and 1015 had a building key without intersecting the other layer. Additionally, some schools were marked with the "building=school" key-tag pair, which is not an appropriate method to mark a school in operation.

Queries could take several minutes to complete [17]. In order to reduce the time required to open the file storing the layers, the merged layers were stored in a parquet file format, which could result in 87% smaller data files and faster read-in times compared to CSV format [18,

19]. To check the intersections of different layers, a MP object was produced as a layer instead of checking multiple objects' intersections.
The resulting queries contained three types of objects for amenities: Points, Polygons and LineStrings.Table 5 presents the count of these different object types for each layer following the UU.

	Number of different objects in the query			
Type of amenity	**Point**	**Polygon**	**LineString**	**Total number of objects**
Healthcare	1231	836	-	2067
Supermarket	996	1258	-	2254
Cafe	2130	191		2321
Park	96	5474	3	5573
Pub	6084	819	-	6903
Restaurant	7919	1736	-	9655
Water	637	17835	527	18999
Public transport	38631	184	10	38825
Forest	24	50780		50804

Table 5: Number of different types of objects in the layers

5.2.1 Unary union and buffer

I conducted measurements of calculation time and RAM usage for the public transportation, healthcare, and parks layers. Table 6 presents the collected results, indicating a decrease in calculation time for a 1000 metre buffer compared to a 10 metre buffer in the case of BO-M. However, in the case of MPB-M, buffering with a 1000 metre radius requires more time compared to the 10 metre buffer. During the analysis, it was not possible to determine which method and under what conditions resulted in lower execution time. Additionally, it appears that the MPB-M method takes significantly more time for layers with a larger number of objects. Further analysis is required to investigate these observations in more detail.

	Buffer by 10 metres			**Buffer by 1000 metres**		
Layer	BO-M	MPB-M	EMPB-M	BO-M	MPB-M	EMPB-M
Health Care	0.73	**0.27**	0.74	**0.48**	0.61	0.50
Park	3.62	**2.48**	3.86	1.99	9.31	**1.58**
Public Transport	17.00	81.00	**16.00**	**6.00**	228.00	11.00

Table 6: The time required in seconds to buffer by 10 and 1000 metres for the given layers, using BO-M, MPB-M, and EMPB-M.. The layers include healthcare, park, and public transport. The lowest results for each buffer method are highlighted in bold.

Figure 3 illustrates the RAM usage. The MPB-M method exhibits significantly higher RAM usage compared to the other methods when using a 1000 metre buffer. Conversely, for a 10 metre buffer, the RAM usage trend of MPB-M differs, displaying lower peaks compared to the two other methods.

