Evaluation of Property Filtering Algorithms Using Tags for a Property Rental Recommendation Application

Sean Lynch, Stephen Anokye, Oisín Fitzpatrick, Fenna Kadir, Charlotte Martin, Nishant Kapur, Brendan Tierney, Andrea Curley, Damian Gordon
School of Computer Science, Technological University Dublin, Ireland
e-mail: brendan.tierney@tudublin.ie, damian.x.gordon@tudublin.ie, andrea.f.curley@tudublin.ie

***Abstract*— Renting Made Easy is a project to create an easy-to-use, aesthetically pleasing rental listing application. The application provides a range of functionalities, including a comprehensive search experience, applying and managing rent applications, and property browsing. Initially based on the Zillow data set, the application has been expanded to include data sets with location-based services and crime-related data. A core feature is to provide a user-driven feature search based on property Tags. Property filtering algorithms were evaluated to determine which one provides suitable properties to the end-users. These algorithms included k-Nearest Neighbors (kNN) and Collaborative filtering. Qualitative research was performed to assess the usefulness and accuracy of the filtering algorithms.**

Keywords - Data Science; Recommendation System; Collaborative Filtering; User Evaluation.

I. INTRODUCTION

Renting Made Easy (RME) is a collaborative project involving students with a diverse range of skills including front-end development, back-end development, Application Development Interfaces (APIs) development, data science and machine learning. The resulting application reduces the anxiety of prospective tenants when searching for accommodation by streamlining the process. The system was developed with data sets including property listing information provided by Zillow [1], coordinate data from Google Maps, and various crime data from Open Baltimore [5]. With this data, a set of property services scores and crime safety scores were created [2]. Additionally, the application includes a recommendation system that leverages the collected data to provide property suggestions to users. This system employs content-based filtering to match properties with user preferences ensuring personalized and relevant recommendations.

The evaluation approaches included usability testing, accessibility testing, cognitive walkthrough, the think-aloud protocol, and expert feedback. The feedback indicated that the application delivered an intuitive and easy-to-use property rental website that displays standard and novel property details to users more clearly than other existing websites and provides users with property suggestions using a built-in recommendation system.

The recommendation system was built using a combination of a Property Scoring System, Property Tag selection and filtering algorithms. The filtering algorithms evaluated included kNN recommendation system and User Collaborative Filtering using Cosine Similarity. These different approaches were evaluated using subject matter experts. These were selected from various industry-related roles including property rental agents, estate agents, property owners and renters in the 20–35-year-old range.

II. RENTING MADE EASY

The Renting Made Easy project was designed to enhance the user experience by allowing tenants to filter properties based on lifestyle suitability in different Baltimore (USA) regions. Each property listing featured detailed scores that evaluated the availability of nearby services, and the safety levels based on local crime data. To enhance the personalized experience, user profiles included features such as saved searches, favorite properties, and a section for tracking property applications.

An integral part of RME was its recommendation system which utilized content-based filtering to suggest properties. The project aimed to surpass existing websites by providing clearer, more detailed property information and tailored property suggestions, thereby ensuring a more user-friendly and efficient property rental process.

The project team comprised two developers focused on front-end software development and User eXperience (UX), one developer dedicated to back-end software development, and three specialists responsible for building the recommendation engine and managing the data infrastructure. Figure 1 illustrates the core technical architecture of the project.

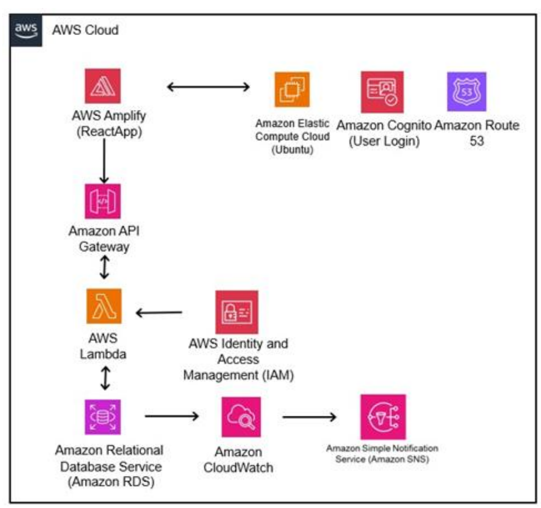

Figure 1. Basic System Architecture.

Architecture and deployment were cross-team responsibilities, as was project management. The technical architecture and technical flow (Figure 2) include: frontend, backend, data systems and data storage. Figure 2 illustrates the favourited properties and the recommendation engine process, including kNN and Cosine Similarity [3] [4].

