Happy-Dining: Datadriven approach to screen restaurants

Kai Hinrichs

2596956

Seminararbeit

Lehrstuhl für Wirtschaftsinformatik und Business Analytics
Universität Würzburg

Betreuer: Prof. Dr. Gunther Gust

Assistent: Deep Poja

Würzburg, den 08.12.2023

Contents

List of Figures

List of Tables

Abstract

This paper presents the development of a restaurant recommendation system using Python, incorporating advanced techniques such as Gower distance calculation, K-means clustering, and standard score. The recommendation system utilizes a Graphical User Interface (GUI) built with Tkinter, allowing users to input their preferences for an ideal restaurant. The system uses K-means clustering for grouping similar restaurants, Gower distance to measure dissimilarity between restaurants, and standard score to eliminate recommendation bias. The evolution of the code is discussed, showcasing versions with increasing complexity. Overall, the system offers a user-friendly experience, suggesting restaurants based on user preferences and enhancing decision-making in restaurant selection.

1 Introduction

1.1 Background

In the search for new dining experiences, the digital era has guided people to rely on online reviews and recommendations from others. Platforms such as Google, Yelp, and TripAdvisor have become indispensable in helping users discover and assess local restaurants, relying on the insights of other diners. Navigating this information-rich landscape, machine learning models emerge as a pivotal yet often unacknowledged player, silently revolutionizing daily activities.

The continuously growing size of the internet, specifically machine learning, has seamlessly integrated into everyone's routines. From personalized shopping suggestions to tailored content recommendations, machine learning algorithms have become adept at understanding user preferences and societal trends.

Depending on reviews of other people and groups, restaurants could leave the guest disappointed after the visit, due to the food, ambiance, or service because they don't have enough votes, and thus need more votes to have an authentic rating and tell the diner if he should visit the restaurant. To address and solve the common issue of post-restaurant visit disappointment, this work introduces a Python-based recommendation system centered on user preferences. This machine learning model will provide the user with the possibility to input their own preferences and get based on multiple factors like Gower distance, K-means clustering, and standard score a perfect restaurant that applies solely to the user's preferences. Using these methods, the system aims to minimize the chances of restaurant visits resulting in disappointment, considering factors like food quality, service, ambiance, and more.

1.2 Motivation and Objective

Since AI has become a huge part of everyone's daily lives choosing the "Chair of Computer Science VI - Artificial Intelligence and Applied Computer Science" was a strategic decision to research a specific subject for the seminar work. Since the skills of programming are heavily required in many jobs that come with studying Economics, it is intriguing to explore programming through the development of a compact application designed to assist individuals in selecting a restaurant. Acquiring new knowledge and understanding deep mechanics and machine learning models will help to solve problems that could potentially come up in the future.

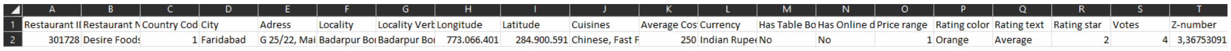
Furthermore, the objective of the work is to get a deep understanding of how the Gower distance, K-means clustering, and standard score work and how this can be connected to build a recommendation system based on the user's preferences in Python. The ultimate goal is to produce a functional program accessible to anyone for personal use.

2 Data and Program Aquisition

2.1 Selection of Datasets and Program

To facilitate the development of the recommendation system, a dataset comprising restaurant information was essential. This dataset was compiled in Excel and is available on Kaggle, sourced from the Zomato API (Mehta, 2018). Each restaurant is depicted with the restaurant ID, the name of the restaurant, the country code, the address, the locality, locality verbose, longitude, latitude, cuisines, average cost for two, currency, the ability to book a table, ability to get the food delivered, and price range. Furthermore, a rating color indicates the quality ranging from white which means the restaurant is not rated, to red for poor ratings, orange for average, yellow for a good rating, green for very good, all the way to dark green indicating perfect service and food. Additionally, the restaurant's info also contains a rating text, a rating star starting from 0 when the restaurant is not rated, to 1 if it is not good, and up to 5 signifying excellent service, food, and ambiance, votes, as well as the Z-number which puts the number of votes and rating in ratio, and will be discussed further in section 6.2. The sheet has 7944 unique Restaurants situated throughout New Delhi and its surroundings.

With this sheet, a good overview of all existing restaurants helps the creation of a recommendation program via Python. The restaurants can have multiple cuisines if offered. Since the restaurants are all located in New Delhi, India the currency used in all Restaurants is rupees. Including all this information, a line in the Excel sheet contains the following information:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Restaurant II	Restaurant N	Country Cod	City	Adress	Locality	Locality Verb	Longitude	Latitude	Cuisines	Average Cos	Currency	Has Table Bo	Has Online d	Price range	Rating color	Rating text	Rating star	Votes	Z-number
2	301728	Desire Foods	1	Faridabad	G 25/22, Mai	Badarpur Bo	Badarpur Bo	773.066.401	284.900.591	Chinese, Fast F	250	Indian Rupe	No	No	1	Orange	Average	2	4	3,36753091

Figure 1: Example of a line in the Excel sheet

For the entire creation of the recommendation system, the Windows app "Visual Studio Code" (VSC) was used since it provides the best programming experience.

2.2 Data Cleaning and Preprocessing

To rule out any useless information from the Kaggle list, slight improvements have been made. Unnecessary columns and error rows have been removed and the list has been adapted to the perfect use for the Python script.

Originating from the original Kaggle list, a nearly identical list was created, focusing solely on relevant information necessary for users to create their ideal restaurant. The reason behind removing information that seems important at first is that the user should have an easy-to-use experience and should use the program without the need for exact coordinates or a random restaurant ID. Given these considerations, only eight columns were selected to be used for the measurement of the Gower distance. Other restaurant details were deemed non-essential for accurate results. The second Excel sheet has the following columns: "City", "Locality Verbose", "Cuisines", "Average Cost for Two", "Has Table Booking", "Has Online Delivery", "Rating star", and "Votes". This is the key information someone needs when creating his "dream restaurant". Both Excel lists are accessible in this GitHub folder (Hinrichs, 2023).

3 Machine Learning Models

3.1 Essentials of Machine Learning Models

Machine learning, especially unsupervised machine learning plays a vital role in the recommendation process and can be defined as "[...] the ability to adapt to new data independently" (decypher, 2018). Machine learning models are Artificial Intelligence (C. S. Lee and A. Y. Lee, 2020) [p. 1];(Portugal, Alencar, and Cowan, 2018) [p. 2] algorithms that learn specific patterns and make predictions or recommendations without explicit programming. These models can identify specific images, text, and other data and can detect patterns that can be used to recommend specified information and data to the user (LeCun, Bengio, and Hinton, 2015) [p. 1]. For instance, "Google Lens" effectively recognizes images and offers relevant information.

Machine learning has become a huge part of everyone's daily life, because it gets used in many fields, such as ads on YouTube, predicting weather, and restaurant recommendations or recommendations in general. To get to the part, where the machine learning model can decide, predict, and recommend by itself, the model needs to be supplied with some kind of data that it can use. This data often needs data preprocessing as also done in section 2.2 with the used restaurant Excel table (L'Heureux et al., 2017a) [p. 6]. The more data machine learning models have, the better they can perform (Portugal, Alencar, and Cowan, 2018) [p. 9]. Machine learning can be split into supervised, unsupervised, and reinforced machine learning. Both supervised and reinforced machine learning models are not needed and thus only will be defined shortly in this work to provide an overview for the reader.

