

1 Introduction - Motivation

According to Census data, relocation in America is at its lowest rate on record (Tavernise, 2019). This is concerning because the lack of people moving for work could negatively impact the economy (Kopf, 2018). Still, trying to find the right place to live can be difficult.

Today, a few popular websites help people find the “best places to live”. Livability and Teleport provide a profile of cities and a breakdown of their basic statistics. Niche and AreaVibes show neighborhoods and their “livability” scores, which are calculated based on several categories (e.g., cost, crime, etc.). These individual groupings also receive a rating, and all scores contribute to a locale’s ranking in comparison to nearby areas. Finally, the data presentations are static and arranged in lists.

There are several limits to the current practice. First, none of the aforementioned sites offer a list of neighborhoods similar to the one a user input. Next, while searches are filterable, there are no personalized recommendations for a user. Moreover, these sites do not provide information about local grocery stores or other essential business in an area, which is important when deciding on a new residence. Lastly, aside from home sale maps, none of the sites offer an interactive visualization.

For our project, we developed an interactive application that makes the process of relocation easier. Our goal was to reduce the time and stress of searching for a new location to live by providing people with useful and personalized information on neighborhoods from across the country. We are confident that our tool will encourage more mobility because it will give people the confidence to potentially relocate when they might have been hesitant to before.

2 Problem Definition

While searching for a neighborhood, individuals may not know what factors are important to them. So, our team created a tool that helps people find the right place to live. We believed the best approach was to provide users with valuable information on new areas based on ones they already know and love. The application utilizes a user’s inputted address, determines its neighborhood, and presents a list of suggested locales from across the US that are like the user’s identified neighborhood. The final output is an interactive visualization that presents the recommended neighborhoods and displays the top 3 locations.

3 Literature Survey

First, the GARM chapter provided important details on Census tracts (U.S. Census Bureau, 1994). They are often used as proxies for neighborhoods, which our group exploited. Next, several papers looked at neighborhood factors people find important, which our group considered. Okulicz-Kozaryn (2011) compared livability and satisfaction of locations and finds only a .36 correlation. Though, there may have been a bias towards subjective metrics. Our team treated all selected attributes equally. Ayub et al. (2020) examined housing indicators to better understand housing markets, but the focus was strictly on urban markets. Our dataset contained Census tracts from every type of locale. He and Xia (2020) determined rental effects for a housing market are largely based on amenities. Yet, their analysis only highlighted Barcelona, while we focused on the entire US. Azadeh and Moghaddam (2012) showed economic and government factors can predict fluctuations in housing markets, but they did not consider other features. We incorporated as many neighborhood attributes as possible. Ashkexari-Toussi et al. (2019) assessed sentiment in geotagged photos, but they used old data. We retrieved the most recent data available.

A key component of our project was the implementation of a clustering model to group similar neighborhoods. Fahad et al. (2014) evaluated well-known single clustering algorithms. Boongoen and Iam-On (2018) surveyed cluster ensembles, which combine different clustering outputs into one consensus result. We evaluated several clustering methods to find the one that produced the optimal results. Finally, Halkidi et al. (2001), Liu et al. (2010), Rendón et al. (2011), and Arbelatriz et al. (2013) all reviewed several internal clustering measures, and they investigated and compared the validation properties of the indexes. Internal validity measures were utilized for our evaluations and experiments.

In addition, ranking and recommendation models were reviewed as other possible approaches. Kiseleva et al. (2015) showed that a Naïve Bayes ranking on features for location recommendations performed better than other rankings, but historical data was not leveraged. Our tool allows users to input the locations they have lived. Liu and Yang (2015) improved VSRank recommendation by focusing on

negative similarity and people's preferences. Unfortunately, geography was not considered, but this was one of our key factors. Kim et al. (2020) improved collaborative filtering-based recommendations with sentiment analysis. Our tool relies on direct user interaction instead of past user sentiment. Su et al. (2020) created a vector of multiple similarities, but only MAE and RMSE were used to measure accuracy. Internal metrics related to clustering were used in our evaluations. Bobadilla et al. (2013) examined several recommendation algorithms, but none of them were effective with sparse data or cold starts. To overcome this problem, we reduced the sparsity in our final dataset. Renjith et al. (2020) studied personalized recommender systems on multiple criteria. Yet, they focused exclusively on travel locations, while we collected more granular neighborhood data. Overall, we employed a ranking of the final cluster results, but a recommendation algorithm could not be adequately deployed within the required timeframe.