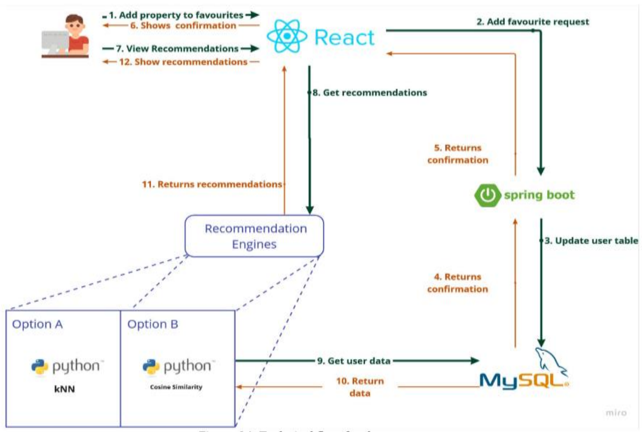

Figure 2. Technical Flow Incorporating Recommendation Engines.

In addition, data ingress from external APIs was incorporated. Due to legal limitations on storing data from Google's Map and Places APIs, a temporary caching strategy was required. This data is then combined with the Properties data set and used by the recommendation engines, scoring systems and tag categories. The entire application was deployed on Amazon Web Services (AWS) with AWS Lambda functions being used for the deployment of the recommendation engine.

III. RECOMMENDATION SYSTEMS

Property scores were generated for each property and displayed as values out of five to users. These scores provided a general overview of the area of each property. This first score displays the crime safety rating based on the neighbourhood of the property.

There was a data balancing issue when generating the safety score ratings. Some crime categories will naturally have higher counts than others due to their frequency which introduces a bias. This bias needs to be addressed to ensure that, when generating an overall crime score, these more common categories do not disproportionately influence the final result. Adjustments or normalization techniques should be applied to balance the impact of different crime categories, allowing for a more accurate and fair representation of crime levels. To solve this problem, a z-score standardisation was implemented. This generates a different score for each category in each neighbourhood. These scores were described in terms of their relationship to the mean, where their values are measured in terms of standard deviations from the mean. For consistency across the application the z-scores were mapped between one and five using sigmoid transformation. A sigmoid function was used over other mapping techniques following experimentation, including Min-max normalization, Winsorized min-max normalization, and Winsorized linear transformation. Using these mapping techniques resulted in unbalanced scores where the outputs were moving the values closer to either one or five. With a sigmoid mapping function, the values closer to the mean could be increased, preventing outliers from overpowering the results. Figure 3 displays the mapping function to map the z-scores. The sigmoid function was inverted, as the goal was to ensure that areas with a low crime rating have a higher safety rating.

The values closer to the mean were increased more than the values further away from the mean. This was because the extreme outliers with the higher summed z-scores were originally pushing these values closer to one after the mapping. This indicated that they were in safe areas. This was not the case. The values around the mean still had high levels of crime, but the distribution of the data was preventing this from being represented correctly. The sigmoid mapping function was shaped so that extreme outliers were increased marginally, and the values that lay before them were increased substantially.

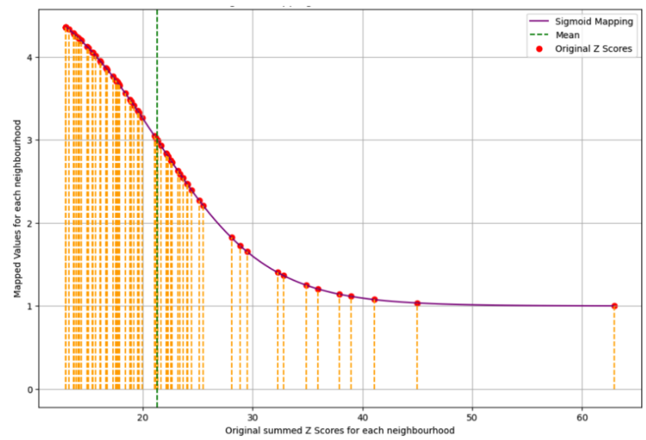

Figure 3. Sigmod Mapping of Z-Scores with Mean Life.

Finally, these z-scores were used to produce an overall crime safety score. The z-scores for each crime category were aggregated for each neighbourhood. Afterwards, the z-scores were standardized against each neighbourhood to generate the overall crime safety score. Using a sigmoid mapping function these z-scores were mapped between 1 and 5. The same approach was used to calculate the property nearby service score. These scores incorporated user weights into their calculation. Each weight represents a user's interest in each category.

A kNN model was incorporated within the system, including the steps for data pretreatment and feature selection. The kNN model obtained its data from a CSV file that included property listings and their corresponding attributes. The preprocessing steps included:

- **Dealing with Missing Values**: The absence of data can have a substantial effect on the accuracy of suggestions. Missing values in columns such as bathrooms, bathroomsFull, and bedrooms were imputed with zeros, based on the idea that the absence of a value may be adequately substituted by '0'.