3.2 Supervised and Reinforced Machine Learning

Supervised machine learning can be defined as a predictive and analytical algorithm that knows both input and output and learns how to map these (L'Heureux et al., 2017a) [p. 2]. As written in the name of the model, supervised machine learning needs guidance to be trained. With the provided information of input and output this machine learning model for example can assign certain images to certain categories by analyzing certain parts of these pictures (LeCun, Bengio, and Hinton, 2015) [p. 1-3].

Additionally, reinforcement learning is a reward-based model, that can be used for building games or training robots. It can be defined as an algorithm that learns decision-making after receiving certain penalties or rewards (Li, 2018) [p. 20]. This can be compared to a self-driving car, that learns a new race track and gets punishment by either crashing or getting a treat by crossing a certain checkpoint. Reinforced machine learning doesn't need any supervision to be trained.

3.3 Unsupervised Machine Learning for K-means

Unsupervised machine learning includes the clustering of certain data based on their similarity (L'Heureux et al., 2017a) [p. 2]. Similar to reinforcement machine learning, unsupervised machine learning doesn't need any supervision to be trained. This means there is no predefined output variable, and the unsupervised machine learning can identify and recommend certain data to a user after an appropriate input. For instance, after the correct input of information about a "dream" restaurant or preferences, unsupervised machine learning can now recommend the best-fitting restaurant. In this work, the unsupervised machine learning model, K-means clustering was utilized as one of the unsupervised machine learning methods, further explained in Section 4.
Unsupervised machine learning can be divided into multiple learning techniques which consist of hierarchical learning, data clustering, latent variable models, dimensionality reduction, and outlier detection. Further explained in Figure 2 but not discussed in detail (Usama et al., 2019) [p. 3-17].

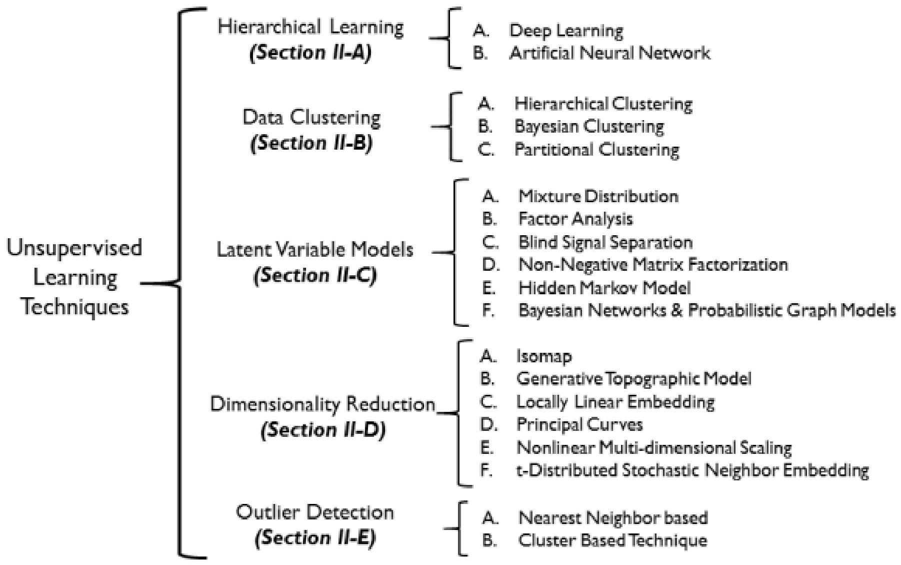

Unsupervised Learning Techniques
Hierarchical Learning (Section II-A)
A. Deep Learning
B. Artificial Neural Network
Data Clustering (Section II-B)
A. Hierarchical Clustering
B. Bayesian Clustering
C. Partitional Clustering
Latent Variable Models (Section II-C)
A. Mixture Distribution
B. Factor Analysis
C. Blind Signal Separation
D. Non-Negative Matrix Factorization
E. Hidden Markov Model
F. Bayesian Networks & Probabilistic Graph Models
Dimensionality Reduction (Section II-D)
A. Isomap
B. Generative Topographic Model
C. Locally Linear Embedding
D. Principal Curves
E. Nonlinear Multi-dimensional Scaling
F. t-Distributed Stochastic Neighbor Embedding
Outlier Detection (Section II-E)
A. Nearest Neighbor based
B. Cluster Based Technique

Figure 2: Taxonomy of unsupervised learning techniques (Usama et al., 2019) [p. 4]

4 Introduction to K-means Clustering

4.1 K-means Clustering in General

K-means clustering is a fundamental data clustering machine learning algorithm that follows a certain pattern until a dataset or image is classified. Initially, it defines k-centroids, each representing one cluster, with a preference for placing these centroids as far away from each other as possible. The usage of this machine learning algorithm is very important because now the data from the Excel sheet can be put in relation to another (Ahmed, Seraj, and Islam, 2020) [p.1-2]. The next step is to assign every point in the data set to the closest centroid. This is achieved with the Euclidean distance.

$$Dist(O_1, O_2) = \sqrt{\sum_{i=1}^{k}(O_1^i - O_2^i)^2} \tag{1}$$

as proved by Bouhmala (2016) [p. 2]

After this step, the mean of all data is recalculated and the centroids will be set at the mean, and the data points get assigned the new closest centroid (Kodinariya, Makwana, et al., 2013) [p. 2]. This process repeats until the clusters no longer change. Each iteration the sum of the variation is taken into account and will be compared to the previous iterations until the best iteration is found.
This clustering method enables the grouping and visualization of data, allowing for quick identification of similar data or facilitating the use of other clustering-based algorithms, such as the Gower distance discussed in Section 5.1. Furthermore, the combination of these two algorithms is useful, especially in the example of the approximately 8000 restaurants, the Gower algorithm doesn't need to calculate the Gower distance for all restaurants in the list, but only for those within the relevant cluster. This efficiency saves significant time and computational resources, especially when dealing with lists containing millions of entities. Even with a k value of two, the calculation time of the Gower matrix is halved. The K-means algorithm can look as follows:

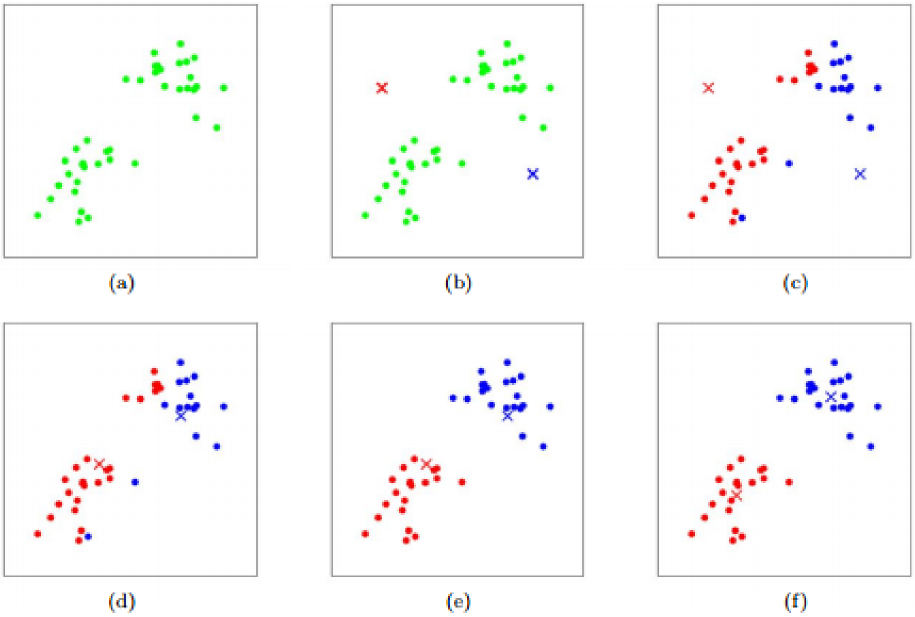


Figure 3: K-means algorithm (Piech, 2012)

In Figure 3 the K-means algorithm begins with (a) being for example all the restaurants in the dataset. In (b) 2 random, but far away from each other, cluster centroids are picked. In (c) the restaurants are assigned the nearest cluster centroid, which then gets moved to the mean of all restaurants in (d). (e-f) show the new assignment of all restaurants to their new cluster and the repositioning of the cluster to the mean. These steps will repeat until there is no longer a change (Piech, 2012).

