Fragrance Recommender Web Application

by

Michael Tutterrow

Dr. Piotr Szczurek, Advisor

Masters project submitted in partial fulfillment of the

requirements for the Master of Science in Data Science degree

in the College of Arts and Sciences of

Lewis University

***Abstract*—This report documents the creation of a fragrance recommender web application. The data supporting the recommendations is derived from user reviews and ratings from fragrance enthusiasts websites. Ratings from Basenotes.com and notes rankings from Fragrantica.com are used to perform a recommender based on knowledge-based, content-based, and collaborative-filtering-based algorithms. Methods of calculating recommendations include ratings' means, weighted means, k-nearest neighbors (kNN), and singular-value decomposition (SVD). The data is analyzed using the Python programming language, with analysis modules including numpy, pandas, scikit-learn, and surprise. These algorithms and datasets are integrated into a web application, where a user can enter fragrance preferences and receive new recommendations in real-time.**

***Index Terms*—classification algorithms, computer aided analysis, crowdsourcing, data acquisition, data engineering, data mining, data processing, information filtering, machine learning algorithms, recommender systems, scientific computing, user-generated content**

I. Introduction

The last few years have seen the number of fragrances available to consumers grow tremendously. The Michael Edwards Fragrances of the World database added 16,664 new entries between 2009 and 2017 [1]. Fragrantica shows more than 2,000 new fragrances have been released thus far in 2018 alone [2]. During this time of growth, websites dedicated to fragrances, like Basenotes [3] and Fragrantica [4], have become popular with fragrance enthusiasts because they provide forums for discussion and allow users to post their own reviews of fragrances. As these websites have grown in popularity over the past 10-20 years, the amount of data collected has also grown. Fragrance lovers now have a wealth of other users' reviews to assist them in making decisions.

Because interpretation of fragrances is highly subjective, determining which reviews are reliable can be a challenge. As a result, buying a fragrance based solely on reviews can be disappointing. A recommendation system built with a knowledge-based, content-based, and collaborative filtering methods, however, can take the guesswork out of this process.

This report documents the process of creating this application. The Discussion of Related Works looks at other fragrance recommenders, as well as work done in recommender systems in general. The Project Description explains in more detail the technical steps involved in data acquisition and implementation of the recommender. Methodology explains the methods used in the various recommender systems implemented. Ethical Considerations includes a discussion of the potential ethical issues with these types of applications. The web application is reviewed in detail in Experiments and Implementations. The results of the model evaluations are discussed in Results Evaluation.

II. Discussion of Related Work

As noted in [18], recommendation systems started to become popular in the mid-nineties, as systems such as *GroupLens* [27] were developed. Since then, these systems have been studied extensively in the context of content-based, knowledge-based, and collaborative filtering methods. An excellent overview of content-based of systems can be found in [28], while an overview of knowledge-based recommendation systems can be found [29]. Collaborative filtering methods are typically one of two methods: neighborhood-based and model-based. Neighborhood-based methods, also known as memory-based, typically use similarity metrics to group either users or items with similar characteristics. A comprehensive study of neighborhood-based methods is found in [30]. Model-based methods, on the other hand, use machine learning methods and are closely related to the problem of classification. [18] provides an overview of these methods.

In her article, "How I Built a Recommendation System for Fragrance", Kelly Peng explains her process for creating a similar recommendation system. She scraped the largest fragrance forum in China to compile data for perfume features and user ratings. Then she used content-based models, item-item similarity collaborative filtering models, and matrix factorization models [5]. Unfortunately, this tool is no longer available online for experimentation.

Other fragrance recommenders include [7]–[10]. One tool, [9], gives a quiz of images, shapes, and colors to determine a user's scent personality. Another tool, [7], recommends three other fragrances in the same general olfactory category as a user's input fragrance. Some of these tools are associated with online sellers, which may have motives beyond making the best overall recommendation, and would presumably be limited by available inventory. These types of recommenders raise ethical concerns, where a user's own interests are not aligned with the goals of the recommender system. In any case, the methods and accuracy of these tools are not documented.