 ISBN: 978-1-68558-187-9

- **Data Normalization**: This involved the use of MinMaxScaler. This ensured that these features had equal contributions in the distance calculations carried out by the kNN algorithm.
- **Categorical Encoding**: The OneHotEncoder was utilized to convert categorical variables into a format suitable for machine learning algorithms.

The feature selection process entailed selecting property qualities that have the greatest impact on a user's decision to rent. For example, considerations such as the number of bedrooms, cost, and accessible amenities were deemed crucial. The dataset has 72 features which were examined to determine their significance and influence on the recommendation results.

The kNN model was trained using the pre-processed and encoded dataset to select attributes closest in similarity based on their features. The training process entailed the following:

- Instantiating the kNN model using the NearestNeighbors class from the scikit-learn library.
- Applying the model to the dataset that has been both normalised and encoded.
- Evaluating the model using a certain attribute and obtaining the closest suggestions.

Once the model has been trained and validated, it was saved using *joblib* for persistence and reuse. The technical architecture incorporated the kNN recommendation system as a back-end service which is integrated into the wider RME platform. Upon completion of the analysis, the model generates property recommendations that are subsequently presented to the user.

The user collaborative filter system only incorporated click data. A click data point represents a user clicking on a property card. The model took a user as input and created a pattern for them. This pattern was represented as a vector with the number of properties in the database in it as its length and the number of clicks for each as the property's value. It took all of the other vectors for the other users in the database and compared the vectors to one another, looking for users with similar interactions. This was measured by finding the Cosine of the angles with values closest to zero between each vector. The formula used can be seen in equation (1) where A is the input vector, B is the comparison vector, n is the number of vectors to be compared, and i is the current index inside vectors A and B.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}}, \tag{1}$$

The ten most similar users to the input user were retrieved and a new data set was made. This data set contained all the combined clicks for each property between all the recommended users. The properties were sorted in descending order by click count, and the first ten were displayed to the user. These were the ten most interacted properties between all recommended users. Once the Cosine similarity model was deployed to a lambda function, the data was stored temporality in the front-end. When the lambda function is called, the user can observe the recommendations presented to them. During the development of the Cosine similarity model other iterations of the model were created. These iterations were hybrid models that combined geocoordinate data as well as property data. The curse of dimensionality caused the model to struggle to find similarities upon initial testing. The Cosine of the angles between all users had a value close to one, this indicates that no similarity was found. For this reason, the model that incorporated click data was chosen to be deployed and evaluated during testing.

IV. Evaluation

The recommendation system for both the kNN and User Collaborative Filtering models were evaluated using user feedback obtained from surveys. The recommendation systems gave recommendations based on content and user interactions. The recommendation engine's performance is based on user sentiment toward their recommendations. The surveys allowed for qualitative feedback that enabled the models to be adjusted based on user preferences. The evaluation of the recommendation system aims to determine:

- The suitability score for each feature in the recommended properties (based on user feedback).
- The overall suitability score of the recommended properties (based on user feedback).

Due to the interconnecting nature of the recommendation and tag systems, the evaluation of both was joined into a single questionnaire in conjunction with the scoring system. As such, the methodology and evaluation metrics are similar for the property tags and the recommendation system. As mentioned, property tags are designed to increase the usability of RME. They also confirm the recommended properties, for example, the user's favourite properties with 'secure' tags, and therefore secure properties should be recommended to the user. The two recommendation models were made available at different intervals with approximately half of the participants testing each model. Users did not know which recommendation system they were testing. The evaluation aimed to explore the tag contribution to the recommended properties, testing kNN and Cosine recommendation models.

Table 1 shows the results of Kendall's Tau which is used to measure the correlation coefficient between each of the suitability ratings of the kNN features and the overall suitability rating of the kNN model. The decision to measure the correlation between participant kNN feature suitability and participant kNN overall suitability scores was based on the actual values of the variables without assuming any specific underlying distribution. This approach focused on evaluating the similarity in the ordering of the data points.

Each feature defined a hypothesis as follows:

- Null hypothesis (H_0): Feature x does not impact the kNN overall suitability score.

 ISBN: 978-1-68558-187-9

- Alternative hypothesis (H_1): Feature x does impact the kNN overall suitability score.

A p-value of 0.05 was set for each test as this is a commonly used metric, where a p-value less than 0.05 is generally considered to be statistically significant and considered to be grounds for rejecting the null hypothesis.