4.2 Elbow Method

Using the Elbow method, the optimal number of clusters k can be determined. This approach is useful to find the optimal balance between the complexity of the model and the data representation. In conclusion, the Elbow method is a visual approach to guide the user in identifying the ideal k for the number of clusters. This method "calculates the squared difference of different k values.[...] Here we introduce a variable, WCSS (Within-Cluster Sum-of-Squares), which measures the variance within each cluster. The better the clustering, the lower the overall WCSS" (Cui et al., 2020) [p. 3].

$$WCSS = \sum distance(P_i, C_1)^2 + \sum distance(P_i, C_1)^2 + \sum distance(P_i, C_1)^2 \qquad (2)$$

as proved by Cui et al. (2020) [p. 3] with each $\sum distance(P_i, C_1)^2$ standing for one k.
The approach starts with a k= 1 and increases the k by one after each iteration.
The Elbow method was plotted as shown in Figure 4.

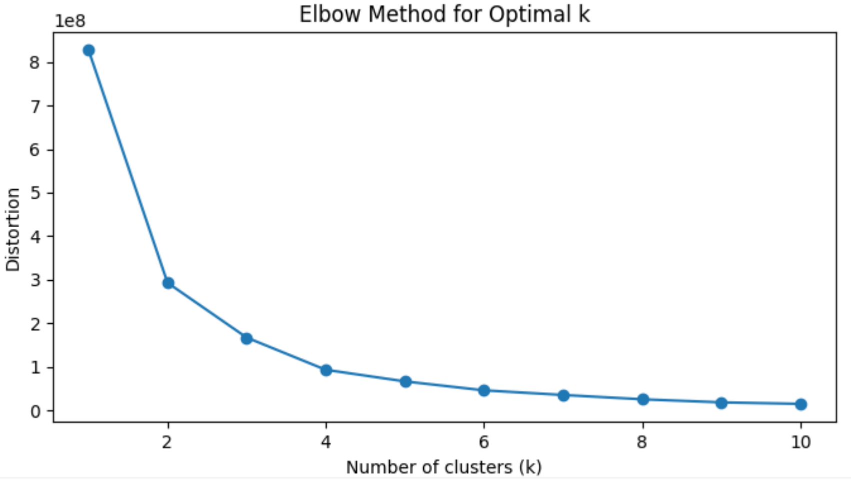


Figure 4: Elbow Method

Seeing this plot, an "elbow" becomes apparent at k = 2. This is visible by the big difference in distortions from one to two and from two to three. This shows the user the best amount of clusters that have to be used in the Elbow algorithm. However, it's worth noting that sometimes the "elbow" might not be identifiable (Kodinariya, Makwana, et al., 2013) [p. 3].

5 Introduction to Gower's Distance

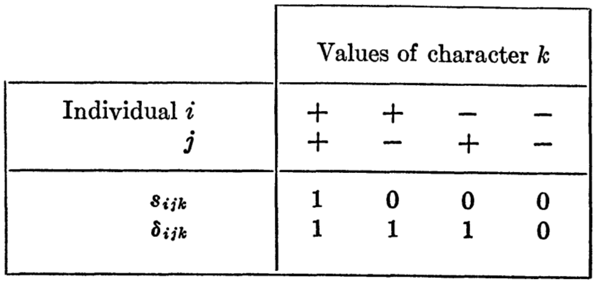
5.1 Gower Distance and its Similarity Measurement

To comprehensively compare approximately 8000 restaurants, an algorithm is essential to assess the value of each restaurant in relation to every other. Published in 1971 J. C. Gower introduced "A General coefficient of similarity and some of its properties" saying: "A general coefficient measuring the similarity between two sampling units is defined. The matrix of similarities between all pairs of sample units is shown to be positive semidefinite [...]" (Gower, 1971) [p. 857]. In short terms, Gower's distance measures the dissimilarity of two or more objects and creates a matrix showing the distance between each other.
With this paper, it is now possible to put all these Restaurants in perspective. Dichotomous, qualitative, and quantitative variates are differentiated. Dichotomous means that only one of two characteristics can apply. This includes responses like "Yes" or "No" or distinctions such as "Healthy" and "Unhealthy". Qualitative data includes elements like a restaurant name, an address, or a cuisine that is offered in a Restaurant, whereas Quantitative data includes longitude, latitude, the average cost or the number of votes (Gower, 1971) [p. 858].
When comparing two individuals, denoted as a and b, either dichotomous, qualitative, or quantitative attributes get compared on a character c. Subsequently, a score, denoted as s_{abc}, is assigned, ranging from 0 to 1, with 1 being completely dissimilar and 0 being identical. The Gower's distance can be defined as follows:

$$S_{Gower}(x_a, x_b) = \frac{\sum_{c=1}^{p} s_{abc}\delta_{abc}}{\sum_{c=1}^{p} \delta_{abc}} \tag{3}$$

as proved by Bektas and Schumann (2019) [p. 3]

With c representing the character, p the total amount of characters in the dataset and δ being the weighting factor determining whether the current descriptor should be considered in the calculation of the Gower's distance. Based on the preferences, δ could be 0 if the descriptor should not be used in the calculation and 1 if it should be used.
Four distinct combinations of unknown values for variable c are considered, and these combinations may arise for two individuals. The corresponding scores and validity are assigned to each combination, as presented in the following table.



	Values of character k			
Individual i	+	+	−	−
j	+	−	+	−
s_{ijk}	1	0	0	0
δ_{ijk}	1	1	1	0

Figure 5: Scores and validity of dichotomous character comparisons (Gower, 1971) [p. 858]

Qualitative characters are set s_{abc} = 1 if the two individuals a and b agree in the c-th character and s_{abc} = 0 if they differ. Quantitative characters with values $x_1, x_2, ..., x_n$ of character c for the complete sample of n individuals s_{abc} is set $= 1 - |x_a - x_b|/R_c$. Here, R_c represents the range of character c and the total range in the population or range in the sample. When $x_a = x_b$ then s_{abc} = 1 ,and when $x_a$and x_b are at opposite ends of their range, s_{abc} is a minimum (0 when R_c is determined from the sample). With intermediate values, s_{abc} is expressed as a positive fraction (Gower, 1971) [p.858-859]. To provide clarity, individuals a and b could be 2 Restaurants that get compared with different characters c which could be the restaurant name, number of votes, and more. Applying these methodologies in the context of the available restaurants, Gower's distance ensures a positive semi-definite matrix of similarities between all restaurants. A Matrix where all approximately 8000 restaurants get compared with each other can look like the following:

$$\begin{bmatrix} 0. & 0.17363708 & 0.1551537 & ... & 0.4936557 & 0.4884038 & 0.47464347 \\ 0.17363708 & 0. & 0.16974124 & ... & 0.4718401 & 0.42556286 & 0.48930582 \\ 0.1551537 & 0.16974124 & 0. & ... & 0.48983473 & 0.49355745 & 0.47082245 \\ ... \\ 0.4936557 & 0.4718401 & 0.48983473 & ... & 0. & 0.15529759 & 0.16903748 \\ 0.4884038 & 0.42556286 & 0.49355745 & ... & 0.15529759 & 0. & 0.1727381 \\ 0.47464347 & 0.48930582 & 0.47082245 & ... & 0.16903748 & 0.1727381 & 0. \end{bmatrix}$$

The value 0 in the matrix can be explained by the fact that the restaurant is compared with itself and thus being identical. In this example matrix, all vectors and characters are assigned equal weighting.

