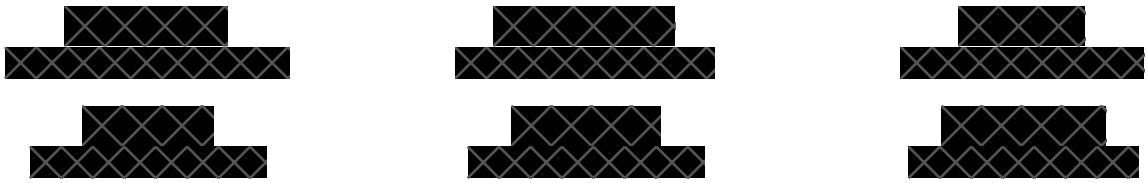
Apartment Recommendation System in New York



1 INTRODUCTION

Each year, millions of people in the world decide or are forced to migrate. It results in flows of people moving from one settlement to another. Then the first difficulties they encounter would be how to find an apartment when they know nothing about the city.

There are plenty of services that offer a centralized search for long-term accommodation, such as Zillow, Apartments, but none of these sites or any other products we have been able to find offer extensive searching capabilities considering different attributes of a living space. Hence, we build an online platform that provides proper recommendations for apartment seekers with the options to take multiple factors into account. We expect to launch a user-oriented platform with an integrated information system for customers that can interactively obtain and filter information based on multiple choices.

2 PROJECT OVERVIEW

2.1 Problem Definition

Our project is to establish a comprehensive living space recommendation website, which allows users to filter factors with customized weight. The website also integrates multiple rating schemes with respect to different factors, therefore form a overall score for user to make the decision.

2.2 Innovations

(1) Propose a map-based interface that present an intuitive and comprehensive InfoMap for living space recommendation.
(2) Design an effective filtering feature that allows users to filter cared factors.
(3) Introduce a new criteria for recommendation ratings that incorporates ratings scheme from different factors.

3 ACADEMIC SURVEY

Paper[17] proposed a tag-based interaction framework to recommend resources for different users based on the tag information associated with different resources. We will mimic this idea to provide users with a map-based interface with the options to filter down with multiple options to improve the apartment search efficiency.

Paper[13] also introduced a novel tag-based collaborative filtering approach for the recommendation system, which is based on the distinctive three dimensional relationship between users, tags and items. Similarly, our project will make a tag-based filter section, like crime, accident, transportation, prices etc, by integrating scattered data sources into one so that users can have a much quicker decision-making process based on the information provided.

3.1 Data Process and Visualization

Since we will process a large amount of housing dataset, crime dataset, transportation dataset etc, paper[3] gave a good direction for us to visulize our data.

Paper[11] used a GIS application to find the relationship between land use patterns and crime rates by giving the geographical visualization. We will generalize this idea to fit our living spaces recommendation filtered by categories.

Besides, Stephen Few introduced the methods for encoding quantitative data on maps like bar graphs, choropleth maps, or circles that vary in size on map, and he also gave the idea of visualizing the data on map by featuring the data[8].

3.2 Recommendation Models

An user-oriented recommendation system model is our goal, and paper[18] gave us such a good inspiration to implement a user-oriented recommendation system combined CBR and ontological structure.

Filter techniques for selection of contents and products has been a popular technique to prune large information so that users' needs can be satisfied at the best[12]. Collaborative filtering is one of the most popular way. The conventional idea of collaborative filtering is to analyze the target user's preference by other users with similar behaviors or interests, and then integrate the preferences of other similar users to recommend the resources[17]. The development of tag-based personalized recommendations is also worth discussing[14].De Gemmis et al.[6] proposed to use the tag information to facilitate the semantic content analysis.

Since our goal is to build up the recommendation system for people who are looking for the living space, the features of locations and transportation are keys should be considered. Paper[7, 15] determined location and transportation based services APIs using machine learning model.