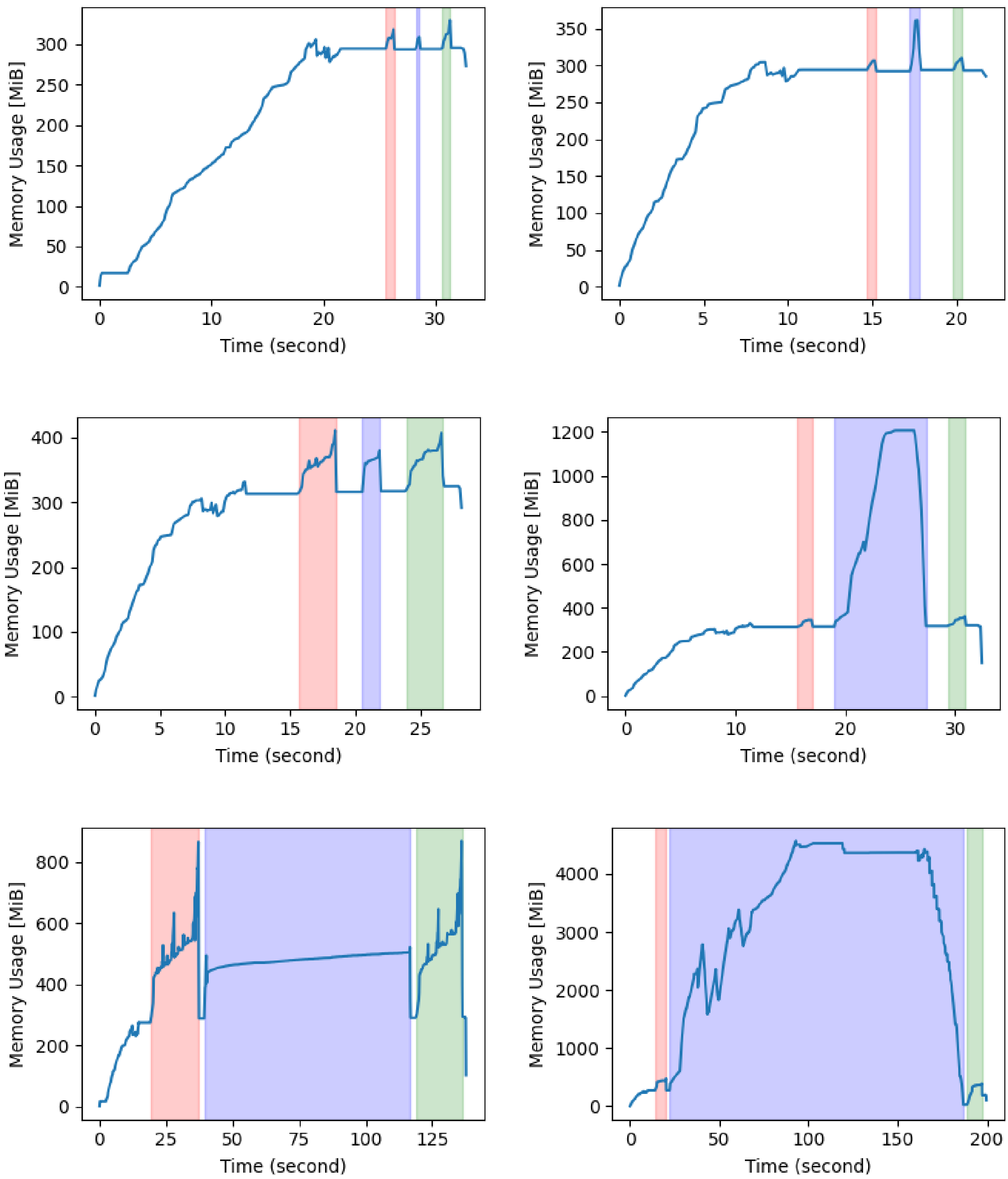

Figure 3: it presents the healthcare, park, and public transport layers in a top-to-bottom order. The figure illustrates the layer buffer values of 10 and 1000 displayed from left to right. The colour range on the figure represents the time range in which a specific function was running. The Y-axis represents memory usage in MiB, while the X-axis represents time in seconds. The red area indicates BO-M, the blue area represents MPB-M, and the green gradient represents EMPB-M.

In case of MPB-M, RAM usage and the calculation time are significantly higher then the two other methods.

Table 7 presents the timing results for the buffer and UU methods across three different layers.

	Buffer by metres and perform UU							
Layer name	1	10	20	50	100	200	500	1000
Park	4	3.48	3.22	3.01	2.9	2.31	1.8	1.59
Pub	3.56	3.09	2.84	2.57	2.38	2.16	1.7	1.66
Restaurant	3.68	3.02	2.8	2.51	2.22	1.98	1.66	1.54

Table 7: The time taken in seconds to buffer objects and perform the UU method with various buffer values.

The most time-consuming operations in the dashboard involve expanding each object in a layer and generating the MP. Among the layers used in the dashboard, the forest area has the highest number of objects and vertices, resulting in a calculation time of 1 minute. In order to reduce the calculation time, the time taken for object buffering is also measured. A buffer value of 1000 metres was chosen, and Table 8 demonstrates that as the number of objects increases, the buffer time also increases.

	Name of the layer							
Measure name	cafe	supermarket	healthcare	pub	park	restaurant	water	forest
Time (s)	0.06	0.13	0.14	0.23	0.63	0.34	7.7	81.28
Number of objects	2321	2254	2067	6903	5573	9653	18999	50804

Table 8: It shows the time required in seconds to perform a 1000 metre buffer for each object in the layer without storing the result. The "Time" row displays the buffering time, while the "Number of objects" row indicates the count of objects in the layer.

5.2.2 Simplification

Table 9 presents the properties of the water and forest layers for different simplification values. It is observed that the area, compared to the ground truth, decreases as the simplification value increases. This decrease in area can lead to inaccurate results when calculating intersection areas for multiple layers, which is an important consideration for further manipulations.

	Forest			Water		
Simplify value	layer area after simplification in square km	number of vertices after simplification	area change in percent	layer area after simplification in square km	number of vertices after simplification	area change in percent
0.1	21084	2041284	100.00	1927	500944	100.00
0.2	21084	2023918	100.00	1927	488890	100.00
0.5	21084	1976059	100.00	1927	456449	100.00
1	21084	1901327	100.00	1927	412488	100.00
2	21083	1766699	100.00	1927	351772	99.98
5	21082	1445089	99.99	1925	257592	99.91
10	21078	1136421	99.97	1922	195766	99.75

Table 9: It presents the forest and water layers without any expansion. The simplify column displays the degree of simplification applied. The number of vertices indicates the count of all objects' vertices in the resulting layer after simplification. The area change in percentage column shows the difference in percentage of the result layer compared for the original unsimplified layer. In the result section, the table lists the number of objects, time required to calculate the intersection of the two layers, the result layer's area in square kilometres, and the difference compared to the unsimplified layer in square metres.

The use of a 10 metre simplification resulted in a computation time of 4.35 seconds for the intersection of the water and forest layers, as shown in Table 10. However, this time was not significantly improved compared to the 6.25 seconds required without simplification. Notably, when a 1 metre simplification was applied, an interesting observation was made. Despite the expected calculation time of between 5.94 and 5.55 seconds, the method took 13 seconds, deviating from the expected trend. Increasing the simplification value results in larger intersection areas.

	Result			
Simplify value	Number of objects	time to calculate intersection (s)	Result layer area is square metre	Result different compared to the ground truth (square metre)
0.1	97667	6.23	11	162
0.2	95455	6.09	11	702
0.5	89513	5.94	11	6126
1	80839	**13.04**	11	25457
2	68228	5.55	11	113203
5	46974	5.00	12	719740
10	32153	4.35	14	2307531

Table 10: Properties of the result layer obtained using different simplification values.

In the next step, each layer is buffered by 1000 metres, and a simplification method is applied to these buffered layers. Table 11 presents the properties of the resulting layers.