5.2 Creation of a Vector

To understand the math of how exactly a numerical vector gets created when provided with both numerical and textual information, it is necessary to understand the mechanics that can be used in such vector generation. Moreover, it's important to recognize that Gower's distance is a concept rather than a fixed formula, and the ways it is implemented can vary. The most complex aspect of understanding the generation of a numerical vector lies in comprehending qualitative data. There are multiple ways to convert qualitative data into

numbers.
One straightforward approach to comparing two qualitative variables, which for example could be 2 Restaurant names like "Subway" and "BurgerKing" is to say if they either match or mismatch. The reason behind comparing restaurant names is that many restaurants with the same cuisine have similar names, such as "Burgerfriends" and "BestBurger" which both offer American cuisine. If the names match the value is set to 0, if they don't then the value is set to 1. In the case of "BurgerKing" and "Subway," the value would be set to 1 since they are distinct words.
Alternatively, the Jaccard Coefficient (Niwattanakul et al., 2013) offers a different way to measure the similarity between these two restaurant names. It is defined as the size of the intersection of the sets divided by the size of their union. The formula for the Jaccard coefficient J(S,B) between "Subway" and "BurgerKing" is as follows:

$$J(S,B) = |\frac{S \cap B}{S \cup B}| \tag{4}$$

as proved by Niwattanakul et al. (2013) [p. 2]

Using this formula and the two restaurants, the vector calculation would be as follows:

{S,u,b,w,a,y} and {B,u,r,g,e,r,K,i,n,g}
Intersection: {u} = 1 (B and b are not the same and are treated as two different characters)
Union: {S,u,b,w,a,y,B,r,g,e,K,i,n} = 13
Here, only the unique letters are taken into account.
Utilizing the Jaccard coefficient formula:

$$J(S,B) = |\frac{1}{13}| = 0.07692 \tag{5}$$

This vector can be interpreted as follows: "Subway" and "BurgerKing" have a Jaccard Coefficient of 0.07692, indicating that 7.692 % of the characters are in common between these two names and 92.307 % are uncommon between these two.
The Dice coefficient is another measure of similarity that can be used for whole language similarities and images as seen in Oco et al. (2013) [p.1-4]. Similar to the Jaccard coefficient, the formula is defined as twice the size of the intersection divided by the total amount of characters in both words and can look as follows:

$$D(S,B) = \frac{2 \cdot |S \cap B|}{|S| + |B|} \tag{6}$$

as proved by Oco et al. (2013) [p. 2]

Using this formula and the same two restaurants as in both examples following calculation will be shown:
{S,u,b,w,a,y} and {B,u,r,g,e,r,K,i,n,g}

Intersection: {u} = 1
Using the Dice coefficient formula:

$$J(S,B) = |\frac{2}{16}| = 0.125 \tag{7}$$

This can be interpreted the same way as with the Jaccard coefficient. For quantitative data, this formula can be applied (McCaffrey, 2020):

$$X = \frac{|diff|}{range} \tag{8}$$

with "diff" being the difference in values for each restaurant and "range" being the highest difference the values can have in the entire list. An example table of four restaurants will be introduced, and a sample calculation will be presented to demonstrate the creation of a vector with the Gower distance.

Restaurant Name	Votes	star rating	offers burger
Burger King	1544	4	Yes
Subway	946	5	No
McDonalds	4965	3	Yes
KFC	1444	3	Yes

Table 1: Example restaurants for vector calculation

With the Formula 3 the Gower distance between "BurgerKing" and "McDonalds" will be calculated. For textual content, the value is set to 1 if it's different and 0 if it's identical. For numeric content, equation 8 is used.

$$(1)\ \ Range\ of\ Votes = 4965 - 946 = 4019$$

$$(2)\ \ Range\ of\ star\ rating = 5 - 3 = 2$$

$$(3)\ \ Name : different = 1$$

$$(4)\ \ Votes : \frac{|1544 - 4965|}{4019} = 0.8512$$

$$(5)\ \ starrating : \frac{|4 - 3|}{2} = 0.5$$

$$(6)\ \ offers\ burger : identical = 0$$

$$(7)\ \ \frac{1 + 0.8512 + 0.5 + 0}{4} = 0.5878$$

In conclusion, the Gower distance between "McDonalds" and "BurgerKing" is 0.5878 meaning they are 58.78 % identical based on the information in the table. Analogously, this process will be applied to all other restaurants and vector calculations until a 4x4 matrix is created, resembling the following:

$$\begin{bmatrix} 0. & 0.6621 & 0.5878 & 0.3812 \\ 0.6221 & 0. & 1 & 0.7809 \\ 0.5878 & 1 & 0. & 0.4690 \\ 0.3812 & 0.7809 & 0.4690 & 0. \end{bmatrix}$$

5.3 Gower Distance for the Recommendation problem

Gower Distance is useful in this context because it achieves to categorize various types of data like cuisine types, costs, ratings, and more. Since Gower distance can work with mixed data types and also qualitative data this makes the Gower distance a great choice for such systems.
In the recommendation system, Gower distance assesses the dissimilarity between restaurants based on their features, after the clustering process. This metric is crucial, as it allows the program to group similar restaurants even further, thereby enhancing the precision of recommendations. This approach contributes to the creation of a more accurate and personalized recommendation system.

6 Eliminate Recommendation Bias via Standard Score

6.1 Introduction to the Recommendation Problem

Another challenge encountered in the recommendation process besides finding a fitting restaurant and grouping the data, is that the recommended restaurant might have a high rating, such as five stars, but with only a limited number of reviews, for example, four. Meaning that those five given stars could be biased and the real value of the restaurant could be far lower. Given the problem that a restaurant with four reviews could be biased, it can be said that restaurants with more ratings have a far more authentic rating than the others. In practical terms, a restaurant with a rating of 3.8 stars based on 1500 reviews holds more significance for the user than a restaurant with a five-star rating but only four reviews implicating that new restaurants that have been opened recently may have the same rating, as a restaurant that has been open for many years. Since the Excel list does not include any dates of the restaurant opening this can't be taken into account.
Additionally, restaurants are engaged in competition for the highest ratings, leading to the unfortunate practice of fake reviews. Given this problem, a restaurant could have 400 out of 500 fake reviews and thus get a fake star rating. The restaurant with only 150 but real human ratings has a lower rating and thus gets less recognition from other guests. Fighting this problem can get too complex and is not taken into account in this work since this requires a deeper understanding of the recommendation functionality.

6.2 Standard Score and its Contents

A standardized score, commonly referred to as a Z score, indicates that it has undergone conversion from its original scale or metric into standard deviation units (Fred Clavel, 2019).