In the process of building up the models, the bias will be the most significant risk and the complexity of the algorithm will affect the efficiency of the recommender system. Those problems will seriously affect the user experience[2]. Paper[1] investigated the popularity bias from the users' perspective and concluded that all algorithms recommend items that are more popular than users in the group of niche, diverse and Blockbuster-focused ratings. To solve the popularity bias, causal embedding is introduced to utilize causal-specific data to guide model learning[5].

3.3 Models for Review Analysis

For comprehensive description of every specific apartment, we want to provide user with some keywords or tags using existing online reviews.Paper [4] has built a sentiment summarizer to extract meaningful information for target people. Paper [10] use aspect-based summerization models to mine the target item features and identify whether the review is positive or negative. This model takes the reviews and produce several relevant aspects and an aggregate score for each aspects. We use it to furthermore clearly evaluate the apartments.

4 PROPOSED METHOD

The method we propose can divide into two parts: back-end data analysis and front-end interactive data visualization.

4.1 Intuition

Consider the geometric feature of our data, we introduce the clustering algorithm to deal with the analysis (Comparing K-means and Affinity, both are helpful clustering algorithms). It will help us to evaluate not just a single apartment, but the living space of a community. And we give ratings to the convenience level of transportation based on distance since distance is the key point when we consider the transportation situation around.

In front end part, in order to give users more options and flexibility, we introduce noUislider to create sliders for filtering function. And for better user experience, we use a fluid responsive grid so that our product will fit any size of page. Moreover, our UI design is focusing on the surrounding environment of apartments, such as restaurant distribution.

4.2 Data Collection

We use python to scrape data from apartments from Zillow. The dataset includes the price, address which can help form the basic of our apartment recommendation system. For the next step to rate the living space of apartments in New York, we consider multiple factors like transportation, crime, dinning. We collect restaurant data from Yelp API , bus and subway data from government website and historical crime data from NYPD Crime Database API. We also get top reviews from Yelp to extract keywords about apartments. We successfully get about 1000 apartments' data and more data about crime, restaurant, transportation in New York.

4.3 Data Integration and Processing

For the first step, we clean the data from original data sets. We extract the required feature(such as the address, price) and we match the data from Zillow and Yelp. We filter out some outliers such as data about street.

Then, we will give scores(see rating schema in 4.3) on different factors to help standardize different kind of data. This step is important for our recommendation system. With standardized data input, we can apply our model to give an overall score to recommend the apartments that satisfy users' need.

Furthermore, we analyze the review data to extract keyword on each apartment to provide more information on certain apartment.

4.4 Rating Scheme of Factors

4.4.1 Geospatial data clustering for restaurants and crime data. For restaurant, we comprehensively consider the single rating, transaction method and diversity of categories to assign a score for each cluster.

$$\text{Score}_{\text{res}} = S_1 + S_2 + S_3$$

where S_1 = mean(restaurant rating) for each cluster. S_2 add 1 if delivery or pickup appears in transaction. S_3 = unique(categories) in each cluster.

For crime, we consider the number of crime events happened in recent years, and give scores to each clusters based on that.

Thus, the first step of giving crime score and restaurant score is to use algorithm to cluster the apartment, restaurant and crime event. And then, we assign cluster-based raw score to apartments in the corresponding cluster and normalize the score to the range of $[0, 5]$.

For clustering algorithms, we've tried K-means and Affinity Propagation. K-means is the most basic clustering algorithm that partition the dataset into predetermined clusters with the nearest mean. Using squared Euclidean distance to evaluate the similarity between data points $x_i, x_{i'}$:

$$d\left(x_i, x_{i'}\right) = \sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^2 = \left\|x_i - x_{i'}\right\|^2$$

For each iteration of data reassigning, we try to minimize the objective function where μ_k denote the centroid of the cluster:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \left\|\mathbf{x}_n - \mu_k\right\|^2$$

where μ_k is the centroid of the cluster. K-means is easy to implement but need suitable random initialization for the convergence of the solution.

