CMPT 120 Project 3: Airbnb User Recommendation System

Motivation & Background

Our group chose to create a searching and sorting algorithm to expand upon our current knowledge of concepts introduced in the CMPT 120 course. The focus of this project was on sorting through a list of Airbnb listings and creating a recommendation system that could be customised according to a user's personal preferences. Note that although this project was created to search and sort through Airbnb listings, it can also be applied to other datasets.

As one of the most popular travel websites for booking accommodations around the world, Airbnb has a number of competitors (including Vrbo, Booking.com, and more) and must continuously improve its existing recommendation system to draw in more consumers and maintain interest from its current customers. In addition to broadening our knowledge on key concepts covered in class, the code developed for this project provides insight into how recommendations work, and thus how Airbnb's current system might be improved to maintain a competitive edge against similar platforms.

Research Questions

The 5 research questions our group has chosen to address for this project include the following:

1. What filters should be prioritised when recommending listings to users (i.e. if a listing doesn't meet all the user's criteria, should price be prioritised over type, or vice versa)?

2. How can we better recommend listings with unique characteristics that are not covered by general filters like location, type, quality, popularity, and price (i.e. accessibility)?

3. How can we implement a user feedback system, and apply the user's feedback for improved recommendations on future searches for listings?

4. How can we create an accessible and easy-to-use searching system for users to improve the recommendation system's usability?

5. How can we recommend listing types according to the user's search history (i.e. if the user always looks for apartments, listings of the same type will appear before others) so the filter listings are better-suited to the user's preferences?

Dataset

The program developed for this project used a data text file containing relevant data (listing ID, name, location, room type, price, and popularity) from Airbnb listings in Vancouver. This information was retrieved from Inside Airbnb, a project with a mission to provide "data and advocacy about Airbnb's impact on residential communities".

Question 1

What filters should be prioritised when recommending listings to users (i.e. if a listing doesn't meet all the user's criteria, should price be prioritised over type, or vice versa)?

Explanation

Filters are essential to any recommendation system, as they allow users to sort through options based on their personal preferences and priorities. To ensure the user's preferences were being addressed in the most effective way possible, one of the goals of our program was to prioritise filters based on their importance to the user.

Methodology

Our program accomplishes this by giving users the option to select how they want to search for their listing (i.e. by price, location, etc) and returning a list of the top 5 listings corresponding to their search. This ensures users are able to view their options, choose which filter is the most important to them, and receive options based on their preferences. To do this, we defined a function for finding listings, printed a list of searching options, took the users input (to choose an option), and used several if statements to perform the searching method chosen.

After a number of searches have been made, users also have the option to sort through the listings based on their search history. This was done by asking the user to choose one of the five recommended listings from their search, and storing information on the type of listing chosen (i.e. location, price, popularity, etc).

Result

This part of our program enabled the user to adjust their browsing experience to better-suit their personal preferences. If developed further, this part of the program could be adjusted to perform multiple searches simultaneously (i.e. search for listings with the best price and location rather than only the best price). Additional changes could also be made to include the user's budget/price range, amenities needed, and other needs; similar to the filters currently available on the Airbnb website.

Question 2

How can we better recommend listings with unique characteristics that are not covered by general filters like location, type, quality, popularity, and price (i.e. accessibility)?

Explanation

Although filters are an effective method for sorting through listings based on common criteria like price or location, they often do not encompass everything that listings have to offer. In some cases listings have unique characteristics or qualities that appeal to users, such as being accessible or clean.

To avoid the issue of limiting the scope of a user's search for listings, our program will aim to incorporate more unique qualities into the listing information to help users find listings based on more discoverable characteristics (for example, "quirky"). In doing so, we can create the framework for adding user-defined filters by giving users the option to assign "tags" to listings they have previously visited, making their browsing experience more interactive, special, and fun.

Methodology

Our program achieves this by giving users the option to provide feedback on a listing they have previously listed by assigning a tag to it. This was done by defining a function that takes the listing ID from a user, and then asks the user to choose a tag from a list of options that corresponds with the listing selected. This information is then stored in a separate document for tags, which will be displayed along with other information about the selected Airbnb listing.

Result

This gives users the ability to contribute to the database by providing their feedback, as well as learn more about the unique traits associated with their listing. Further development of this part of the program could be done by adding a function for searching listings by unique traits. This would enable users to broaden the scope of their search using feedback from other users and/or tags added by hosts, making it possible for users with unique criteria to find listings with traits not covered by the general filters (for example, pet owners may find filters like "fenced yard" useful when searching for pet friendly listings).