	Forest			Water		
Simplify value	layer area after simplification in square km	number of vertices after simplification	area change in percent	layer area after simplification in square km	number of vertices after simplification	area change in percent
0.1	74669	132691	100.00	32876	397341	100.00
0.2	74669	131634	100.00	32876	387162	100.00
0.5	74669	128207	100.00	32876	365652	100.00
1	74669	123044	100.00	32875	345459	100.00
2	74668	115496	100.00	32873	324310	99.99
5	74637	75923	99.96	32831	220419	99.86
10	74594	52288	99.90	32769	150924	99.68

Table 11: Intersection and simplification results when both layers are expanded by 1000-1000 metres.

Table 12 compiles the properties of the resulting layer when the forest and water layers are expanded by 1000 metres each. A difference of 108 square kilometres was observed with a 10 metre simplification when both layers were expanded by 1000 metres. However, the calculation time for finding the intersection of the two layers did not exhibit a significant reduction with larger simplifications.

	Result			
Simplify value	Number of objects	time to calculate intersection (s)	Result layer area is square metre	Result different compared to the ground truth (square metre)
0.1	2711	0.47	25909	-12801
0.2	2711	0.46	25908	-61148
0.5	2711	0.45	25908	-317372
1	2711	0.44	25908	-894593
2	2711	0.42	25906	-2598591
5	2712	0.36	25863	-46000000
10	2711	0.31	25801	-110000000

Table 12: The percentage change in area values was calculated by comparing the ground truth values with the area of the result layer after applying the given simplification value. The area change in percent values represent the percentage difference between the ground truth area and the result layer area after applying a given simplification value.

In response to the significant observation of a 13-second calculation time during the intersection process for 1 metre simplification without expanding layers, as detailed in Table

10, I conducted a comprehensive investigation to evaluate the intersection time across additional simplification values. Specifically, I selected simplification values of 0.9, 1, and 1.1 metres and determined the intersection values through a repetitive 5 times loop. The outcomes of this investigation are visually presented in Figure 4.

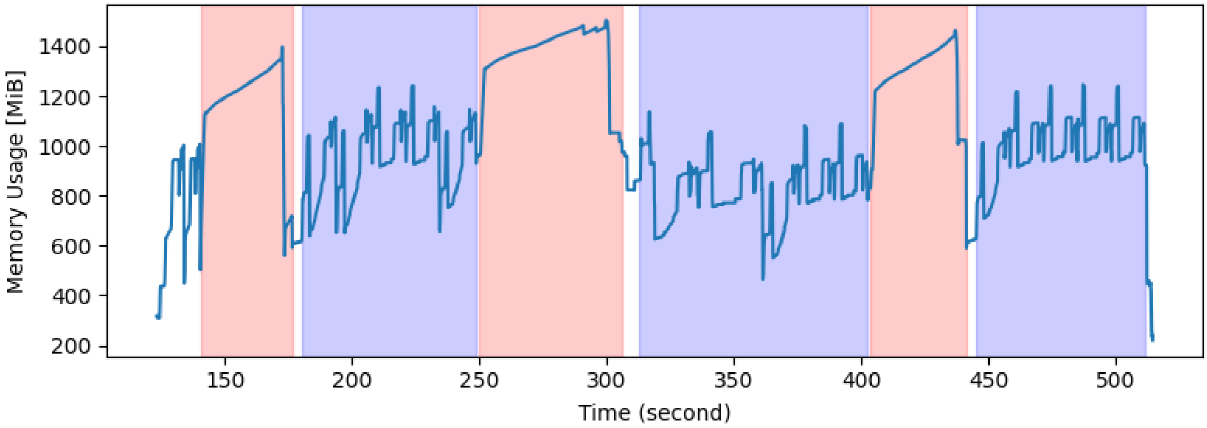

Figure 4: The RAM profile illustrates the intersection process for simplification values of 0.9, 1, and 1.1 metres, each with a 5 times loop. The red areas represent the initialization of new simplified layers, while the blue areas indicate the 5 times loop for determining the intersection of the two layers.

The results obtained from the investigation reveal intersection times of 13.6 and 13.2 seconds for the respective simplification values. These findings provide valuable insights into the computational requirements associated with different levels of simplification. It is worth noting that throughout the evaluation, the RAM usage remained within the constraints of the maximum available RAM, ensuring the stability and efficient execution of the algorithm. This emphasises the robustness and reliability of the implemented methodology.

5.2.3. Air quality

The average air quality index was calculated based on five variables, including O3, PM10, PM2.5, CO, and NO2. The classification of the raw values into the AQI was carried out according to the guidelines of the Hungarian Meteorological Service [4]. While the guideline suggested taking SO4 into consideration, the values were consistently classified as "excellent" [7], rendering their use unnecessary for the calculation of AQI. The unit of measurement was in ug/m3, with the exception of CO, which was measured in ppb. To convert CO values, European Commission directives at 20°C were used, where 1 ppb = 1.15 µg/m3 [5]. The results revealed that the AQI was classified as "good" throughout the entire area of Hungary, except for Budapest.

It should be noted that the classification of raw values may vary among different organisations, and since the average AGI is classified as "good" except for the area of Budapest, the use of AQI-based classification was not incorporated into the dashboard.

5.3 Dashboard

Figure 5 visually presents the final structure of the dashboard, depicting the arrangement of the main sections in a two-column format. This layout aims to provide users with a comprehensive overview of the dashboard's content and functionality. To further enhance the understanding of how the dashboard operates, a video demonstrating its functionality is available on YouTube [39].