Moreover, Xiaoyu et al. (2017), Al-Ghossein et al. (2018), and Wasid and Ali (2018) all employed a two-stage architecture. The first paper used matching and ranking on geotagged photos to recommend attractions, yet there was a large assumption about people's photo habits. For our project, we did not make any assumptions about human behavior. The second paper recommended hotels using clustering and ranking. Though, there may have been issues with the contextual data. Our dataset consists exclusively of numerical columns. The third paper used a two-stage approach to cluster then produce recommendations for movies. Unfortunately, the distance measure they employed was used specifically for their dataset, while we have tried to produce a more generalized model. Based on these three papers, we decided to employ a two-stage approach of clustering and ranking for our final visualization tool.

Finally, we looked at how various models were applied to geographical problems, which was the basis of our project. Shashank and Schuurman (2019) showed how variable selection impacts walkability across different models, but the variables could have been combined to create a composite model that was more informative. With our model, all significant factors were considered simultaneously. Nicholson et al. (2019) used spatial regressions on Census blocks to estimate disaster vulnerability. The analysis demonstrated a feasible use case of neighborhood data, but platform selection may have produced bias. Our model included data from all Census tracts to prevent bias. Sadler et al. (2019) scored areas and weighted expert opinions to improve a "healthfulness index". With our project, the data pertaining to neighborhood amenities were weighted based on their distance from Census tract centers.

4 Proposed Method - Intuition and Description of Approaches

Our final tool exhibits several innovations that make it better than "state of the art". To start, we enhanced the underlying dataset. Metrics that are typically used by the existing websites (e.g., income, demographics, etc.) have been combined with data on essential businesses, like grocery stores and healthcare centers. So, users will have an awareness of the different amenities accessible in an area.

Next, our visualization tool uses the results of a clustering model that was applied to Census tracts from all 50 states. The clustering algorithm grouped similar tracts together based on the features in the improved dataset. So, when a user inputs an address into the application, the Census tract, or "neighborhood", of the address will be identified, and similar "neighborhoods" from across the US will be displayed for the user to review. This group of similar "neighborhoods" represents the initial step in generating personalized recommendations, which is not an available option on the existing websites.

Moreover, we enriched the user experience with our visualization tool, "RELO". A major improvement is that the tool is exclusively focused on finding the best place for someone to live based on their input, without limiting to a certain region of the US. RELO displays all potential relocation areas, and users can further examine the top 3 identified results by viewing local amenities and comparing key metrics. Furthermore, the interface improves on competitors by making the style more welcoming and reducing the amount of excess information. Finally, RELO provides an innovative and dynamic map.

4.1 Data Collection and Cleaning

The Census tract data comes from the Census 5-year American Community Survey Responses for 2018, which was found through tables available on data.census.gov. In total, eight Census tables were identified that contained data relevant to our project. After combining the data, there were around 74,000 Census tracts and 1,300 fields. The data was transformed to the true percentage of each population group. Columns were dropped due to sparsity, high correlation with a more applicable column, or being too

specific of a category. To prevent skewed results, extreme outliers were either truncated or the columns were dropped entirely. Null values were imputed, but rows with majority null values were dropped.

A Python script pulled the data on neighborhood amenities by looping through US zip codes and calling the Foursquare API, which has a high free data limit. The data collected for specific amenities was between 50,000-75,000 rows per amenity with several features. For each amenity, the data was combined into a CSV file, and OpenRefine was used for cleaning. Duplicates and venues that did not match the amenity type were dropped, places with similar names were merged, and incorrect tags were modified. Our team obtained data on grocery stores, healthcare centers, gyms, hardware stores, and parks.

4.3 Data Integration

The amenities data was aggregated and integrated with the Census data. For each Census tract, the number of venues of a specific amenity type were counted within 2, 5, 10, 25, and 50 miles of the center and placed into distinct columns. The location counts in these columns were weighted by the inverse of the search radius. To calculate the distance, the following algorithm was applied: 1) project the coordinate system onto a 2D plane; 2) build a KDTree of all the amenity's points; 3) iterate through Census tracts and use a tract's center as a query to the KDTree, along with the distance limit.

4.4 Analysis

The final dataset had 73,056 rows and 30 numeric features. To improve the efficacy of our algorithms and give equal weight to all features, non-normal features were transformed by applying Box-Cox, and then the data was standardized. The standardization of the data also helped with dimensionality reduction.