Question 3

How can we implement a user feedback system, and apply the user's feedback for improved recommendations on future searches for listings?

Explanation

User feedback systems allow users to provide their perspective and communicate their needs, as well as know that their opinions are valued. This feedback can be used to improve future recommendations, determine the most popular listings, and identify listings that are best-suited to what the user is looking for.

Methodology

Our program includes a feedback system that gives users the option to provide feedback for a listing. The program then takes a listing ID from the user and then asks the user to leave a "review" by choosing a tag from a list of options corresponding to the listing selected. This information is then stored in a separate document containing all tags associated with listings. The next time a user searches for the listing in question, the tag will be included along with other relevant information on the listing, such as location or type.

Result

This enables users to contribute to the recommendation system by providing additional information about listings. The feedback system could be further developed by giving users the option to (1) search for listings by tag or to (2) create their own tags instead of asking them to choose from a list of options. Tags assigned to listings could also be sorted from "most frequently assigned" to "least frequently assigned" to ensure that the most relevant and/or agreed upon feedback is displayed.

One of the feedback Airbnb frequently receives is that the filters available are too limiting and do not always encompass all the needs of users. As a result, users must often spend a great deal of time searching through each listing's information individually or email hosts to inquire further about the amenities/features of their prospective listing. By making all of these adjustments described above, users can create tags suited to their needs and the scope of a user's search could be broadened to include specific criteria not usually encompassed by general filters or "default" tags, making the search for listings more time efficient.

Question 4

How can we create an accessible and easy-to-use searching system for users to improve the recommendation system's usability?

Explanation

Usability is an essential quality to any recommendation system. An organised, understandable, and user-friendly searching system makes it easy for users to find and provide feedback on listings. Our system will implement a menu where the user has a choice to select a search option, and all the user will have to do is enter a number that corresponds to the option.

Methodology

Our program employs a "menu" system that presents the user with a numbered list of options at each stage of their browsing experience, along with clear instructions on what to do. This is done using print() and input() functions to print instructions/options and/or pose questions to the user. Once presented with a set of options, the user can simply enter the number corresponding with the option they wish to choose. If the user's input matches the number of one of the options available, an if statement corresponding with the option in question will perform the function chosen. If the input does not match any of the listed options, the system will inform the user their input is invalid and request they re-enter the number corresponding with their option. The "main menu", which allows users to choose from providing feedback, finding a listing, or quitting the program is contained in a while loop, allowing users to perform multiple searches until they choose to "quit" their browsing experience.

In an effort to provide a personalised experience for users, our program also asks the user for their name, assigns it to a variable, and uses it to address the user in future interactions with the system.

Result

This made it easy for users to use the various functions available via the program without any difficulty. To make the browsing experience more personalised, additional efforts could be made to tailor messages/instructions to the user and/or a "login" system could be implemented to store specific user information. The implementation of a "login" system would also lay the groundwork for further customisation.

Question 5

How can we recommend listing types according to the user's search history (i.e. if the user always looks for apartments, listings of the same type will appear before others) so the filter listings are better-suited to the user's preferences?

Explanation

Storing a user's search history makes it possible to provide recommendations better-suited to a user's interests and needs. Adding a searching function based on search history makes it easier for users to access listings similar to ones they have previously indicated an interest for, simplifying the searching process by recommending listings that may not appear in more specific searches (such as searching for listing by ID or by popularity).

Methodology

Instead of a visit based search history, we implemented a tag system in which the user is able to apply tags to different listings. By using the tag system, the user is able to add certain feedback tags to listings, and search through them in the future to have a more user-oriented search system. Therefore, users are able to label listings as "Affordable" or "Quirky" and then in the future look through the listings with feedback applied to them.

Result

This makes it possible for users to access and receive recommendations based on their personal interests/preferences, allowing for a more personalised browsing experience. One of our initial plans for our program was to incorporate an option for recommending listings based on a user's search history. Unfortunately, we were unable to accomplish this within the timeline for this project, and opted for using tags as an alternative method. If we had the time to do so, recommendations based on search history could have been done by recording the number of times a user selects any particular listing from their top 5 recommendations in a separate file. After a number of searches are made, in theory the recommendation system would be able to rank the user's most visited listings, and in doing so identify the most frequently viewed filter types as well as allow users to access previously viewed listings. Further development of this aspect of the code could be done by creating a tailored list of recommendations based on previously viewed listings. This would likely be accomplished by finding the most frequent filter types (i.e. price, location, room type, etc.) of viewed listings, searching through the available listings, and returning the top 5 listings with the most matching qualities.