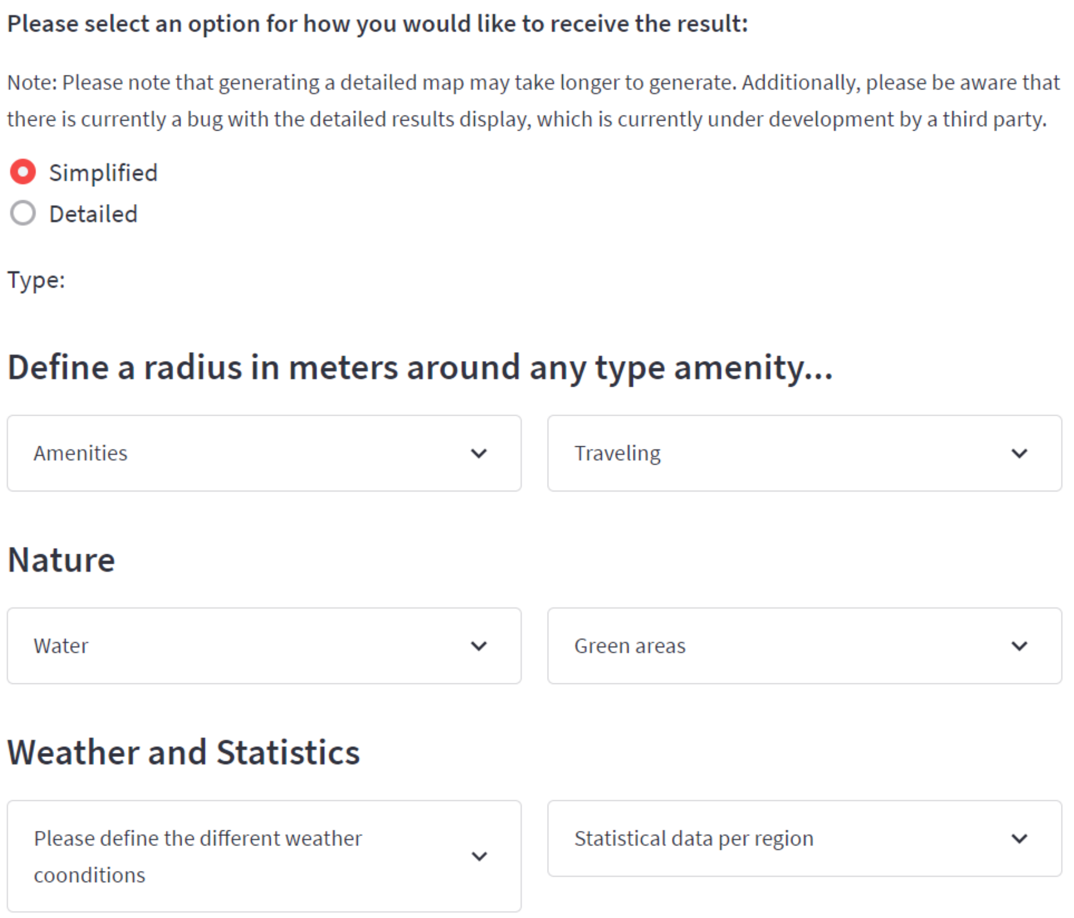

Figure 5: A snippet about the structure of the dashboard

In Figure 6, an example is shown of how the code appears when the double-ended sidebar is utilised. On the other hand, Figure 7 illustrates the appearance of the code when the single sidebar is employed.

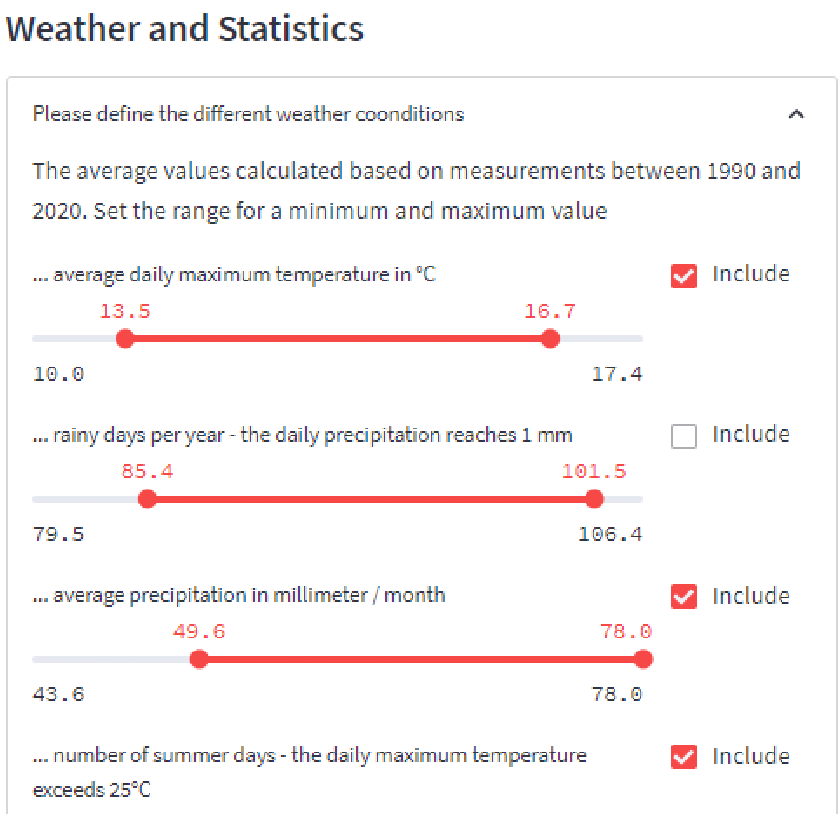

Figure 6: Double ended sidebar options

Figure 7: Single sidebar options of the dashboard

Figure 8 provides a snippet demonstrating an example result in simplified view, while Figure 9 showcases the detailed result. When utilising the code in simplified view, it automatically zooms in on the area determined by the bounds of the results layer, facilitating users in identifying specific areas. Conversely, in detailed view, the code consistently displays the entire region of Hungary.