TABLE 1. KENDALL'S TAU CORRELATION COEFFICIENT OF KNN FEATURES

Feature	P-value	Kendall's Tau	Correlation Strength	Failed to reject:
Available amenities	p = 0.191	r = 0.29	Weak correlation	H (0)
Nearby personal care	p = 0.069	r = 0.41	Moderate correlation	H (0)
Nearby banks	p = 0.049	r = 0.46	Moderation correlation	H (1)
Price	p = 0.007	r = 0.60	Strong correlation	H (1)
Nearby emergency services	p = 0.028	r = 0.50	Moderate correlation	H (1)
Nearby public transportation	p = 0.208	r = 0.29	Weak correlation	H (0)
Nearby leisure activities	p = 0.009	r = 0.59	Moderate correlation	H (1)
Nearby retail	p = 0.014	r = 0.56	Moderate correlation	H (1)
Nearby gyms	p = 0.060	r = 0.44	Moderate correlation	H (0)
Area safety	p = 0.210	r = 0.28	Weak correlation	H (0)
Number of bathrooms	p = 0.042	r = 0.46	Moderate correlation	H (1)
Number of bedrooms	p = 0.267	r = 0.25	Weak correlation	H (0)

Table 1 illustrates half of the input features were positively correlated with the overall recommended suitability score. Nearby banks, emergency services, leisure activities, retail, and bathroom count all appeared to be positively correlated with the overall suitability score, which was unexpected. Price appeared to have a strong correlation with the overall suitability score. The alternative hypothesis for each of these features failed to be rejected as they had p-values < 0.05. The relationship between the input features and the overall suitability score is not linear. The features with a positive correlation with the overall suitability score indicate there is a consistent but not constant relationship between the two variables.

When users were asked to rate their interest in these features, most results showed neutral and negative sentiment. Bathroom, retail, and price having a moderate–strong correlation was expected, as people expressed interest in these features when asked. It was expected that bedrooms, bathrooms, area safety, transportation, and available amenities would have a stronger positive correlation, with lower p-values due to the overall initial interest expressed by participants for these categories.

The only feature with a strong positive correlation with the overall suitability score was property price. The kNN model does not weight its parameters. This feature affects the result just as much as the other features inputted into the model do. It is possible for different property price ranges to be paired with similar values of the other popular categories that make up the majority input of the kNN's parameters. This could suggest the reason for the property price's strong correlation.

Before testing the user collaborative filtering model, it was presumed that user collaborative filtering would prosper when recommending properties to users with an interest in the property information. This information consists of price, property size, bathroom count, bedroom count, amenity, and tag information, all of which are displayed on the front of the property cards.

A click is intended to represent a genuine interest in a property. However, false signals of interest can occur if the information displayed on the property card highlights only certain 'key' points that attract the user's attention. For example, if a user is specifically searching for properties with a microwave, they might click on properties with an "amenities" tag, even if the property does not fully meet their other criteria. This can lead to misleading data about user preferences and interest levels.

Most users had a positive sentiment for the crime tags and nearby service scores. The crime tag is not based on weights. Changing the shape of the crime tag sigmoid function was the only way to modify the influence of the crime tag. The nearby service scores could be improved by changing the application of the weights with users specifying their preference using service weights.

V. CONCLUSION

The RME application was built to provide a better user experience for the end-user. A key component of this application was the recommendation system which included a combination of a Property Scoring System, Property Tag selection and filtering algorithms. The filtering algorithms evaluated included kNN recommendation system and User Collaborative Filtering using Cosine Similarity. These different approaches were evaluated using subject market experts. These were selected from various industry-related roles including property rental agents, estate agents, property owners and renters in the 20–35-year-old range. The outcomes from the initial testing demonstrated positive outcomes and feedback from the end-users with particular feedback pointing to the usefulness of the Tagging system and the inclusion of their preferences. As noted, the evaluation also revealed a relationship between the input features and the overall suitability score that was consistent but not constant. Future work will further develop the recommendation engines. This will involve expanding the dataset by incorporating data from different cities or geographic regions and increasing the number of subject matter experts to ensure a broader and more comprehensive analysis.

REFERENCES

[1] Zillow, Real Estate, Apartments, Mortgages & Home Values, www.zillow.com, date last accessed August 19, 2024.

[2] I. Borg, D. Hermann and W. Bilsky, "The perceived seriousness of crimes: inter-individual commonalities and differences." *Quality & Quantity*, vol. 57, pp. 765–784, 2023. https://doi.org/10.1007/s11135-022-01379-9

[3] A. Mucherino, P. J. Papajorgji and P. M. Pardalos, "*k*-Nearest Neighbor Classification." Data Mining in Agriculture. Springer Optimization and Its Applications, vol. 34, pp. 83-106, Springer, 2009. https://doi.org/10.1007/978-0-387-88615-2_4.

[4] B. Li, L. Han, "Distance Weighted Cosine Similarity Measure for Text Classification.", Intelligent Data Engineering and Automated Learning – IDEAL 2013, vol. 8206, pp. 611-618. https://doi.org/10.1007/978-3-642-41278-3_74

[5] Open Baltimore Datasets, Crime Data, https://data.baltimorecity.gov/search, date last accessed September 21, 2024.