Unlike K-means, Affinity Propagation (AP) is a graph-based algorithm that does not require the number of clusters to be predetermined[9]. By viewing each data point as a node in a network, AP recursively transmits real-valued messages along edges of the network. There are two kinds of message exchanged between data points: "responsibility" $r(i,k)$ and "availability" $a(i,k)$. They are computed as below:

$$r(i,k) \leftarrow s(i,k) - \max_{k' \text{s.t.} k' \neq k} \left\{a\left(i,k'\right) + s\left(i,k'\right)\right\}$$

$$a(i,k) \leftarrow \min\left\{0, r(k,k) + \Sigma_{i' \text{ s.t. } i' \notin \{i,k\}} \max\left\{0, r\left(i',k\right)\right\}\right\}$$

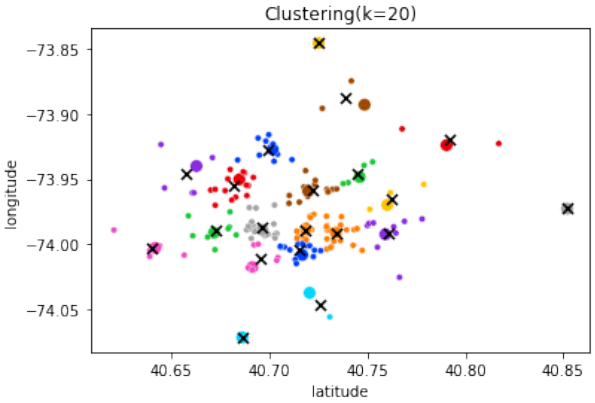


Figure 1: K-means Clustering for Restaurants

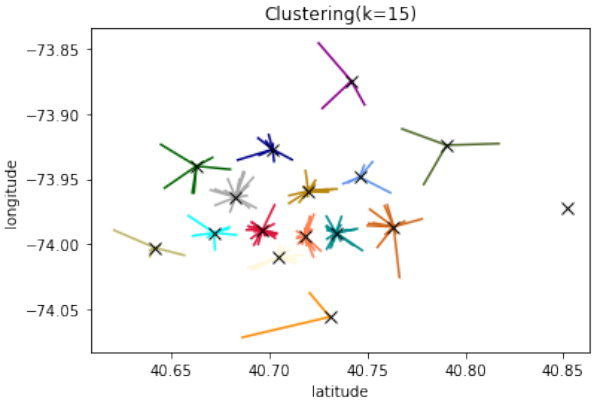


Figure 2: Affinity Propagation Clustering

For clearer visualization, we choose random data points from the whole dataset and use a smaller cluster number. We can observe the difference from Figure 1 and 2 that K-means have even clusters based on distance while AP works better with non-flat geometry. And also according to our result, AP algorithm has better performance on large amount of clusters which will help us better distinguish the geometric features in city, and it's convenient that we do not need to specify the number of the clusters when data source changes.

Therefore, we choose Affinity Propagation clustering algorithm to help us calculate the restaurant and crime score. Every apartment and restaurant will be in a certain cluster. According to restaurants data, we will give a res-score to the cluster, all the apartment in the same cluster will have the same score. It works similarly with crime score.

4.4.2 Rating of transportation data based on distance. Apart from restaurants and crime rate, easy access to public transportation is an important factor for apartment seekers as well. Therefore, we also provide score to measure the apartment proximity to public transit.

In this case, we mainly consider access to bus and subway station, and justify whether the apartments have easy access to public transit by distance in-between. Two scores are calculated independently, then combined as an overall public transit score.

$$\text{Score}_{\text{tran}} = \frac{1}{2}\left(S_{\text{bus}} + S_{\text{subway}}\right)$$

In particular, for each public transit dataset, we define each location using longitude and latitude, and obtain the actual distance between transit station and apartment based on Haversine formula.

$$\text{a} = \sin^2\left(\frac{\Delta\alpha}{2}\right) + \cos\alpha_1 \times \cos\alpha_2 \times \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

$$\text{distance} = 2 \times \text{atan2}\left(\sqrt{a}, \sqrt{(1-a)}\right) \times R$$

where α is latitude, λ is longitude, R is the earth radius.