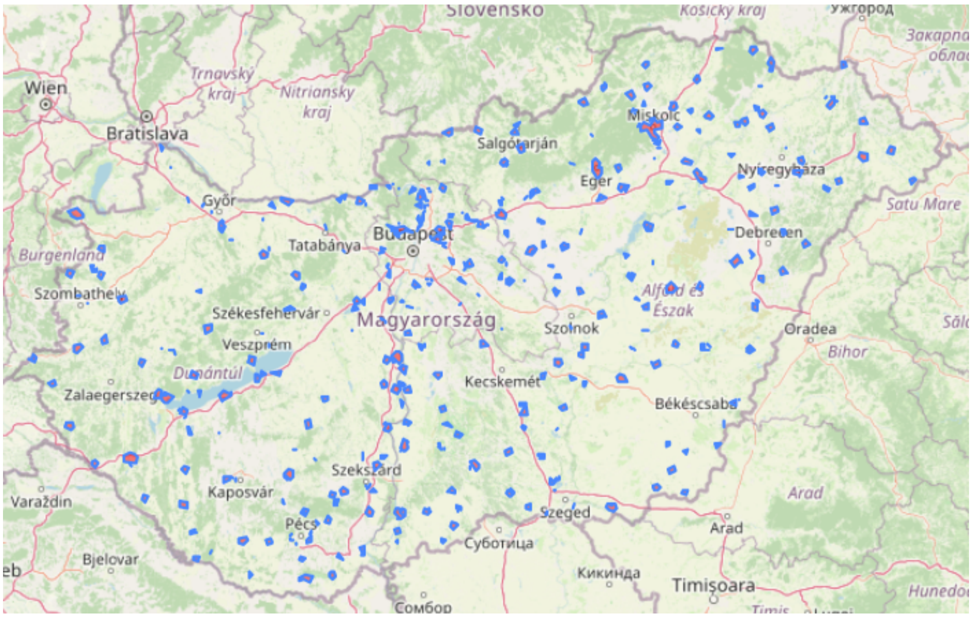

Figure 8: The figure depicts the output of an example query, showing the regions where the average housing prices per square metre are below 1000 EUR, the average wind speed is less than 3m/s, and there are supermarkets, gas stations, train or bus stops, rivers or lakes, and forests located within a radius of 2 km.

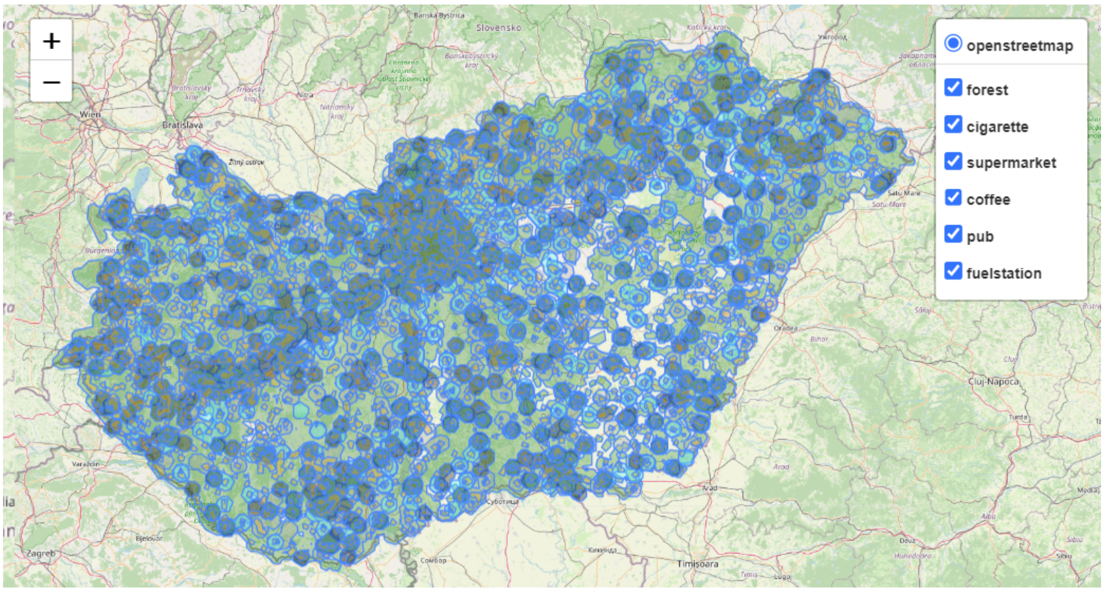

Figure 9: It illustrates the detailed result. On the right side, users have the option to toggle the visibility of different layers, providing a means to refine the search criteria.