As mentioned earlier, consider a scenario where a restaurant may have fewer stars but a more defined score than another restaurant with five stars but only four votes. In this work, this issue is addressed through the utilization of the standard score. Using this method values that match the recommendations and ratings are found. Using the mean µ and the standard deviation σ a formula assigns values for each line in the Excel sheet is found:

$$\mu = \frac{1}{n} \cdot \sum_{n=0}^{N} \tag{9}$$

$$\sigma = \sqrt{\sum_{i=1}^{n} (x_i - \mu)^2 \cdot p_i} \tag{10}$$

$$Z = \frac{x - \mu}{\sigma} \tag{11}$$

as proved by Curtis et al. (2016) [p. 2]

When referring to Z-transformation, it signifies that through standard score, any distribution is transformed into a distribution with a mean µ equals zero and the standard deviation σ equals 1. The relative position of the values and thus the original distribution form is retained.

The following illustration demonstrates the effect of Z-transformation with two examples. In the left example, the mean is 10, and the standard deviation is 30, while on the right, the mean is 200, and the standard deviation is 20. After Z-transformation, both examples exhibit a mean of 0 and a standard deviation of one.

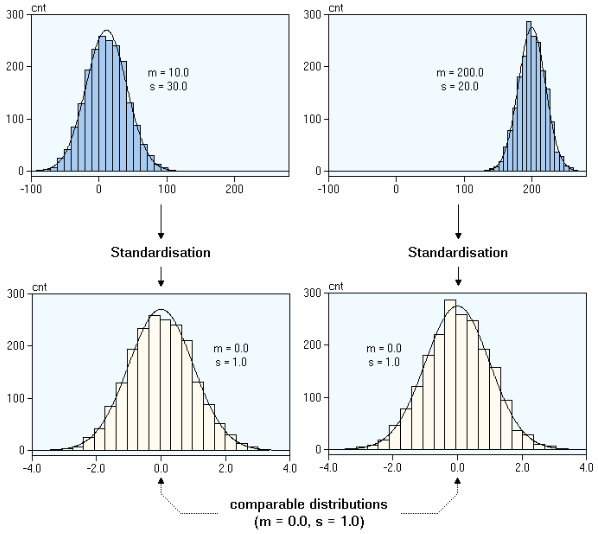


Figure 6: Two examples of standardisation (Statistic, 2023)

6.3 Implementation into the Excel Sheet

In the Excel sheet, the variable x corresponds to the "Votes" column, situated in column "S." This column represents the current number of recommendations the restaurant has received. The next subsection, "Standard Score Example and Evaluation of Restaurants", will provide a clear example for this. Using this method values that match the recommendations and ratings can be achieved. Essentially, the Z-score a restaurant receives is directly proportional to its rating-value ratio – the higher the Z-score, the higher the ratio. An updated formula was used to achieve an accurate standard score based on the restaurant's star rating (s). Without these adjustments, a clear overview of the restaurants would not be possible, as it would ignore the weighting of the stars. So to take the star rating (s) into account it is necessary to multiply the standard score with the current restaurant's star rating (s) to ensure the right implementation of the formula. The addition of two to the formula at the end is designed to guarantee only positive results, simplifying the utilization of the standard score.

$$Z = ((\frac{x-\mu}{\sigma}) + 2) \cdot s \tag{12}$$

With this formula, the recommendation problem of being suggested a restaurant that may not be suitable for the user due to having minimal votes and thus possessing a biased rating is mitigated.

After applying these values to each restaurant in the list, it shows that restaurants with a low rating and few reviews yield a small Z-score close to zero, while restaurants with a good vote and many reviews have a Z-score reaching up to 104,94588 in this particular Excel sheet. This indicates that the higher the Z-score, the lower the chances of being disappointed after visiting that specific restaurant.

6.4 Standard Score Example and Evaluation of Restaurants

For the next step Restaurant "The Chocolate Room" in line 23 in the Excel sheet was chosen to present an example calculation for the standard score that could look like the following:

$$\mu = \frac{1}{7944} \cdot 842689 = 106,0786757 \tag{13}$$

$$\sigma = \sqrt{(4-106,0789757)^2 \cdot \frac{1}{7944} + ... + (1-106,0786757)^2 \cdot \frac{1}{7944}} = 332,8364 \tag{14}$$

$$Z = ((\frac{59-106,0786757}{332,8364}) + 2) \cdot 3 = 5,5623911 \tag{15}$$

with a Z-score mean of 3.967, this score is very good, and the restaurant's authenticity is high.

To find out, which restaurant is the "best", the Z-score plays a significant role. Since now, there are no preferences that are given by the user, the recommendation program must identify the best restaurant based on other available information than the user input. Notably, with a Z-score of 104,94588 the restaurant "Hauz Khas Social" has the biggest authenticity. Furthermore, implying a good rating and many votes. In detail, this restaurant has a "Rating star" of 4 and 7931 individual votes. Visiting this restaurant is likely to provide the user with the most satisfying experience among all the listed restaurants, making it the standout "best" choice.

7 Implementation in Python

7.1 Selection of Python Libraries

To execute the code and use all the features, different Python libraries were used to access different functionalities for the recommendation problem. To implement an Excel file into the Python program "Pandas" was used to read that file and create a data frame inside "Visual Studio Code" (VSC).
To not only get an output in the VSC console the Graphical User Interfaces (GUI) "Tkinter" and more specifically "customtkinter" were used to achieve a user-friendly experience when clicking through the finished program. Tkinter comes from the "TK interface" and allows developers to create specific desktop applications with graphical windows featuring buttons, text, and other elements. This transformation changes the program's functionality, allowing users to interact by clicking buttons and entering numbers or strings, enhancing the recommendation system's user-friendliness. This makes the system accessible to a broader audience, as it is self-explanatory and eliminates the need for detailed instructions.
The import of the "random" library was necessary to get a random index from the Excel list, which is required only once. The "NumPy" library short for Numerical Python supports the use of large multi-dimensional matrices along with mathematical functions to operate these matrices. The use of "Gower" and "sklearn.cluster (K-means)" was explained in the chapter "Gower Distance and its Similarity Measurement" and "K-means Clustering in General". The usage will be shown in the following subsection where the code is analyzed and partially explained.

7.2 Description of Used Code

As the GUI is not a central focus of the seminar work, a detailed explanation of how each window was created and how each button is positioned will not be provided.
Given that the longitude and latitude in the Excel sheet are not in the correct mathematical format, it is needed to rewrite these two values and insert a point after every second digit. This is achieved by a small algorithm shown below:

```
def get_location(restaurant):
    def get _coord(val):
        s = str(int(val))
        return float(s[:2] + "." + s[2:])
     return(get _coord(restaurant["Latitude"]), get _coord(restaurant["Longitude"]))
```