Then we further formulate the scores as:

$$S_{\text{bus}} = \frac{1}{\text{distance}^2},\ S_{\text{subway}} = \frac{1}{\text{distance}^2 - 1000}$$

Finally, we normalize the raw scores to the range of $[0, 5]$.

4.5 Apartment Review Analysis

We design part of apartment review analysis, aiming to provide intuition for users to select apartment. We use scikit-learn library and natural language toolkit to implement the processing and TF-IDF algorithm [16] to extract top 3 keywords for each apartments.

$$\text{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

$$\text{idf}(t,D) = \log\frac{N}{|\{d \in D : t \in d\}|}$$

Then tf-idf is calculated as

$$\text{tfidf}(t,d,D) = \text{tf}(t,d) \cdot \text{idf}(t,D)$$

We can therefore get lists of keywords according to the tf-idf score for each word from high to low. The keywords will shown in apartment pop-up window.

4.6 Front-end

In the front-end part, we did not do the attribute keyword filtering function in the initial design. Because in the process of collecting data, we found that our sliding filters can basically generalize the information of the collected attribute keywords. We believe that the function of attribute keyword filtering does not greatly improve the user experience, so we did not do it. Apart from that, we implemented everything from the initial idea. We divide the interface into left and right sections, the filter and the map, and use the standard 12 column fluid responsive grid system to lay out the page, so that the content can be automatically adjusted according to the size of the window opened by the user. We've added a list below the map area that shows all apartments that match the filter. We have included a picture of the apartment to give users a better understanding of the apartment.

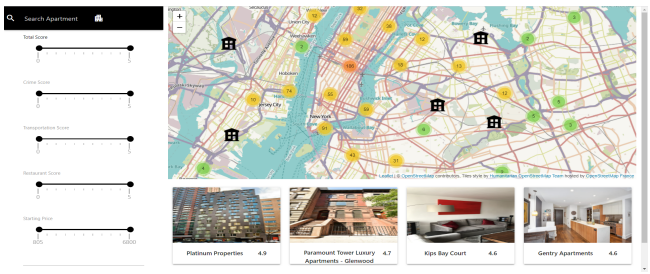


Figure 3: Front-end Style

4.7 Front-end Filter

In order to provide an effective filtering feature that allows users to shortlist the living space recommendations based on adjusted factors, we create an all-in-one filter using Leaflet in JavaScript. An all-in-one filter is realized by integrating the rating schema of all factors, including crime rate, number of restaurants, access to public transportation and rental price. One can adjust the level of importance on each factor using slider bars, or search by the apartment name or its keywords directly in the search bar, as shown in figure 3.

To the side of the map is the all-in-one filter for users to search apartments conditioned on different weighted factors. Once the weights are determined, the overall scores will be calculated for each apartment, and a list of apartments ranked by overall scores will display to the bottom of the map. Each apartment can be located

on the map correspondingly by a single click. Users can also clear the current filters by clicking the "Zoomback" button.

4.8 Front-end Map

The User Interface (UI) of our Living Space recommendation system prototype can be seen in figure 3. We go with a map-centered UI as inspired by Google Maps, where user can obtain the overall picture by changing the zoom level or adjusting the center of the map.

We completed the basic configuration and user interaction of the map using Leaflet and jQuery. We extracted longitude and latitude and all apartment detailed information from GeoJson file to mark apartments on the map using Leaflet.

The circled numbers indicate how many apartments locate within the community, where the house markers represent each specific apartment. A single click on circled number will zoom in the underlying community where users are able to see distinct apartments and neighborhood amenities. The scope of underlying community can be seen by hovering over the circled number, as show in figure 4.