Dimensionality reduction helped address the “curse of dimensionality” by further reducing the number of features. t-Distributed Stochastic Neighbor Embedding (t-SNE) was the primary technique used because it is particularly well suited for the visualization of high-dimensional datasets. t-SNE is a probabilistic technique that minimizes the divergence between two distributions. As recommended, Principal Component Analysis (PCA) was implemented before t-SNE to reduce the number of dimensions to a reasonable amount (Maaten and Hinton, 2008). PCA transformed the input data to have 18 components, which accounted for 95% of the variability. A t-SNE decomposition was then applied on the newly transformed data. We used two-dimensional embeddings as features to better visualize the results and help speed up the fitting of the clustering algorithms.

The optimal clustering method was a KMeans model with 50 clusters. The discussion of which algorithms were tested, how their performances were evaluated, and how the final model was selected can be found in section “5 Experiments / Evaluation”. The labels produced by the optimal method were merged with the original preprocessed data to obtain a set of overall cluster assignments for all Census tracts. So, Census tracts in the same group are more similar to each another than tracts from other groups.

4.5 Visualization

Our visualization application has been created with Python Flask, and full interaction with the tool occurs in a web browser. Users begin by entering an address of a location they like, and the application identifies and captures the Census tract with the closest center in terms of distance. We originally planned to utilize the Census Bureau's “Geocoding” API, but it was unreliable because it often delivered errors.

Once the Census tract is retrieved, the final clustering results are filtered based on the cluster assignment of the identified tract. The visualization tool displays a map of similar Census tracts, or “neighborhoods”, in the same group as the identified tract, which users can consider as possible relocation options. In addition, the returned Census tracts are ranked according to their Euclidean distance in the feature space from the identified tract. The top 3 ranked locales are displayed, and the user may select any of these neighborhoods. Once one is selected, this will zoom in the country map, identify some of the amenities offered in the area, and display a table that compares the inputted and selected locations based on several features. Finally, if users search for a Census tract that was dropped from the dataset, then the closest Census tract to the inputted location is used as a substitute for the results.

5 Experiments / Evaluation

All our evaluations were performed on a Ryzen 1700 8 core/16 thread processor (base 3.0 GHz, boost 3.7 GHz) with 64 GB of 3000 Mhz RAM. We created a pipeline in a Jupyter notebook to evaluate several clustering algorithms. Since true class labels were not available for comparison, we evaluated the results

of different cluster models by relying on intrinsic criteria, which focus on the quality of the partitions. These metrics evaluate the “goodness” of a clustering result by considering the two properties expected in quality clusters: compact and well separated groups (Han et al., 2012). Over 30 intrinsic criteria can be found in the literature, and Arbelatriz et al. (2013) exhaustively tested many of them. Some measures had significantly better performance than others, but the Silhouette, Calinski-Harabasz, and Davies-Bouldin indexes performed well in a wide range of situations. Thus, we primarily used these three metrics during our evaluations of various clustering methods.

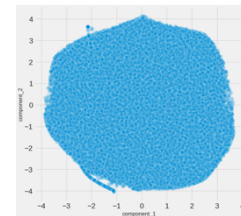
It is important to note what each of the three indexes measure, and how they should be interpreted. The Silhouette coefficient measures the maximum class spread or variance, with the higher number being better. The Calinski-Harabasz index measures the interclass-intraclass distance ratio, with the higher number being better. The Davies-Bouldin criteria measures the maximum interclass-intraclass distance ratio, with the lower number being better. Finally, we developed a list of questions that our evaluations were designed to answer. The details of our experiments and their results are described below.

5.1 Does a non-random structure exist in the dataset?

Clustering results are only meaningful when there is non-random structure in the data. So, we needed to assess the feasibility of clustering analysis on our cleaned dataset, prior to standardization and dimensionality reduction. We conducted a statistical test that employed the Hopkins statistic, which measures the probability that a given data set is generated by a uniform distribution (Lawson and Jurs, 1990). Our dataset being uniformly distributed was the null hypothesis, and the alternative hypothesis was that our dataset was not uniformly distributed and contained clusters (Han et al., 2012). A Python method calculated the Hopkins Statistic on our full dataset and outputted the measure. Using 0.5 as a threshold, we obtained a result of 0.008, so we were able to reject the null hypothesis in favor of the alternative.

