NEIGHBORHOOD SCORE CALCULATION

SECKIN ADALI

Contents

Introduction

This note outlines the methods for assigning neighborhood scores in the project 'The Neighborhood Vibe Score'.

The project introduces two methods to calculate the neighborhood score of a given address. In both methods, a weighted sum on four types of data is used to calculate the score.

- The first method enables users to input their preferences which are then used to determine weights, and return a score tailored to the user's interests.
- The second method automates the weighting process, using k-means clustering on a dataset of addresses to determine representative weights for each.

For detailed code with explanations and examples, please refer to the `notebooks/assign_scores.ipynb` Jupyter notebook. The main code script can be found at `src/assign_scores.py`.

Method 1: Personal Interests Score Based on User Input

The input consists of:

- Facility count weights w_i for each facility type $i = 1, \ldots 8$.
- Commute-to-work score, an ordinal value from 1 to 4 if a work address is provided (optional).

The facility count weights w_i reflect the user's preferences on different types of facilities. In the current prototype, there are 8 facility types, but this could be expanded in future versions. The commute-to-work score incorporates the user's commuting preferences if they provide a work address, converting this information into an ordinal value that is integrated into the overall score.

The score for each address is calculated using four types of data, each with its own weight coefficient. These coefficients determine the relative importance of each data type in the final score:

- c_1: Facility counts (per facility type)
- c_2: Weighted average of ratings (per facility type)
- c_3: Travel time to the closest facility (per facility type)
- c_4: Commute-to-work score (set to 0 if no work address is provided)

While the weight coefficients can be customized by the user in a future version, for simplicity, I'll use the following default values, prioritizing facility counts and commute-to-work preferences (if provided):

Weight Coefficients:

$$\begin{cases} \text{If no work address:} & c_1 = 0.8, \quad c_2 = 0.1, \quad c_3 = 0.1, \quad c_4 = 0 \\ \text{If work address provided:} & c_1 = 0.4, \quad c_2 = 0.1, \quad c_3 = 0.1, \quad c_4 = 0.4 \end{cases}$$

Let f_i be the facility count for each facility type ($i = 1, ..., 8$). *The weighted average of ratings* per facility type is calculated as follows:

$$r_i = \frac{\sum_j (a_{ij} \cdot n_{ij})}{\sum_j n_{ij}}$$

where a_{ij} is the rating of the search result j (of facility type i), and n_{ij} is the number of ratings.

This weighted average leverages the additional information from facilities with more ratings, though it may cause facilities with very high ratings to disproportionally influence the neighborhood score.

Once the data (facility counts, average ratings, and travel times) is available, I'll apply log-scaling followed by min-max scaling for normalization. Log-scaling reduces the impact of outliers with high facility counts, while min-max scaling ensures all values fall between 0 and 1:

$$N(x) = f_s(\log(1 + x))$$

where f_s represents min-max scaling.

If a commute-to-work score is provided, it will be normalized separately using min-max scaling, with a range of 1 to 4.

Now, let

- f_i be the normalized facility counts
- r_i be the normalized weighted average ratings
- t_i be the normalized travel times to closest facilities
- o be the normalized commute-to-work-score

The neighborhood score is calculated using the following weighted sum:

$$s = c_1 \sum_i w_i f_i + c_2 \sum_i w_i r_i + c_3 \sum_i w_i (1 - t_i) + c_4 o$$

Note that I'm using $(1 - t_i)$ to give higher scores to smaller travel times.

Remark: It is required that the input satisfies $\sum_i w_i = 1$ and $\sum_k c_k = 1$.

Method 2: Neighborhood Vibe Score Based on Clustering

The second method automates the scoring process by using the data on the pool of addresses only. Addresses are clustered using data on facility counts, average ratings, travel times, and neighborhood populations. Then each address is scored according to its cluster.

Here are the steps of the score calculation:

(1) Using all available data apply elbow rule on k-means clustering to determine the optimal cluster number k.

(2) Divide addresses into clusters using k-means with this k.

(3) On each cluster, calculate weights using

$$\frac{\text{median}}{1 + \text{stdev}}$$

where median and standard deviation are calculated for facility counts for each facility type. The resulting values are then adjusted to have a sum of 1, providing a cluster-specific weight set.

(4) For each address, calculate a cluster-specific score using the determined weights with Method 1.

The motivation here is to give higher weights to facilities that are more abundant in the neighborhood (higher median), and more consistent within the cluster (lower standard deviation). This helps to prioritize the facility types that are most representative of the cluster's characteristics.

This method ensures an automated and balanced comparison between neighborhoods by scoring each one according to the characteristics of its cluster. For example, an urban location like downtown Zurich naturally scores higher in terms of facility counts compared to a rural area like Allenwinden. However, when using Method 2 to calculate the Neighborhood Vibe Score, Allenwinden is clustered with other rural areas and scored based on facilities that are more common in such environments, like kindergartens rather than bars. *This approach prevents the scoring system from blindly favoring urban areas and gives rural and suburban locations the opportunity to showcase what they uniquely offer.*

Additionally, comparing median differences in facility counts across clusters provides a straightforward way to assess feature importance within the clustering model. In our prototype, this method effectively separates rural and urban areas, with features like public transportation and schools emerging as the most significant factors. Expanding the dataset with more addresses and facility types would likely improve the accuracy and relevance of the results. For a detailed comparison of clusters in our prototype, please refer to `notebooks/assign_scores.ipynb`.