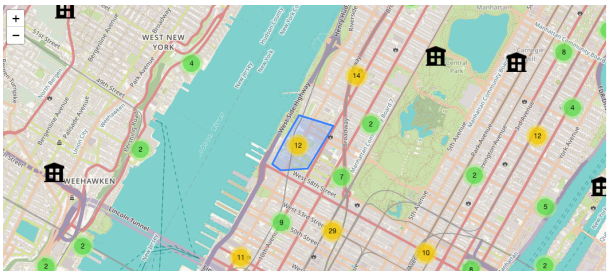


Figure 4: Apartment Cluster

In addition, users can also see a pop-up with detailed apartment information when hovering over the house marker. As shown in figure 5, the pop-up displays the name, overall score and top 3 keywords of the apartment extracted from their reviews.

In addition, by clicking the house marker on map, one can also see the nearby restaurants, as shown in figure 6.

5 EVALUATIONS

5.1 Survey Introduction

To better evaluate our platform, we made a Google survey to test our online platform. We invite our friends,

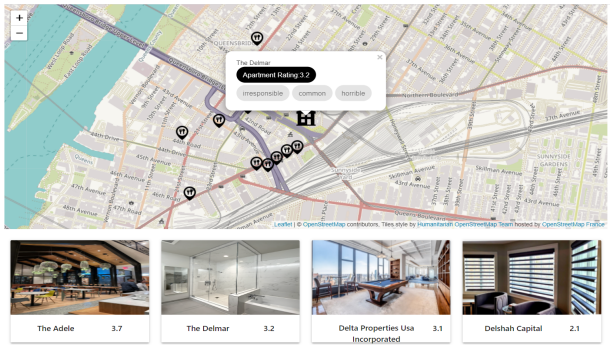


Figure 5: Pop-up Window

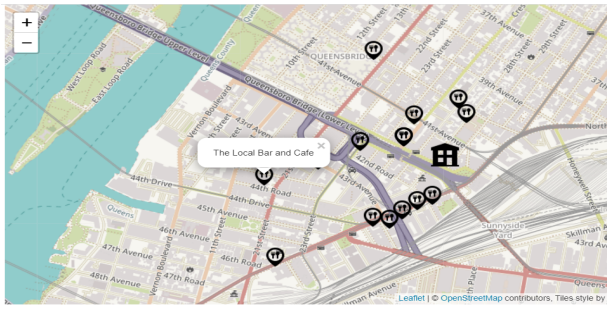


Figure 6: Nearby Restaurants

collect feedback from them to optimize our website and let users have a better user experience in the future. The survey link is https://forms.gle/ikqqiujHNZiXk2JCA.

The survey is designed with 13 questions. Questions 1-10 collect evaluations of the specific features, questions 11-12 collect the basic information like gender and student status, and question 13 is designed for suggestions.

Q#	Feature	Avg Score
Q3	Apartment Name Keyword Search	3.328
Q4	Factors in Filter Section	3.397
Q5	Index Score in Filter Section	3.397
Q6	Apartment Pictures	3.362
Q7	Restaurants Near Apartments	3.345
Q8	Apartment Information	3.517
Q9	Locate Apartment Picture on Map	3.569
Q10	Cluster Apartments Region	3.603

Table 1: Feature Average Score

5.2 Survey Results

We received 58 responses in total from April 19 to April 21. Q11 and Q12 shows that 70.7% of the participants are master students and 51.7% of the participants are male. Q1 and Q2 give the general feedback of the platform design. 50% of the participants are very satisfied with our platform and the majority of the participants thought the filter section, map design, and apartment design were excellent. From Q3 to Q10, we specified the features and designs to the participants and received the vote score. (4 — very helpful, 3 — helpful, 2 — little helpful, 1 — not helpful). Table 1 shows the average score of each factors (total score is 4). The scores show that we have done a good job at clustering region and apartment locating, but it needs to be proved at keyword search.