5.2 What clustering models should be evaluated?

There are four main categories of clustering approaches: partitioning, hierarchical, density-based, and grid-based. Each technique has its own strengths and weaknesses, and we needed to select the appropriate algorithms for our dataset. After producing the two-dimensional embedding of our dataset with t-SNE, we created a visualization of our data in 2D space, shown on the right. The visualization showcases that our data produced a flat geometry with no distinct separation. To quickly test density-based and grid-based approaches, we tried to generate clusters using the DBSCAN and OPTICS algorithms with default parameters. Both models simply returned the single structure in the visualization as one cluster. Thus, density-based and grid-based approaches were disregarded.



Also, we were unable to fulfill our original intention of evaluating at least one cluster ensemble technique. We performed several tests on the “SimpleConsensusClustering” method from the “kernlglearn” Python library. Unfortunately, due to the large number of samples in our data, the method took up a majority of the computing resources, and it often failed to run. In addition, “OpenEnsembles” was another Python library that we reviewed, but it was obvious that many of the package’s methods had inherent bugs. When we tried to recreate the examples provided in the documentation, we received multiple errors. Hence, the plan of evaluating cluster ensembles was discarded.

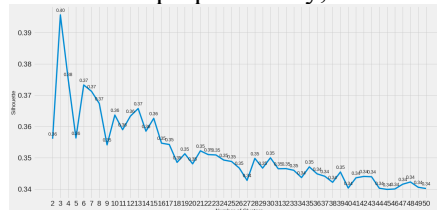
Finally, we were left to choose methods from partitioning and hierarchical approaches. We needed to select algorithms that were scalable based on the number of samples in our dataset, the number of clusters that could be produced, and the size of the generated clusters. In addition, the algorithms needed to perform well on flat geometry. After reviewing the literature, we decided to evaluate: KMeans, Mini-Batch KMeans, and Spectral (Partitioning); Ward Agglomerative and BIRCH (Hierarchical).

5.3 What is the correct number of clusters in the dataset and for the visualization tool?

All the algorithms we evaluated required the number of clusters in the dataset be used as a parameter. When focusing on a single algorithm, changing this parameter generates unique clustering models with diverse results and outputs. So, before we could compare the five clustering methods we selected, we needed to find the correct number of clusters for each one. Then, we could generate the “optimal” model for each algorithm and evaluate the results and outputs of the different methods.

We determined the correct number of clusters for a specific clustering algorithm by calculating the internal validity metrics for various parameter values. We focused on one algorithm at a time, and we looped through ranges of values with 50 as the absolute maximum. For each iteration, we created a clustering model using the processed dataset, the specific algorithm, and the current iterative value as the number of clusters. Then, we calculated the three internal validity measures for the generated output. Finally, the value that produced the results with the best overall measure for each metric was identified.

The visualization below is an example of the output produced by the described method. It plots the trend of the Silhouette coefficient for the KMeans algorithm with respect to the number of clusters over a range of 2 to 50. It is shown for demonstrative purposes only; true results can be found in the next table.



Moreover, the “elbow method” and the “gap statistic” were additional approaches that were used to further evaluate the KMeans and Mini-Batch KMeans algorithms. The “elbow method” is a heuristic for selecting the correct number of clusters by determining the “elbow” in the curve of the sum of within-cluster variances with respect to the number of clusters (Han et al., 2012). With the “gap statistic”, the inter-class distance matrix, Sw , was computed for the actual dataset and data generated from a uniform distribution for each iterative value. The correct number of clusters was where the widest gap appeared between the Sw of the actual data and the uniform data (Tibshirani et al., 2002).

We ran our proposed loop for the five clustering algorithms we were considering to find the correct number of clusters for each. The table below shows the correct number of clusters specified by the various internal validity metrics, with their values at that point in parentheses, for each clustering method.

| | Silhouette | Calinski-Harabasz | Davies-Bouldin | "Elbow Method" | Gap Statistic |
|-------------------|--------------|-------------------|--------------------|------------------|---------------|
| KMeans | 3 (.40) | 50 (72462.59) | 15 or 39 (0.77) | 10 (51231.32) | 34 (0.47) |
| Mini-Batch KMeans | 3 (.39) | 50 (69040.65) | 7 (0.78) | 10 (55459.62) | 6 (0.35) |
| Spectral | 48 (.322) | 50 (64344.15) | 2 (0.61) | | |
| Ward | 2 (.35) | 50 (58761.53) | 4 (0.87) | | |
| BIRCH | 3 (.35) | 50 (57272.38) | 4 (0.86) | | |

