TotallyMakeScents

Juntong Su and Yevgeniya (Jonah) Tarasova

Introduction

We developed a perfume recommendation system that returns a list of perfumes from user preferences according to the tastes of similar users and the notes of perfumes. Choosing a perfume on the internet is almost always difficult, especially for a fresh customer without access to samples. Flipping through the perfume encyclopedia, drowning in the sea of top, middle, and base notes, deciphering the ingredients and accords, emerging in the comments and debates on fragrance forums, visualizing the descriptions from KOLs – we got rid of all those hassles with a straightforward app. Just enter the scents and the sentiments, we make scents for the users within a few seconds.

Datasets

Our recommendation system combines content-based and collaborative filtering with multiple large datasets. For collaborative filtering we used The Perfume Co-preference Network dataset (user preference dataset), which is one of the main results from this research (K. Kalashi et al, 2024). The dataset contains 36,434 reviews and sentiments on the scents of perfumes from 7,387 unique users collected from the Persian retail platform Atrafshan. The dataset was cleaned by dropping NaN rows and unused columns. For content-based filtering we experimented with the (1) Fragrantica dataset, including columns of perfume name, top notes, middle notes, and base notes for more than 24,000 perfumes; (2) Aromo dataset with similar columns for more than 78,400 perfumes. In addition to the columns listed above, we also used url and rating counts for data cleaning. For consistency, we selected the overlapping perfumes between content-based and user-based datasets. We discarded perfumes with the same names unless one of them has an overwhelmingly large number of ratings. Despite the different naming conventions, we ended up choosing Fragrantica, since it has more overlaps (804). The initial embeddings for content-based filtering were created by one-hot encoding, resulting in a sparse representation of high dimensional perfume-note matrices.

Models

We decided to use a combination of three models: A user based model (aka collaborative filtering) that recommended perfumes based on how similar a user's preferences were to those of other users in our dataset; a content-based model that recommended perfumes with similar notes to perfumes the user enjoyed in the past; and a model that recommended the most popular perfumes among users deemed "relevant" enough. In order to measure similarity between users and similarity between notes, we used cosine similarity, while "relevant" users were simply those who had one liked perfume in common with the target user. Each model takes in a list of liked and disliked perfumes and outputs a recommendation list.

During testing we observed that, for users on whom we had information on fewer than twenty total likes and dislikes, the popularity model best predicted their preferences. However, the other two models had greater improvements in performance than the popularity model when we had more user data. It should be noted that the user based model inherently has a popularity bias, while the content-based model does not. As one goal for our recommender is to help users discover perfumes they may have otherwise overlooked, our final model pulls from all three of the aforementioned models.

We first allow the user to decide the split they prefer between the user and content-based models. As the user based model has a popularity bias, and performs worse than the popularity model for users who have given us fewer than twenty total likes and dislikes, the user based model is replaced with the popularity model when the user gives us fewer than twenty total likes and dislikes.

Testing

For testing we used the user preferences dataset to generate mock user profiles. To do this, we first reserved 80% of the users for our training data while leaving 20% of the users alone to be the final testing data. Within the training data we looked for users who had reviewed at least ten perfumes, and removed 30% of those users from the set to be our validation set. We split the reviews of each user in the validation set in two and used one half to run our models, while reserving the other half to evaluate the models' performance. After running our tests on a few splits of the training data, we determined our final model and ran the same tests on our testing data.

To evaluate our models we used two methods: The first was a hit counter. We had each of our models generate a list of 20 recommendations, generated three lists of 20 random perfumes, along with a list of the top twenty perfumes overall. Then we computed the overlap between each list and the half of the data reserved for evaluation, before comparing the results.

The second evaluation method consisted of using the perfume ratings generated by the user, popularity, and content-based models, as well a randomly generated list of of ratings (that had the same positive to negative review split as the training data) then using cosine similarity to compare the ratings given by the respective models to the true ratings in the half of the data reserved for evaluation. Once more we compared the results of each model.

Future

This project is the very beginning of a long term project. In the future, the recommender should be able to generate lists of perfumes based on user queries, which could be short descriptions, moods, or even movie quotes. What's it smell like in the rain, at the end of a hiking trail full of blossoms? What fragrance would a wizard wear in a magical world? Looking for a bittersweet scent for a farewell party. Fragrance finders like Parfumo or Fragrantica can only suggest a similar perfume to the users' already existing preferences. We would like to accomplish more. Please stay tuned while we MakeScents!