To streamline the data manipulation process and optimise the performance of the dashboard, a decision was made to utilise pre-defined layers. For instance, a layer named "forest" was created to encompass all forest objects within the study area. Subsequently, each individual object within the "forest" layer underwent a buffering operation at various distances, including 1, 10, 20, 50, 100, 200, 500, 1000, 2000 and 5000 units. These buffered layers were then saved as separate files denoted by names such as "forest_1," "forest_10," and so on, each corresponding to the respective buffer distance. By employing this approach, the dashboard is able to efficiently determine the intersection of desired layers, allowing for real-time intersection calculations only when the user interacts with the dashboard. This methodology ensures that the dashboard operates smoothly, reducing processing time while providing seamless functionality for examining layer intersections.

By arranging the layers in ascending order based on their covered areas, the algorithm further enhances code efficiency when handling user-defined constraints. This sorting strategy allows the intersection method to commence by examining the layer with the smallest area first. As the final result layer must cover an area that is equal to or smaller than the first layer, the algorithm can eliminate the need for further calculations involving larger layers. Consequently, this strategy significantly reduces computation time and enhances the overall performance of the algorithm.

The processing time required for calculating the intersection of multiple layers and plotting them on a map depends on the user's constraints and the chosen method, whether it is a simplified or detailed view. Generating a detailed view significantly increases the processing time due to the involvement of multiple layers that need to be plotted on a single map. This complexity arises from the large number of points and the necessity to create an HTML file for displaying the results.

6. Discussion

The present paper provides additional support for existing traditional place satisfaction studies, but also serves as an extension of them, due to the fact that it uses a partially free-form survey. Therefore, I could explore the value attributions of users to various needs and preferences of amenities, nature- and weather-related aspects. By gaining a deeper understanding of different aspects that influences place satisfaction, I could build a dashboard

that takes into consideration the perspectives of the users when spotting an ideal place to live or to go on vacation.
Additionally, the uniqueness of the dashboard lies in the fact that it is built upon a code that takes into consideration the intersecting geospatial layers and user needs and preferences when spotting ideal places to live and to go on vacation to in Hungary.
This paper solely focuses on the manipulation of geospatial data and the dashboard within the geographical scope of Hungary, which spans a total area of 93,000 km2. It offers a solid foundation that could be further used for expanded geospatial scopes. However, it must be noted that for larger areas, it becomes necessary to apply more efficient methods to build a dashboard that can handle the spatial complexities of larger regions. Therefore, for future, larger research projects, indexing techniques [31, 32] are indispensable for larger-scale analyses, and SQL environments, such as PostGIS and mongoDB offer promising solutions for geospatial data manipulation [33].
Furthermore, the current dashboard not only enables users to find places based on user-defined constraints, but it also has the potential to be useful in a variety of fields, including real estate and urban planning. For example, it could aid in analysing prices and demand in the real estate market, as well as in determining optimal locations. In urban planning, it could be used to analyse urban areas or plan infrastructure. With extended functionality, the dashboard could also be used for deeper analysis of regions. Real estate agents may find the features of the dashboard useful for assessing properties, while agricultural actors could use it to find places based on weather conditions. Furthermore, it could prove useful in any field requiring spatial analysis.

7. Limitations

Regrettably, time constraints hindered my capacity for data processing, resulting in the omission of classifications for responses indicating a few user needs and preferences, such as proximity to the crime rate, availability of concerts, clubs, gym, and football fields. Therefore, further adjustments are needed in the dashboard to better meet the needs of users.
The inaccuracy of the results generated by the dashboard can be attributed to the inaccurate tagging of amenities data in OSMnx. This inaccuracy stems from inadequate tagging or the omission of objects within the amenities data. The majority of amenities are marked as points, which may not accurately represent the corresponding building objects. Additionally,

it is important to note that users may expect information about building entrances, which is not included in the OSMnx queries [29].

The option to view average house prices per square metre may also be prone to inaccuracies due to the transformation of data from the more generalised KSH database [26]. While utilising the real estate company's frequently used database [30] could potentially yield more reliable results, access to this data is currently restricted.

The weather-related options in the dashboard incorporate average values from the past 20 or 30 years. Given the current phenomenon of climate change [40], it may be more informative for users seeking an ideal place to live in the future to include predicted values for future weather conditions.

Regarding the predefined distance range for the different layers, which is displayed in a single sidebar, there is a limited amount of information available concerning the suitable values. Moreover, extending the ranges beyond 5 kilometres has the potential to improve the user experience and yield more valuable insights.

Challenges in terms of computation time arose when users opted for the "avoid" option, as it necessitated comparing the original layer to the entirety of Hungary, resulting in prolonged processing durations.

8. Conclusion

In conclusion, this paper introduces a geospatial dashboard that empowers users to conduct customised place searches based on their specified criteria. The implementation of the dashboard involved the selection and utilisation of various place attributes based on user feedback, leveraging Python's Shapely and GPD libraries. Throughout the development process, performance and accuracy metrics were assessed, focusing specifically on polygon simplification, buffer, and unary union methods.

The evaluation results revealed that the unary union operation, performed after expanding each object in the layer, consumed the most computational time. To optimise computation time, pre-defined layers were generated for a specified range. Although layer simplification did not significantly impact calculation time, it could lead to inaccuracy. The implemented code exhibited exceptional computational efficiency, even when handling the maximum number of layers and range.

Overall, I have obtained comprehensive results demonstrating that predefined layers must be considered for decreasing computational processing time when building a dashboard that intersects several MP objects. Additionally, this paper highlighted the importance of exploring user needs and preferences related to place satisfaction when building a localised geospatial dashboard, based on intersecting several MP objects.