Unfortunately, we were unable to simply select a result set with a smaller number of clusters because of the computational problems that arose with our visualization tool. Results with smaller number of clusters had more members per cluster, so it took longer for the application to start and search results to load. As a result, there was a trade-off between choosing the correct number of clusters based on the internal validity measures, and choosing a high enough number of clusters to improve the efficiency of the visualization tool. We conducted some stress tests on the visualization tool by varying the number of clusters in the underlying dataset and measuring the application’s performance.

| Sample | 15 clusters - 5,235 cluster members in demo cluster | | 50 clusters - 1,384 cluster members in demo cluster | |
|--------|---|---|---|---|
| | Time (in sec.) from app start to server ready | Time (in sec.) from search of "Atlanta, GA" to results page loaded. | Time (in sec.) from app start to server ready | Time (in sec.) from search of "Atlanta, GA" to results page loaded. |
| 1 | 12.74 | 31.54 | 10.94 | 7.19 |
| 2 | 12.77 | 31.6 | 10.74 | 6.97 |
| 3 | 12.64 | 29.84 | 10.38 | 6.27 |
| 4 | 12.79 | 33.32 | 10.5 | 6.98 |
| 5 | 12.65 | 29.99 | 10.39 | 7.28 |
| Mean | 12.7 | 31.3 | 10.6 | 6.9 |
| SD | 0.1 | 1.4 | 0.2 | 0.4 |

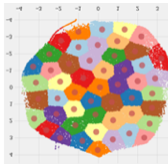
After increasing the number of clusters from 15 to 50, the time from “app start to server ready” and “search of ‘Atlanta, GA’ to results page loaded” improved by 16% and 78%, respectively. Thus, to improve the visualization tool’s user experience, it was beneficial to set the number of clusters to 50

because a lower number would have caused unreasonable load times for the user. Moreover, there was not a significant difference between the values of the internal validity measures when the number of clusters was set to 50 as opposed to values that were indicated by the measures. This is evident from the results in the next subsection. In fact, the Calinski-Harabasz index indicated that 50 was the correct number of clusters for all algorithms. Furthermore, we were hesitant to test models with higher numbers of clusters. To elaborate, when testing the range of values from 2 to 50, the pipeline took about 18 hours to run. Also, Spectral and Ward clustering took several hours to run even for ranges of values on the low end.

5.4 What clustering algorithm should be chosen for our visualization tool?

As discussed above, we used 50 as the parameter for the number of clusters for all five clustering methods. Then, a unique model was created for each algorithm using the processed data. The five models were compared by calculating and comparing the three internal validity metrics for each result set. The model with the best overall measures was chosen for the tool. The KMeans algorithm had the best results for two out of the three measures. In addition, the KMeans algorithm only took 22 seconds to train, while the Spectral algorithm took around 8 minutes. Therefore, the KMeans algorithm with 50 clusters was chosen as our final model. Here is a table of the results, and a 2D visualization that displays the clustering output and cluster centers of the final chosen model:

| | Silhouette | Calinski-Harabasz | Davies-Bouldin |
|-------------------|------------|-------------------|----------------|
| KMeans | 0.34 | 72462.59 | 0.79 |
| Mini-Batch KMeans | 0.33 | 69040.65 | 0.83 |
| Spectral | 0.32 | 64344.15 | 0.76 |
| Ward | 0.27 | 58761.53 | 0.91 |
| BIRCH | 0.26 | 57272.38 | 0.93 |



5.5 What are the features that make each cluster unique?

After retrieving the clustering labels from our final model, we wanted to determine the variables that accounted for the greatest differences between the various clusters. We grouped the data points based on the generated clusters, and we retrieved the mean value for each feature. Then, we calculated the variance of means between clusters within each feature, and we selected the top 7 with the highest variance.

| Variable | Standard Deviation |
|---|--------------------|
| # of parks within 25 miles | 0.203 |
| % of population that is white (single race) | 0.200 |
| # of gyms within 25 miles | 0.197 |
| # of grocery stores within 25 miles | 0.192 |
| # of hardware stores within 25 miles | 0.191 |
| # of medical facilities within 25 miles | 0.189 |
| % of housing that are multi-unit housing | 0.185 |

6 Conclusion

All team members have contributed similar amount of effort. Our innovative methods effectively solve the relocation problem. Our clustering and ranking approach represents an initial step in providing personalized recommendations of neighborhoods, which is not a feature offered by other tools. Users input an address they know and love, and they can view suggestions from across the US. In addition, they have the option to see more information about specific locations, such as Census statistics and amenities in the area. This gives users assurance in choosing the right neighborhood for them.