5.3 Improvements

We received the suggestions in Q13. After reviewing the suggestions, we will improve our platform from the following perspectives in the future.

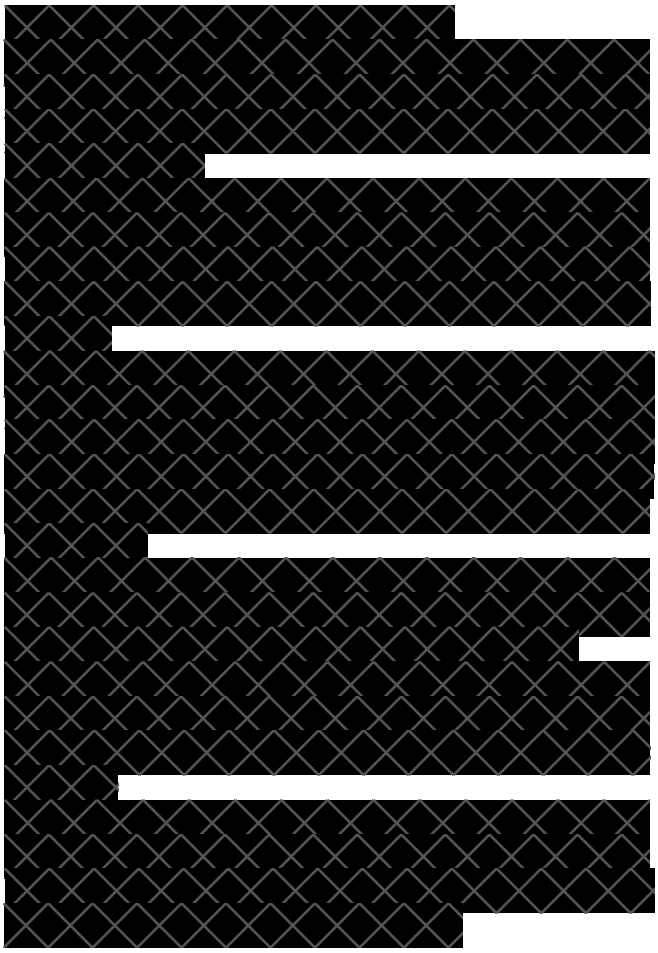
- Adding the feature of 3D maps to view the location and the room setting of apartments.
- Adding more features in the filter section.
- Improve the score index in the filter section. For example, adding a pop up note to clarify the meaning of the score.

5.4 Comparison

At the end, we randomly selected some friends at Georgia Tech who had apartment-seeking experience to have short interviews about comparison our apartment platform with the Zillow website in terms of strengths and weaknesses. Below are the feedbacks from three friends:

- Friend A: Compared with Zillow, this platform can select more factors as filters and we can locate the apartments through clicking the images. However, this platform cannot choose the house type.
- Friend B: I feel that this platform is a good start when looking for a house. It provides more information and is more comprehensive.
- Friend C: I think the overall page display is more intuitive than Zillow, and I can directly select the corresponding metrics and housing resources. However, the number of housing resources displayed by several results is relatively limited.

6 DISTRIBUTION OF EFFORT



7 CONCLUSION AND DISCUSSION

We implemented an interactive apartment selection system with integrated information to be shown. According to the filter section, our platform will successfully provide users with information to help them find their ideal living space and have a more comprehensive cognition of the area, such as safety, dining, transportation and so on.

In the future, the platform can be improved by adding more functional modules such as filtering by keywords, taking user browsing behavior into consideration of recommendation. Most importantly, to reinforce the outcome of our project, data needs to be updated in regular time to provide users with latest apartment information.

REFERENCES

[1] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2019. The unfairness of popularity bias in recommendation. *arXiv preprint arXiv:1907.13286* (2019).