III. Project Description

This project implements a web application which will give recommendations based on a user's fragrance preferences. Data from popular fragrance review website(s) are used to populate a dataset of fragrance characteristics and reviews. Recommendations are made based on knowledge-based, content-based, and collaborative filtering models.

A. Dataset

Reviews and fragrance data was downloaded from Basenotes [3] during several days in July 2018, with reviews added since July downloaded in November 2018. Fragrance data from Fragrantica [4] was downloaded during November 2018. Python scripts used for data acquisition can be found in [11].

TABLE I
DATASET SCHEMA

Attributes	Type	Basenotes Instances	Fragrantica Instances
Reviews	Categorical	147,810	N/A
Fragrances	Categorical	16,528	50,674
Fragrance Notes	Categorical	9,661	873

Basenotes has a system of reviews, whereby a user may write a review and give their rating, either a thumbs up, thumbs down, or thumbs sideways for a positive, negative, or neutral rating, respectively. At the time of the data acquisition, there were 147,810 reviews, for 16,528 fragrances. This feature is used for user-user collaborative-filtering recommendations.

Fragrantica also compiles user reviews; however, these reviews do not have ratings applied to them, only a user's written review. Fragrantica does have some attractive features of its own: several fields are available for users to apply their own feedback regarding fragrances. A user can rank the listed olfactory notes of a fragrance, according to the perceived strength. This feature is used to calculate similarity between different fragrances, and create knowledge-based and content-based recommendations.

It appears that Basenotes records fragrance notes verbatim from the fragrance house, while Fragrantica tries to group unique notes into more broad categories. For example, Basenotes lists 'Vermont red maple wood and Wyoming cottonwood' [12] while Fragrantica lists 'Red Maple' [13]. The Fragrantica approach groups similar notes into a single note in their directory, essentially performing a clustering analysis. This approach is, therefore, much more practical for content-based recommendations.

More than half the total time spent on this project involved scraping the websites, and parsing, manipulating, and cleaning the resulting data. Python scripts used for data manipulation and cleaning can be found in [11].

One of the initial primary goals with this project was to utilize datasets from two different websites, since it has been shown that "more data usually beats better algorithms" [14]. The idea was to use the best features from both websites to create a stronger recommendation system. Fragrantica data could be used for classification and knowledge/content-based recommendations, while Basenotes could be used for collaborative filtering.

In order to link individual fragrances from Basenotes to Fragrantica, brands in the Basenotes directory were first linked to the brands in the Fragrantica directory. Next, fragrances within specific Basenotes brands were linked to the fragrance names/IDs within the corresponding Fragrantica brand. A script was used to analyze the edit distance (Levenshtein distance) between specific fragrances, with the minimum edit distance used to identify the most similar name. Edit distance is defined as the smallest number of insertions and deletions of single characters that will convert string x to string y [17].

IV. Methodology

This application provides recommendations based on three methods: knowledge-based, content-based, and collaborative filtering. Methods for creating content-based recommendations and collaborative filtering recommendations are documented in [17]–[19]. [19] includes an example of a hybrid recommender with content and collaborative filters. Portions of the code from [19] are used to perform the collaborative filtering model evaluation.

A. Content-Based Recommendation

Content-based recommender systems are designed to match users to new items based on the content of the items the user already prefers. These systems focus on the ratings the user has given to preferred items, and the characteristics of those items. These systems do not rely on the ratings of other users, and do not rely on the user having any intrinsic knowledge of their preferred items, other than the fact that they prefer them. In general, accurately assessing item characteristics is critical to the accuracy of such a system.

1) The Challenges of Fragrance Classification

Achieving the goals of the content-based fragrance recommender depends largely on the ability to accurately classify fragrances. A summary of the challenges of fragrance classification is found in [20]. It is noted that it is virtually impossible to achieve a fragrance classification system with universal acceptance. The way odors are perceived between different people varies, and is difficult to explain, especially for someone untrained in this task. Fragrance companies use experts who are trained for years, and are versed in multiple descriptors for classification. Even then, classifications vary among different companies and their experts.