Although we are confident with our approaches, we still experienced several limitations. To elaborate, the Census data only provided a small number of viable features. There may have been more appropriate metrics for clustering neighborhoods, but they were not readily available. Furthermore, with around 73,000 observations in our dataset, there may have been more effective techniques to offer personalized recommendations, other than clustering. Unfortunately, other methods, such as a Naïve Bayes ranking on features, would have required historical data on preferences and choices, which was simply not feasible.

Moreover, there are several ways we can improve our approaches in future updates. The first upgrade could be the generation of multiple clustering results for different feature subsets. This would allow users to filter for different neighborhoods based on specific categories. To ensure geographically distributed clusters, we could use a multi-objective optimization model. Cluster ensembles could also improve our method by ensuring the stability of our clustering results. Unfortunately, we were unable to incorporate these elements into our project due to time constraints.

Works Cited

- Abdi, H. & Williams, L. (2010). Principal Component Analysis. *WIREs Computational Statistics*, 2(4). <https://doi.org/10.1002/wics.101>
- Al-Ghossein, M., Abdessalem, T., and Barré, A. (2018, July 8-11). Exploiting Contextual and External Data for Hotel Recommendation [Paper Presentation]. In *UMAP '18: Adjunct Publication*. 26th Conference on User Modeling, Adaptation and Personalization, Nanyang Technological University, Singapore (pp. 323-328). Association for Computing Machinery. <https://doi.org/10.1145/3213586.3225245>
- Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Perez, J., & Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1), 243-256. <https://doi.org/10.1016/j.patcog.2012.07.021>
- Ashkezari-Toussi, S., Kamel, M., & Sadoghi-Yazdi, H. (2019). Emotional maps based on social networks data to analyze cities emotional structure and measure their emotional similarity. *Cities*, 86, 113-124. <https://doi.org/10.1016/j.cities.2018.09.009>
- Ayub, B., McGough, T., Boukamp, F., & Naderpajouh, N. (2020). Housing market bubbles and urban resilience: Applying systems theory. *Cities*, 106. <https://doi.org/10.1016/j.cities.2020.102925>
- Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46, 109-132. <https://doi.org/10.1016/j.knosys.2013.03.012>
- Boongoen, T., & Iam-On, N. (2018). Cluster ensembles: A survey of approaches with recent extensions and applications. *Computer Science Review*, 28, 1-25. <https://doi.org/10.1016/j.cosrev.2018.01.003>
- Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A., Fofou, S., & Bouras, A. (2014). A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis. *IEEE Transactions on Emerging Topics in Computing*, 2(3), 267-279. <https://doi.org/10.1109/TETC.2014.2330519>
- Garcia-Lopez, M., Jofre-Monseny, J., & Segu, M. (2020). Do short-term rental platforms affect housing markets? Evidence from Airbnb in Barcelona. *Journal of Urban Economics*, 119. <https://doi.org/10.1016/j.jue.2020.103278>
- Halkidi, M., Batistakis, Y., Vazirgiannis, M. (2001). On Clustering Validation Techniques. *Journal of Intelligent Information Systems*, 17, 107-145. <https://doi.org/10.1023/A:1012801612483>
- Han, J., Kamber, M., Pei, J. (2012). Chapter 10: Cluster Analysis: Basic Concepts and Methods. *Data Mining: Concepts and Techniques* (3rd ed., pp. 443-495). Boston: Morgan Kaufmann. <https://doi.org/10.1016/B978-0-12-381479-1.00016-2>
- He, Y., & Xia, F. (2020). Heterogeneous traders, house prices and healthy urban housing market: A DSGE model based on behavioral economics. *Habitat International*, 96. <https://doi.org/10.1016/j.habitatint.2019.102085>
- Kim, T., Pan, S., & Kim, S. (2020). Sentiment Digitization Modeling for Recommendation System. *Sustainability*, 12(12). <https://doi.org/10.3390/su12125191>