[2] Gediminas Adomavicius, Jesse Bockstedt, Shawn P Curley, Jingjing Zhang, and Sam Ransbotham. 2019. The hidden side effects of recommendation systems. *MIT Sloan Management Review* 60, 2 (2019), 1.

[3] Paul A Austin, Marco Marini, Alberto Sanchez, Chima Simpson-Bell, and James Tebrake. 2021. Using the Google Places API and Google Trends Data to Develop High Frequency Indicators of Economic Activity. *IMF Working Papers* 2021, 295 (2021).

[4] Sasha Blair-Goldensohn, Kerry Hannan, Ryan McDonald, Tyler Neylon, George Reis, and Jeff Reynar. 2008. Building a sentiment summarizer for local service reviews. (2008).

[5] Jiawei Chen, Xiang Wang, Fuli Feng, and Xiangnan He. 2021. Bias Issues and Solutions in Recommender System: Tutorial on the RecSys 2021. In *Fifteenth ACM Conference on Recommender Systems*. 825–827.

[6] Marco De Gemmis, Pasquale Lops, Giovanni Semeraro, and Pierpaolo Basile. 2008. Integrating tags in a semantic content-based recommender. In *Proceedings of the 2008 ACM conference on Recommender systems*. 163–170.

[7] Sarah A Elariane. 2022. Location based services APIs for measuring the attractiveness of long-term rental apartment location using machine learning model. *Cities* 122 (2022), 103588.

[8] Stephen Few and Perceptual Edge. 2009. Introduction to geographical data visualization. *Visual Business Intelligence Newsletter* 2 (2009).

[9] Brendan J. Frey and Delbert Dueck. 2007. Clustering by Passing Messages Between Data Points. *Science* 315, 5814 (2007), 972–976. https://doi.org/10.1126/science.1136800 arXiv:https://www.science.org/doi/pdf/10.1126/science.1136800

[10] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 168–177.

[11] Jianling Li and Jack Rainwater. 2000. The real picture of land-use density and crime: a gis application. In *Proceedings of the 20nd annual ESRI International User Conference*.

[12] Qing Li, Yuanzhu Peter Chen, and Zhangxi Lin. 2008. Filtering techniques for selection of contents and products. *Personalization of Interactive Multimedia Services: A Research and development Perspective* (2008).

[13] Huizhi Liang, Yue Xu, Yuefeng Li, and Richi Nayak. 2009. Tag based collaborative filtering for recommender systems. In *International Conference on Rough Sets and Knowledge Technology*. Springer, 666–673.

[14] Shaowei Wang, David Lo, Bogdan Vasilescu, and Alexander Serebrenik. 2018. EnTagRec++: An enhanced tag recommendation system for software information sites. *Empirical Software Engineering* 23, 2 (2018), 800–832.

[15] Glen Weisbrod, Steven R Lerman, and Moshe Ben-Akiva. 1980. Tradeoffs in residential location decisions: Transportation versus other factors. *Transport Policy and Decision Making* 1, 1 (1980), 13–26.

[16] Ho Chung Wu, Robert Wing Pong Luk, Kam Fai Wong, and Kui Lam Kwok. 2008. Interpreting tf-idf term weights as making relevance decisions. *ACM Transactions on Information Systems (TOIS)* 26, 3 (2008), 1–37.

[17] Qing Xie, Feng Xiong, Tian Han, Yongjian Liu, Lin Li, and Zhifeng Bao. 2018. Interactive resource recommendation algorithm based on tag information. *World Wide Web* 21, 6 (01 Nov 2018), 1655–1673. https://doi.org/10.1007/s11280-018-0532-y

[18] Xiaofang Yuan, Ji-Hyun Lee, Sun-Joong Kim, and Yoon-Hyun Kim. 2013. Toward a user-oriented recommendation system for real estate websites. *Information Systems* 38, 2 (2013), 231–243.