Consider the classification of one fragrance, Eau de Cartier. [20] summarizes the olfactive family classifications of this fragrance according to several different companies and authors:

TABLE II
EAU DE CARTIER OLFACTORY CLASSIFICATIONS

Organization	**Olfactory Group**	**Gender**
Osmoz (Firmenich)	Citrus-Aromatic	Unisex
ScentDirect	Floral-Fresh	Feminine
Haarmann & Reimer	Chypre-Fresh	Feminine
The Fragrance Foundation	Citrus-Rich	Unisex
Turin & Sanchez	Violet leaf-Woody-Citrus	N/A
ISIPCA	Floral-Woody-Citrus	Unisex

Perfume Radar	Oriental-Citrus-Woody	N/A

While Fragrantica classifies Eau de Cartier the same as Osmoz, as a citrus-aromatic [21], it appears that other fragrance classifications are not consistent with any single source. However, Fragrantica does provide a potentially useful feature that allows users to rate the presence of the different notes listed by each fragrance. This feature shows that Fragrantica users rate the notes of Eau de Cartier in the following order of importance.

TABLE III
EAU DE CARTIER NOTE RATINGS

Note	Eau de Cartier Profile	Normalized Weight
Violet Leaf	349	0.15
Yuzu	324	0.14
Violet	308	0.13
Lavender	247	0.11
Bergamot	243	0.11
Cedar	235	0.10
Coriander	219	0.10
Musk	181	0.08
White Amber	102	0.04
Patchouli	95	0.04
Total	2303	1.00

2) Fragrance Profile Vectors

This application leverages the ratings provided by Fragrantica users to create a profile vector, or a set of features, for each of the fragrances in the data set is generated. The features used to define this vector include the most common notes of the 873 fragrance notes listed on the Fragrantica website. As noted in [18], when selecting features for a vector space representation, a size cut-off should be used, otherwise the presence of so-called noisy words results in overfitting. Experimental results of [31] indicated that the number of extracted words should be between 50 and 300. This application uses the 100 most frequent notes in the Fragrantica dataset, although larger numbers have been used without adverse effects, and may be used in future implementations. The top 100 most frequent notes are used in the content-based analysis, shown in TABLE IV below.

The content of a user's preferred fragrance(s) is, therefore, defined based on the pre-calculated fragrance vectors. Next, the user-preferred content is compared to the other fragrance vectors in the dataset, and the most similar fragrances are recommended. The similarity is measured based on the cosine similarity metric, which is also used in the knowledge-based recommender, described in the next section.