- Kiseleva, J., Mueller, M., Bernardi, L., Davis, C., Kovacek, I., Einarsen, M., Kamps, J., Tuzhilin, A., & Hiemstra, D. (2015, August 9-13). Where to Go on Your Next Trip? Optimizing Travel Destinations Based on User Preferences [Paper Presentation]. In *SIGIR '15: Proceedings*. 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile (pp. 1097-1100). Association for Computing Machinery. <https://doi.org/10.1145/2766462.2776777>
- Kopf, D. (2018, November 30). Americans are moving less than ever, and it's bad for the economy. *Quartz*. <https://qz.com/1480835/the-share-of-americans-moving-hit-a-record-low/>
- Lawson, R., & Jurs, P. (1990). New index for clustering tendency and its application to chemical problems. *Journal of Chemical Information and Computer Sciences*, 30(1), 36-41. <https://doi.org/10.1021/ci00065a010>
- Liu, Y., Li, Z., Ziong, H., Gao, X., & Wu, J. (2010, December 13-17). Understanding of Internal Clustering Validation Measures [Paper Presentation]. In *IEEE ICDM '10*. IEEE International Conference on Data Mining, Sydney, Australia (pp. 911-916). IEEE Computer Society Press. <https://doi.org/10.1109/ICDM.2010.35>
- Liu, Y., & Yang, J. (2015). Improving Ranking-based Recommendation by Social Information and Negative Similarity. *Procedia Computer Science*, 55, 732-740. <https://doi.org/10.1016/j.procs.2015.07.164>
- Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86), 2579-2605. <http://jmlr.org/papers/v9/vandermaaten08a.html>
- Nicholson, D., Vanli, O.A., Jung, S., & Ozguven, E.E. (2019). A spatial regression and clustering method for developing place-specific social vulnerability indices using census and social media data. *International Journal of Disaster Risk Reduction*, 38. <https://doi.org/10.1016/j.ijdrr.2019.101224>
- Okulicz-Kozaryn, A. (2011). City Life: Rankings (Livability) Versus Perceptions (Satisfaction). *Social Indicators Research*, 110, 433-451. <https://doi.org/10.1007/s11205-011-9939-x>
- Rendón, E., Abundez, I., Arizmendi, A., & Quiroz, E. (2011). Internal verses External cluster validation indexes. *International Journal of Computers and Communications*, 5(1), 27-34.
- Renjith, S., Sreekumar, A., & Jathavedan, M. (2020). An extensive study on the evolution of context aware personalized travel recommender systems. *Information Processing & Management*, 57(1). <https://doi.org/10.1016/j.ipm.2019.102078>
- Sadler, R.C., Hippensteel, C., Nelson, V., Greene-Moton, E., & Furr-Holden, C.D. (2019). Community-engaged development of a GIS-based healthfulness index to shape health equity solutions. *Social Science & Medicine*, 227, 63-75. <https://doi.org/10.1016/j.socscimed.2018.07.030>
- Shashank, A., & Schuurman, N. (2019). Unpacking walkability indices and their inherent assumptions. *Health & Place*, 55, 145-154. <https://doi.org/10.1016/j.healthplace.2018.12.005>
- Su, Z., Zheng, X., Ai, J., Shen, Y., & Zhang, X. (2020). Link prediction in recommender systems based on vector similarity. *Physica A: Statistical Mechanics and its Applications*, 560. <https://doi.org/10.1016/j.physa.2020.125154>

- Sun, X., Huang, Z., Peng, X., Chen, Y., & Liu, Y. (2019). Building a model-based personalised recommendation approach for tourist attractions from geotagged social media data. *International Journal of Digital Earth*, 12(6), 661-678. <https://doi.org/10.1080/17538947.2018.1471104>
- Tavernise, S. (2019, November 20). Frozen in Place: Americans Are Moving at the Lowest Rate on Record. *The New York Times*. <https://www.nytimes.com/2019/11/20/us/american-workers-moving-states-.html>
- Tibshirani, R., Walther, G., & Hastie, T. (2002). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2). <https://doi.org/10.1111/1467-9868.00293>
- U.S. Census Bureau. Economics and Statistics Administration. (1994, November). Census Tracts and Blocking Numbering Areas. In *Geographic Areas Reference Manual* (pp. 10.1-10.15). Retrieved from <https://www.census.gov/programs-surveys/geography/guidance/geographic-areas-reference-manual.html>
- Wasid, M., & Ali, R. (2018). An Improved Recommender System based on Multi-criteria Clustering Approach. *Procedia Computer Science*, 131, 93-101. <https://doi.org/10.1016/j.procs.2018.04.190>