Pictures of the code will be shown at "Examples of Code". Upon running the code, the user gets presented with a graphical window displaying: "Welcome to Kai's Dining Experience in New Delhi". The window features four buttons, with the fourth button, "End Program," terminating the code and closing the window. The first, second, and third buttons each have specified functions that execute upon clicking.
The first and third buttons both have small functions that will either give the user the best restaurant in the city (in this case, the best restaurant in the entire Excel table) or a randomly chosen restaurant using the "random" library returns a random index of the Excel table. How the best restaurant in the Excel list is defined was already discussed in "Standard Score Example and Evaluation of Restaurants". The concept behind having the possibility to get shown a random restaurant is that a user spontaneously can find a restaurant if they have no specific preferences. These two features aim to give users a sense of flexibility, allowing them not only to input their preferences but also to receive a restaurant suggestion from the application.
Upon clicking either of those buttons, the user is presented with a new window that shows the "OpenStreetMap", the name, and pin of the exact location of the specified restaurant. The user can freely navigate the map and can zoom in and out to see where the given restaurant is located in New Delhi and its surroundings.
The main part of the code happens, when clicking on the second button "Give me the best 3 restaurants based on my preferences". A new window opens up, allowing the user to choose their preferences for the perfect restaurant. The user can input details such as the city, locality, cuisine, cost for two people, star rating, delivery availability, table booking availability, and the minimum number of votes a restaurant should have. Since the cost for two people needs to be entered as the amount of rupees, a small text above the box shows the value of the currency in euros. The user can now either fill in all his preferences or leave some of these possibilities "empty" if it's not important. Empty boxes will either not be taken into account or will have a standard value that is already shown. Via a drop-down menu or a box where the user can enter numbers, it is assumed that the user has an easy-to-understand selection process. The code of such an Input box can look like the following:

```
ctk.CTkLabel(choice_window, text= "How much stars should the restaurant have?").pack()
rating _entry = ctk.CTkEntry(choice _window)
rating _entry.insert(0, "0")
rating _entry.pack()
```

After pressing "Enter" the input is saved and a new window with the three best-fitting restaurants based on the dream restaurant" opens up, where the user can now click on each restaurant name and is again presented with the "OpenStreetMap" window, where he can see the location and name of the restaurant and click on the Pin to get further information about the restaurant. If the user is satisfied he can press the "close" button in the window and get back to the main menu, where he can either end the program or create a

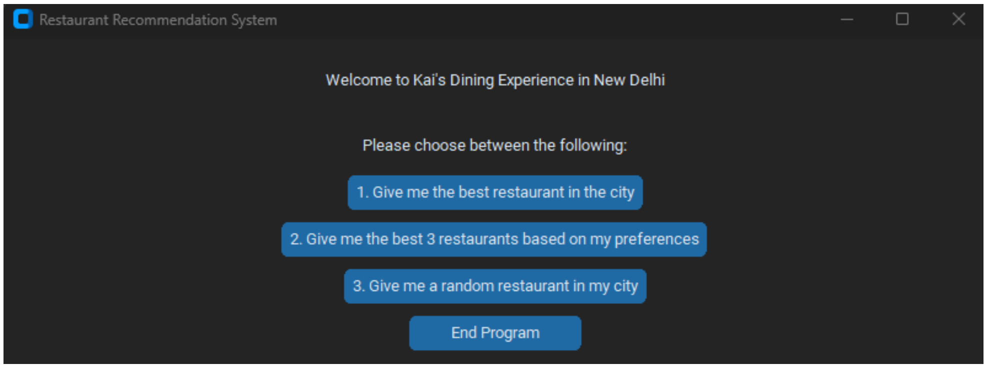
new restaurant with improved preferences.
While not directly shown to the user but visible in the console, after entering their preferences and pressing enter, additional information about the methodological part is displayed. The "new" dream restaurant gets put into one of the two already existing clusters that were created and also the Gower distance between this restaurant and all other restaurants in the cluster gets calculated and printed. Given this Gower distance 1 x n matrix, the program then identifies the lowest distance and prints the distance and the corresponding restaurant.

7.3 Evolution of Code

The complete Code has been written with the help of various sources, including "Guba (2021), Bowne-Anderson (2022), eduardoftdo (2021), TomSchimansky (2021), Weikert (2021), Code (2023), wwwjk366 (2022), Wu (2021), Ahuja, Solanki, and Nayyar (2019), acw1668 (2022), NumPy (2023), Pedregosa et al. (2011), and Burgaud (n.d.)". The code has multiple versions, each version getting more and more complex. The first code called "main" was a simple implementation that would only print the text in the console and apply an easy filter on the Excel data frame which doesn't give the algorithm a lot of space to suggest the perfect restaurant since the user could apply such a special filter that no restaurant could be found.
The second version introduced a Tkinter GUI with an enhanced filtering algorithm. This version provided users with a more user-friendly interface, allowing them to click on buttons and input values into boxes.
The third version, named "Final GUI recommend 2.0," featured an improved GUI and allowed users to visualize the desired restaurant on the "OpenStreetMap". The final version used in this paper is called "Final GUI recommend 3.0." This version can calculate the Gower distance between multiple restaurants in a K-means cluster and present the results in a visually appealing GUI. Each version reflects a deeper understanding of the problem and the use of more advanced code. All code versions are available in this GitHub folder (Hinrichs, 2023).

7.4 Examples of Code

Consider a scenario where an individual is keen on dining in New Delhi, open to exploring a new restaurant. A second person has a specific craving for North Indian cuisine in New Delhi but is constrained by a budget of 700 Rupees for two. This individual aims to reserve a table at a 4-star restaurant with at least 30 votes. The third person is inclined to dine at a new North Indian restaurant, irrespective of ratings, with a budget of 1000 rupees.
All 3 users are presented with the following Menu:

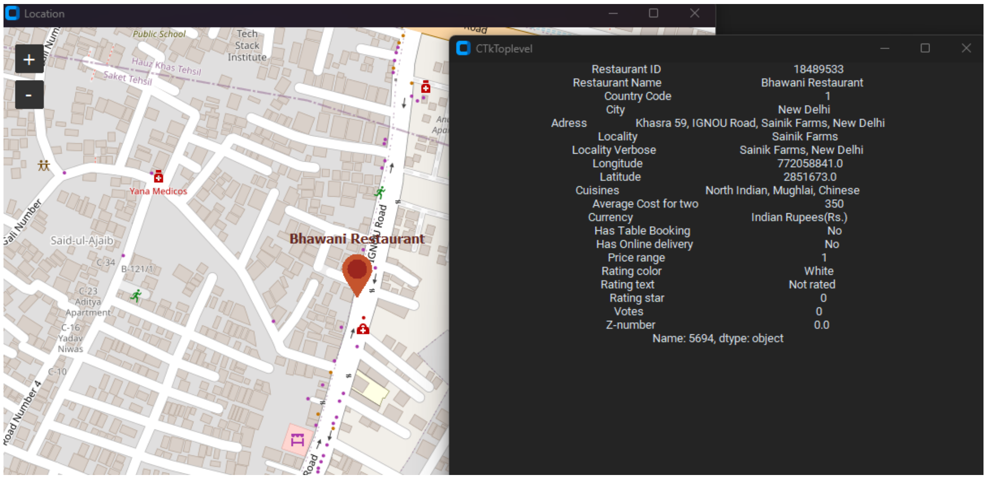

Restaurant Recommendation System
1. Give me the best restaurant in the city
2. Give me the best 3 restaurants based on my preferences
3. Give me a random restaurant in my city
End Program

Welcome to Kai's Dining Experience in New Delhi

Please choose between the following:

Figure 7: Main menu

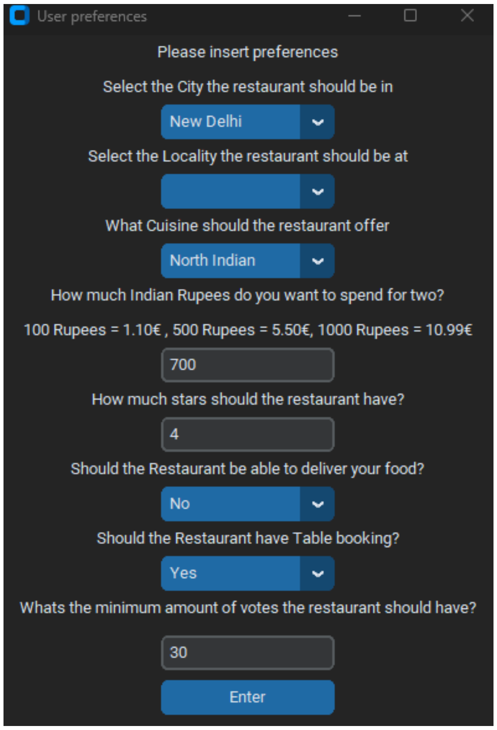
The first User is then presented a random restaurant from the Excel list.



```
Restaurant ID                          18489533
Restaurant Name               Bhawani Restaurant
Country Code                                  1
City                                  New Delhi
Adress        Khasra 59, IGNOU Road, Sainik Farms, New Delhi
Locality                           Sainik Farms
Locality Verbose        Sainik Farms, New Delhi
Longitude                           772058841.0
Latitude                              2851673.0
Cuisines          North Indian, Mughlai, Chinese
Average Cost for two                        350
Currency                   Indian Rupees(Rs.)
Has Table Booking                            No
Has Online delivery                          No
Price range                                   1
Rating color                              White
Rating text                           Not rated
Rating star                                   0
Votes                                         0
Z-number                                    0.0
Name: 5694, dtype: object
```

Figure 8: A random Restaurant

The User can now see the Z-number of the restaurant, indicating the authenticity of the current voting. Since this is a new restaurant with 0 votes the Z-number is 0.0. The choice to randomly select a restaurant also comes with trying out something new, so getting a not-rated restaurant is reasonable. For the second and third user, the preference Menu figure 9 allows both to input their preferences. After entering their preferences, the menu and output for the second user appear as follows:



Please insert preferences

Select the City the restaurant should be in

New Delhi

Select the Locality the restaurant should be at

What Cuisine should the restaurant offer

North Indian

How much Indian Rupees do you want to spend for two?

100 Rupees = 1.10€ , 500 Rupees = 5.50€, 1000 Rupees = 10.99€

700

How much stars should the restaurant have?

4

Should the Restaurant be able to deliver your food?

No

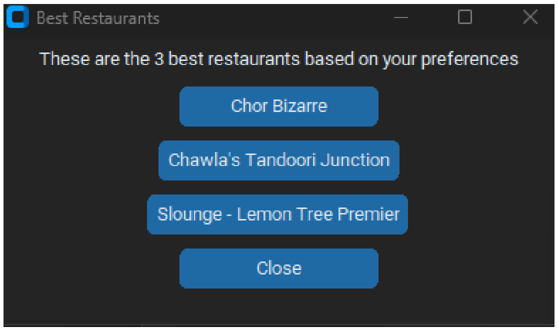
Should the Restaurant have Table booking?

Yes

Whats the minimum amount of votes the restaurant should have?

30

Figure 9: Preference Menu



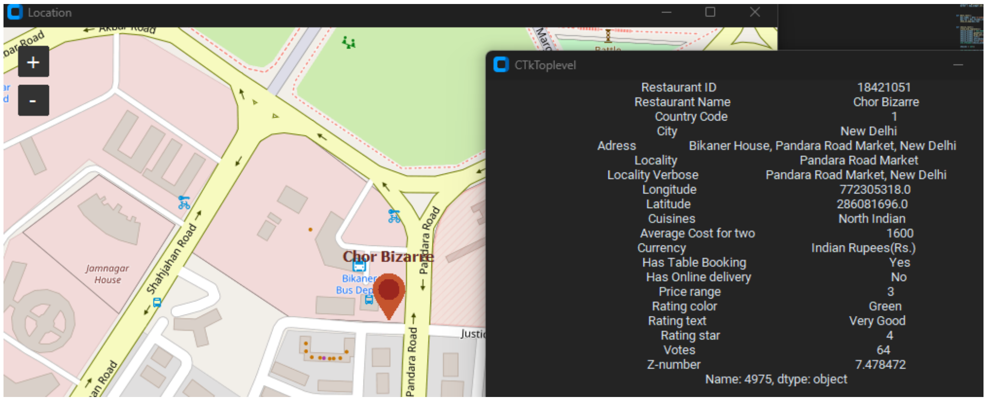
These are the 3 best restaurants based on your preferences

Chor Bizarre

Chawla's Tandoori Junction

Slounge - Lemon Tree Premier

Figure 10: Suggestion of three restaurants



```
        Restaurant ID                                          18421051
      Restaurant Name                                      Chor Bizarre
         Country Code                                                 1
                 City                                         New Delhi
Adress             Bikaner House, Pandara Road Market, New Delhi
             Locality                               Pandara Road Market
     Locality Verbose                    Pandara Road Market, New Delhi
            Longitude                                       772305318.0
             Latitude                                       286081696.0
             Cuisines                                      North Indian
Average Cost for two                                               1600
             Currency                                Indian Rupees(Rs.)
    Has Table Booking                                               Yes
  Has Online delivery                                                No
          Price range                                                 3
         Rating color                                             Green
          Rating text                                         Very Good
          Rating star                                                 4
                Votes                                                64
             Z-number                                          7.478472
Name: 4975, dtype: object
```

Figure 11: Best fitting restaurant

The second user is presented with a choice between three fitting restaurants for control, as depicted in section 11. The first restaurant recommended aligns the most with the essential features the user set for their "dream" restaurant. "Chor Bizarre" in New Delhi offers North Indian cuisine with table booking at an average cost of 1600 Rupees and has garnered 64 votes, resulting in a 4-star rating. As observed, the Z-number is notably high at 7.478, surpassing the mean Z-number in the entire list, which is 3.967. Although the contents of this restaurant do not exactly match those of the desired restaurant, it represents the closest fit. The Gower distance is at its minimum, and the high Z-number suggests that this restaurant should leave the user most satisfied after the visit. The same principle applies to the third user, who, in this case, inputs different preferences than the second user and consequently receives different results. The third user now has the choice between three restaurant's named "Rama Desi Ghee Meat Wala", "Best Chicken Corner" and "Bansal Food". All three of these restaurants have a Z-number of 0.0 and thus are not rated and new restaurants. By clicking on the pin the user can have mutual insights into the restaurant data and choose one of those three suggestions and get the exact address of the restaurant.

8 Conclusion

In conclusion, this study delved into the development of an effective restaurant recommendation system, addressing key challenges such as recommendation bias. The integration of advanced methodologies, including Gower distance, K-means clustering, and standard score, has resulted in the creation of a user-friendly application capable of offering personalized restaurant suggestions.
K-means clustering helped enhance the precision of recommendations by categorizing these restaurants based on certain given characteristics. Afterwards, the Gower distance calculated the dissimilarity between restaurants, enabling the clustered data to be put in matrices.
Standard score played a pivotal role in eliminating recommendation bias by transforming restaurant data into a standardized distribution. This approach allowed for a fair evaluation of restaurants, considering both their ratings and the number of reviews.
The implementation via Python offered a graphical interface for users to interact seamlessly and a step-by-step process without the need for instructions. To sum up, this study has not only addressed the existing recommendation bias but has also provided the user with machine learning algorithms for a personalized dining experience to reduce disappointment after a restaurant visit.

ChatGPT was used for parts of the code, which included solving problems, implementing 2 codes into one and creating small algorithms. For the Seminar text, ChatGPT helped finding fitting words and adjectives without adding new information that wasn't written by the author.

References

acw1668 (Nov. 5, 2022). *Custom Tkinter OptionMenu not shown*. URL: https://stackoverflow.com/questions/77425909/custom-tkinter-optionmenu-not-shown/77428211#77428211.