TABLE IV
THE TOP 100 MOST FREQUENT NOTES

Rank	Note	Frequency	Rank	Note	Frequency
1	Musk	17756	51	Pear	1576
2	Jasmine	12289	52	Heliotrope	1535
3	Rose	11806	53	Coriander	1531
4	Vanilla	11509	54	Cloves	1512
5	Sandalwood	11505	55	Saffron	1511
6	Bergamot	11102	56	Orchid	1470
7	Amber	10889	57	Lily	1445
8	Patchouli	9395	58	Tea	1403
9	Cedar	9133	59	Plum	1396
10	Mandarin-Orange	5662	60	Orris-Root	1367
11	Vetiver	5630	61	Carnation	1329
12	Lemon	5389	62	Lime	1302
13	Woody-Notes	4705	63	Honey	1277
14	Oakmoss	4169	64	Fruity-Notes	1254
15	Orange-Blossom	4010	65	Violet-Leaf	1226
16	Tonka-Bean	3949	66	Tobacco	1207
17	Lavender	3741	67	Pineapple	1203
18	Lily-of-the-Valley	3696	68	Ambergris	1177
19	Violet	3600	69	Aldehydes	1124
20	Grapefruit	3494	70	Basil	1112
21	Iris	3438	71	Petitgrain	1102
22	Ylang-Ylang	3218	72	Guaiac-Wood	1086
23	Peach	2966	73	Sea-Water	1078
24	Pepper	2890	74	Galbanum	1072
25	Cardamom	2865	75	Tangerine	1035
26	Orange	2772	76	Sage	1023
27	Leather	2743	77	Myrrh	1015
28	Green-Notes	2627	78	Coconut	1013
29	Freesia	2583	79	Currant-leaf-and-bud	994
30	Geranium	2535	80	Rosemary	960
31	Agarwood-Oud-	2319	81	Lotus	936
32	Pink-Pepper	2295	82	Caramel	914
33	Benzoin	2274	83	Osmanthus	907
34	Cinnamon	2262	84	Mimosa	890
35	Citruses	2213	85	Olibanum-Frankincense-	878
36	Incense	2182	86	Almond	817
37	Peony	2174	87	Cashmir-wood	772
38	Neroli	2165	88	Artemisia	762
39	Black-Currant	2140	89	Water	760
40	Tuberose	2108	90	Melon	731
41	Apple	2011	91	Apricot	730
42	Labdanum	2009	92	Hyacinth	728
43	Nutmeg	1996	93	Cyclamen	721
44	Floral-Notes	1959	94	Cypress	708
45	Ginger	1883	95	Litchi	698
46	Raspberry	1770	96	Honeysuckle	697
47	Spicy-Notes	1755	97	Clary-Sage	680
48	Mint	1672	98	Red-Berries	672
49	Magnolia	1648	99	Lilac	640
50	Gardenia	1589	100	Bitter-Orange	638

B. *Knowledge-Based Recommendation*

The knowledge-based recommendation is useful in the cold-start condition, where a user has given limited feedback about preferences. It can also be effective in cases where a knowledgeable user is interested in specific search criteria.

The knowledge-based recommendation works by having a user enter in preferred fragrance notes. A user preference vector, $\boldsymbol{p}$, is a vector of values corresponding to the user's preferred notes. The fragrance note utility matrix, $\boldsymbol{U}$, is an m x n matrix, with rows m corresponding to vectors of fragrance notes and columns n corresponding to different fragrance notes.

The cosine similarity between two vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ is calculated as the cosine of the angle between the two vectors:

$$\cos\sigma_1 = \frac{\boldsymbol{A} \cdot \boldsymbol{B}}{\|\boldsymbol{A}\|\|\boldsymbol{B}\|}$$

The system works by computing the cosine similarity of the user vector to each of the fragrance vectors, with the largest similarity value corresponding to the most similar fragrance to the user's preferences. The implementation of knowledge- and content-based portions employ the cosine_similarity utility from the *metrics.pairwise* submodule of the *scikit-learn* Python library [26].

A simple example is shown below. Consider a utility matrix of three fragrance vectors, each comprised of three notes. The user preference vector is also included, indicating a positive preference for two of the three notes.

TABLE V
FRAGRANCE VECTOR EXAMPLES

Fragrance Vectors	Note 1	Note 2	Note 3
User Preference	0.50	0.00	0.50
Fragrance 1	0.45	0.10	0.45
Fragrance 2	0.33	0.33	0.33
Fragrance 3	0.00	0.50	0.50

Intuitively, it is obvious that the user profile is most similar to the vector of fragrance 1. The cosine similarity calculation, (1), between the user preference vector, **p**, and the fragrance vector $\mathbf{f_1}$, shows the similarity of the two vectors:

$$\cos\sigma_{\mathbf{pf_1}} = \frac{0.50*0.45+0.00*0.10+0.50*0.45}{\sqrt{0.50^2+0.00^2+0.50^2}*\sqrt{0.45^2+0.10^2+0.45^2}} = 0.99 \tag{1}$$

TABLE VI shows the complete list of cosine similarities for the user preference vector, further confirming that fragrance 1 is the most similar of the three fragrance vectors considered.