9. References

[1] Survey best practices & design guidelines. SurveyMonkey. (n.d.). Retrieved March 11, 2023, from https://www.surveymonkey.com/mp/survey-guidelines/

[2] Boeing, G. (2017). OSMnx: New methods for acquiring, constructing, analyzing, and Visualizing Complex Street Networks. *Computers, Environment and Urban Systems, 65*, 126–139. https://doi.org/10.1016/j.compenvurbsys.2017.05.004

[3] sohrab4748. (n.d.). SOHRAB4748/AgriMetSoft: Software Tools of agrimetsoft. https://agrimetsoft.com. GitHub. Retrieved March 13, 2023, from https://github.com/sohrab4748/AgriMetSoft

[4] Hungarian Air Quality index. Országos Légszennyezettségi Mérőhálózat. (n.d.). Retrieved March 13, 2023, from https://legszennyezettseg.met.hu/en/air-quality/aq-index-information

[5] Heinecke, R. (2022, August 3). Air pollution - how to convert between Mg/m3, µg/m3 and PPM, PPB. Breeze Technologies. Retrieved March 13, 2023, from https://www.breeze-technologies.de/blog/air-pollution-how-to-convert-between-mgm3-%C2%B5gm3-ppm-ppb/ 7

[6] European City Air Quality Viewer. European Environment Agency. (2023, February 7). Retrieved March 13, 2023, from https://www.eea.europa.eu/themes/air/urban-air-quality/european-city-air-quality-viewer

[7] 2021. évi összesítő értékelés Hazánk levegőminőségéről AZ automata ... (n.d.). Retrieved March 13, 2023, from https://legszennyezettseg.met.hu/storage/media/ertekelesek/2021_automata%20ertekeles.pdf

[8] Streamlit • the fastest way to build and share data apps. (n.d.). Retrieved March 19, 2023, from https://streamlit.io/

[9] Reja, U., Manfreda, K. L., Hlebec, V., & Vehovar, V. (2003) Open-ended vs. close-ended questions in web questionnaires.

https://www.researchgate.net/publication/242672718_Open-ended_vs_Close-ended_Questions_in_Web_Questionnaires

[10] Gillies, S., & others. (2007). Shapely: manipulation and analysis of geometric objects. Retrieved from "https://github.com/Toblerity/Shapely"

[11] "OpenStreetMap contributors. (2015) Planet dump [Data file from $date of database dump$]. Retrieved from https://planet.openstreetmap.org."

[12] Tags. Tags - OpenStreetMap Wiki. (n.d.). Retrieved April 27, 2023, from https://wiki.openstreetmap.org/wiki/Tags

[13] Jordahl, K., Bossche, J. V. den, Fleischmann, M., McBride, J., Wasserman, J., Richards, M., Badaracco, A. G., Snow, A. D., Gerard, J., Tratner, J., Perry, M., Ward, B., Farmer, C., Hjelle, G. A., Cochran, M., Taves, M., Gillies, S., Caria, G., Culbertson, L., … Mesejo-León, D. (2022a, October 24). *Geopandas/geopandas: V0.12.0*. Geopandas. http://doi.org/10.5281/zenodo.7243632

[14] Klimadat Térinformatikai Rendszer. KLIMADAT Climate data Geospatial information system. (n.d.). Retrieved April 27, 2023, from https://klimadat.met.hu/

[15] Brovelli, M., & Zamboni, G. (2018). A new method for the assessment of spatial accuracy and completeness of OpenStreetMap building footprints. ISPRS International Journal of Geo-Information, 7(8), 289. https://doi.org/10.3390/ijgi7080289

[16] Ciepłuch, B., Jacob, R., Mooney, P., & Winstanley, A. C. (2010, July 20). Comparison of the accuracy of OpenStreetMap for Ireland with google maps and bing maps. Comparison of the accuracy of OpenStreetMap for Ireland with Google Maps and Bing Maps. Retrieved April 28, 2023, from https://mural.maynoothuniversity.ie/2476/

[17]Overpass API. Overpass API - OpenStreetMap Wiki. (n.d.). Retrieved April 28, 2023, from https://wiki.openstreetmap.org/wiki/Overpass_API

[18] Spicer, T. (2022, July 18). Apache parquet: How to be a hero with the open-source columnar data format. Medium. Retrieved April 28, 2023, from https://blog.openbridge.com/how-to-be-a-hero-with-powerful-parquet-google-and-amazon-f2ae0f35ee04

[19] Sarkar, T. (2021, October 28). How fast is reading Parquet File (with arrow) vs. CSV with pandas? Medium. Retrieved April 28, 2023, from https://towardsdatascience.com/how-fast-is-reading-parquet-file-with-arrow-vs-csv-with-pandas-2f8095722e94

[20] Memory-profiler. PyPI. (n.d.). Retrieved April 28, 2023, from https://pypi.org/project/memory-profiler/

[21] Weather. Hungarian Meteorological Service (OMSZ) (2023, April 28). Retrieved April 28, 2023, from https://www.met.hu/en/idojaras/

[22] Visvalingam, M., & Whyatt, J. D. (1990). The Douglas-Peucker Algorithm for line simplification: Re-evaluation through visualisation. *Computer Graphics Forum, 9*(3), 213–225. https://doi.org/10.1111/j.1467-8659.1990.tb00398.x