Ahmed, Mohiuddin, Raihan Seraj, and Syed Mohammed Shamsul Islam (2020). "The k-means Algorithm: A Comprehensive Survey and Performance Evaluation". In: *Electronics* 9.8. ISSN: 2079-9292. DOI: 10.3390/electronics9081295. URL: https://www.mdpi.com/2079-9292/9/8/1295.

Ahuja, Rishabh, Arun Solanki, and Anand Nayyar (2019). "Movie recommender system using k-means clustering and k-nearest neighbor". In: *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, pp. 263–268.

Bektas, Alperen and René Schumann (2019). "How to optimize gower distance weights for the k-medoids clustering algorithm to obtain mobility profiles of the swiss population". In: *2019 6th Swiss Conference on Data Science (SDS)*. IEEE, pp. 51–56.

Bouhmala, Noureddine (2016). "How good is the euclidean distance metric for the clustering problem". In: *2016 5th IIAI international congress on advanced applied informatics (IIAI-AAI)*. IEEE, pp. 312–315.

Bowne-Anderson, Hugo (Sept. 1, 2022). *Python Select Columns Tutorial*. URL: https://www.datacamp.com/tutorial/python-select-columns (visited on 10/26/2023).

Burgaud, Andre (n.d.). *How to Use Python Lambda Functions*. URL: https://realpython.com/python-lambda/.

Code, Clear (Mar. 23, 2023). *The ultimate introduction to modern GUIs in Python [with tkinter]*. URL: https://www.youtube.com/watch?v=mop6g-c5HEY&t=36135s&ab_channel=ClearCode.

Cui, Mengyao et al. (2020). "Introduction to the k-means clustering algorithm based on the elbow method". In: *Accounting, Auditing and Finance* 1.1, pp. 5–8.

Curtis, Alexander E, Tanya A Smith, Bulat A Ziganshin, and John A Elefteriades (2016). "The mystery of the Z-score". In: *Aorta* 4.04, pp. 124–130.

decypher (Mar. 16, 2018). *Machine Learning: What it is and Why it Matters*. URL: https://www.decypher.com/machine-learning-matters/ (visited on 12/02/2023).

eduardoftdo (Mar. 23, 2021). *Python - How to filter out data in Excel*. URL: https://stackoverflow.com/questions/66770678/python-how-to-filter-out-data-in-excel (visited on 10/28/2023).

Fred Clavel, Ph. D. (Mar. 18, 2019). *Basics: Standardization and the Z score*. URL: https://fredclavel.org/2019/03/18/basics-standardization-and-the-z-score/ (visited on 11/29/2023).

Gower, J.C. (1971). "A General Coefficient of Similarity and Some of Its Properties". In: *Biometrics* 27.4, pp. 857–871.

Guba, Zoltan (Oct. 27, 2021). *How to Read Excel Files Using Pandas*. URL: https://blog.devgenius.io/reading-excel-files-with-pandas-the-basics-6a6be9cc8763 (visited on 10/26/2023).

Hinrichs, Kai (Nov. 28, 2023). *Python Codes and Excel tables*. URL: https://github.com/Kailong66522/HappyDining (visited on 12/03/2023).

Kodinariya, Trupti M, Prashant R Makwana, et al. (2013). "Review on determining number of Cluster in K-Means Clustering". In: *International Journal* 1.6.

L'Heureux, Alexandra, Katarina Grolinger, Hany F. Elyamany, and Miriam A. M. Capretz (2017a). "Machine Learning With Big Data: Challenges and Approaches". In: *IEEE Access* 5, pp. 7776–7797. DOI: 10.1109/ACCESS.2017.2696365.

— (2017b). "Machine Learning With Big Data: Challenges and Approaches". In: *IEEE Access* 5, pp. 7776–7797. DOI: 10.1109/ACCESS.2017.2696365.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *nature* 521.7553, pp. 436–444.

Lee, Cecilia S and Aaron Y Lee (2020). "Clinical applications of continual learning machine learning". In: *The Lancet Digital Health* 2.6, e279–e281.

Li, Yuxi (2018). *Deep Reinforcement Learning: An Overview*. arXiv: 1701.07274 [cs.LG].

McCaffrey, James D. (Apr. 21, 2020). *Example of Calculating the Gower Distance*. URL: https://jamesmccaffrey.wordpress.com/2020/04/21/example-of-calculating-the-gower-distance/ (visited on 11/28/2023).

Mehta, Shruti (Mar. 13, 2018). *Zomato Restaurants Data*. URL: https://www.kaggle.com/datasets/shrutimehta/zomato-restaurants-data?select=file4.json (visited on 11/28/2023).

Niwattanakul, Suphakit, Jatsada Singthongchai, Ekkachai Naenudorn, and Supachanun Wanapu (2013). "Using of Jaccard coefficient for keywords similarity". In: *Proceedings of the international multiconference of engineers and computer scientists*. Vol. 1. 6.

NumPy (Sept. 16, 2023). *NumPy: the absolute basics for beginners*. URL: https://numpy.org/doc/stable/user/absolute_beginners.html.

Oco, Nathaniel, Leif Romeritch Syliongka, Rachel Edita Roxas, and Joel Ilao (2013). "Dice's coefficient on trigram profiles as metric for language similarity". In: *2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*, pp. 1–4. DOI: 10.1109/ICSDA.2013.6709892.

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

Piech, Chris (Sept. 1, 2012). *K Means*. URL: https://stanford.edu/~cpiech/cs221/handouts/kmeans.html (visited on 11/29/2023).

Portugal, Ivens, Paulo Alencar, and Donald Cowan (2018). "The use of machine learning algorithms in recommender systems: A systematic review". In: *Expert Systems with Applications* 97, pp. 205–227.

Statistic, Fundamendals of (2023). *z-Transform*. URL: http://www.statistics4u.info/fundstat_eng/ee_ztransform.html (visited on 11/29/2023).

TomSchimansky (Dec. 16, 2021). *TkinterMapView - simple Tkinter map component*. URL: https://stackoverflow.com/questions/66770678/python-how-to-filter-out-data-in-excel (visited on 11/02/2023).

Usama, Muhammad, Junaid Qadir, Aunn Raza, Hunain Arif, Kok-Lim Alvin Yau, Yehia Elkhatib, Amir Hussain, and Ala Al-Fuqaha (2019). "Unsupervised machine learning for networking: Techniques, applications and research challenges". In: *IEEE access* 7, pp. 65579–65615.

Weikert, Daniel (Sept. 30, 2021). *Python für Anfänger Excel bearbeiten und visualisieren*. URL: https://www.youtube.com/watch?v=jaaob6cbBks&ab_channel=DanielWeikert (visited on 10/21/2023).

Wu, BoKai (2021). "K-means clustering algorithm and Python implementation". In: *2021 IEEE International Conference on Computer Science, Artificial Intelligence and Electronic Engineering (CSAIEE)*. IEEE, pp. 55–59.

wwwjk366 (Nov. 13, 2022). *gower 0.1.2*. URL: https://pypi.org/project/gower/#:~:text=Introduction,and%20some%20of%20its%20properties (visited on 11/22/2023).

Hiermit versichere ich, die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie die Zitate deutlich kenntlich gemacht zu haben.
Ich erkläre weiterhin, dass die vorliegende Arbeit in gleicher oder ähnlicher Form noch nicht im Rahmen eines anderen Prüfungsverfahrens eingereicht wurde.

Würzburg, den 8.12.2023

Kai Hinrichs

Hinrichs