[23] Kaur, J., & Singh, J. (2018). An automated approach for quality assessment of OpenStreetMap Data. 2018 International Conference on Computing, Power and Communication Technologies (GUCON). https://doi.org/10.1109/gucon.2018.8674899

[24] (https://www.klokantech.com/), K. T. G. H. (n.d.). EPSG:23700. HD72 / EOV. Retrieved May 1, 2023, from https://epsg.io/23700

[25] Spot ideal places in Hungary. (n.d.). Retrieved May 4, 2023, from https://spot-ideal-places-to-live-in-hungary.streamlit.app/

[26] 18.1.2.9. mean price per sqm by region and settlement type [thousand huf]. 18.1.2.9. Mean price per sqm by region and settlement type. (n.d.). Retrieved May 1, 2023, from https://www.ksh.hu/stadat_files/lak/en/lak0025.html

[27] GTFS - Menetrendek. Volánbusz. (n.d.). Retrieved May 1, 2023, from https://www.volanbusz.hu/hu/menetrendek/gtfs

[28] Terms and definitions. (n.d.). Retrieved May 1, 2023, from https://www.fao.org/3/I8661EN/i8661en.pdf

[29] Hu, X., Noskov, A., Fan, H., Novack, T., Li, H., Gu, F., Shang, J., & Zipf, A. (2021). Tagging the main entrances of public buildings based on OpenStreetMap and binary

imbalanced learning. *International Journal of Geographical Information Science*, 1–29. https://doi.org/10.1080/13658816.2020.1861282

[30] *Real estate statistics in Hungary*. Ingatlannet.hu. (n.d.). Retrieved May 4, 2023, from https://www.ingatlannet.hu/statisztika

[31] *Performance Analysis of MongoDB Vs. PostGIS/PostGreSQL Databases For Line Intersection and Point Containment Spatial Queries*. (2015). Retrieved May 4, 2023, from https://doi.org/10.7275/R5MG7MQG

[32] *Indexing postgis databases and spatial query performance evaluations*. (n.d.). Retrieved May 4, 2023, from https://www.researchgate.net/publication/294782438_Indexing_PostGIS_databases_and_spatial_Query_performance_evaluations

[33] Bartoszewski, D., Piorkowski, A., & Lupa, M. (2019). The comparison of processing efficiency of spatial data for postgis and mongodb databases. *Beyond Databases, Architectures and Structures. Paving the Road to Smart Data Processing and Analysis*, 291–302. https://doi.org/10.1007/978-3-030-19093-4_22

[34] OpenAI. (2023). ChatGPT (Mar 14 version) [Large language model]. https://chat.openai.com/chat

[35] *List of cities in Hungary*. Detailed Gazetteer of Hungary. (n.d.). Retrieved May 5, 2023, from https://www.ksh.hu/apps/hntr.egyeb?p_lang=EN&p_sablon=LETOLTES

[36] *Homogenised gridded data series*. ODP. (n.d.). Retrieved May 7, 2023, from https://odp.met.hu/climate/homogenized_data/gridded_data_series/

[37] Källström, L., & Hultman, J. (2019a). Place satisfaction revisited: Residents' perceptions of "A good place to live." *Journal of Place Management and Development*, *12*(3), 274–290. https://doi.org/10.1108/jpmd-07-2017-0074

[38] *OpenStreetMap - Tags*. Tags - OpenStreetMap Wiki. (n.d.-a). https://wiki.openstreetmap.org/wiki/Tags

[39] Gyori, G. (2023, May 8). *Spot ideal places in Hungary - bachelor project*. YouTube. https://www.youtube.com/watch?v=njxK7QAPoFs

[40] Thornton, P. K., Ericksen, P. J., Herrero, M., & Challinor, A. J. (2014). Climate variability and vulnerability to climate change: A Review. *Global Change Biology*, *20*(11), 3313–3328. https://doi.org/10.1111/gcb.12581

[41] Gyori G.. (n.d.). *CSIPAPICSA/Bachelor-Project: Spot the Best Places to Live*. GitHub. https://github.com/csipapicsa/Bachelor-Project

[42] *Deciding where to live – 10 factors to help you choose*. Charles Cameron & Associates. (2023, March 1). https://ccameron.co.uk/hub-post/deciding-where-to-live-10-factors-to-help-you-choose/

[43] Reback, J., McKinney, W., Jbrockmendel, Bossche, J. V. den, Augspurger, T., Cloud, P., Gfyoung, Sinhrks, Klein, A., Hawkins, S., Roeschke, M., Tratner, J., She, C., Ayd, W., Petersen, T., MomIsBestFriend, Garcia, M., Schendel, J., Hayden, A., … Winkel, M. (2020, June 17). *Pandas-dev/Pandas: Pandas 1.0.5*. Zenodo. https://zenodo.org/record/3898987

[44] *Python.org*. Python.org. (n.d.). https://www.python.org/

10. Appendix

10.1 Dashboard.zip

The file contains the code for implementing the dashboard discussed in this study. Please note that the code is provided as-is and may require additional dependencies or modifications for specific computing environments. The "data" folder must be downloaded; the link is included in the "info.txt" file.

10.2 Analysis.zip

The file contains two codes: "memory_profiler_intersection_unary_union-final.py" measures the time required to calculate the intersection between two layers with different simplification values, and "unary_union_memory_profiler.py" measures the RAM usage and time taken for different buffering methods.