TABLE VI
COSINE SIMILARITY EXAMPLE

Notes Vector	User Preference Similarity
User Preference	1.00
Fragrance 1	0.99
Fragrance 2	0.57
Fragrance 3	0.50

C. User-User Collaborative Filtering

Once a user has entered enough preferred fragrances, and given feedback about the knowledge-based and content-based fragrance recommendations, the application employs a collaborative filtering approach to find users with similar preferences, and make recommendations based on their preferences. This approach typically requires a utility matrix where each user is a row, and each fragrance is a column. Each user's rating for a fragrance is a number in this matrix.

1) The Utility Matrix for Collaborative Filtering

The reviews of Basenotes [3] use a simple method for rating a fragrance. Reviewers choose thumbs up, thumbs down, or thumbs sideways to rate their approval, disapproval, or neutrality in regards to the fragrance, as seen in the Fig. 1 below.

Fig. 1. Typical Basenotes user reviews [22].

The method of user-collaborative filtering may be idealized by imagining a ratings utility matrix, where ratings of a given user are listed in a row, with each column representing a different fragrance in the dataset. Each rating is entered in the appropriate row/column as 3, 2, and 1, based on a user's preferences of like, neutral, and dislike. Cells with no entry represent fragrances that have not been reviewed. In this manner, each reviewer in the data set has a profile vector of reviews with which to be compared to a user of the application. The ratings utility matrix is similar to the fragrance vector utility matrix, as seen in TABLE VII below. Considering User 3 as a user of this hypothetical application, user-user collaborative filtering would recommend Fragrance 3, based on similarity with User 1.

TABLE VII
RATINGS UTILITY MATRIX EXAMPLE

Notes	Fragrance 1	Fragrance 2	Fragrance 3	Fragrance 4
User 1	3	1	3	2
User 2	3	-	2	-

User 3	3	1	-	-

Based on a comparison of several methods of collaborative filtering, summarized in the Results Evaluation section, this project uses the SVD method, or Singular Value Decomposition, to derive recommendations. The Surprise Python package is used to implement the SVD algorithm [25].

D. *Ethical Considerations*

Recommender systems by necessity make use of a person's individual preferences and, at times, personally identifiable information. As a result, there can be several ethical challenges when implementing a recommender system. As noted by [24], the data collection of recommender systems is often performed with a lack of transparency and informed consent. Furthermore, user profiles built for serving personalized recommendations can also potentially be used for malicious purposes, such as phishing or social engineering. [24] also notes that numerous examples exist of supposedly anonymized datasets that, nevertheless, resulted in user re-identification.

While these types of scenarios can occur in many systems, this recommender system only makes use of information publicly available on other websites and does not have access to any other personally identifiable information. The greatest risk to a user of this application is probably the potential to spend unnecessary money based on a recommendation that turns out to be inaccurate. Users should be advised of this risk and to proceed accordingly.

V. Experiments and Implementations

A. Content-Based Implementation

A common problem with recommender systems is the cold-start scenario, where little to no information about a user is available with which to make a recommendation. However, a person seeking a fragrance recommendation will typically know of at least one fragrance she prefers. This scenario is handled best by the content-based implementation, where the recommendation system can evaluate the content of the user's preference, and return similar items. In this simple example of the web application, we consider a user with one known preference, Eau de Cartier. The closest matching fragrances are returned: *Cartier: Eau de Cartier Concentrée, Serge Lutens: Bois de Violette, and Dsquared2: He Wood.* The user adds the first two to her preferred fragrances, and adds the third one to her disliked fragrances.

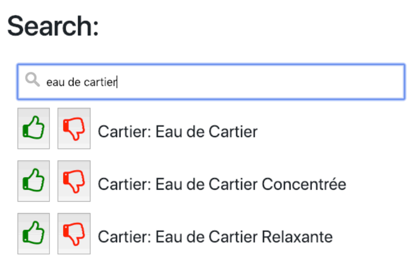
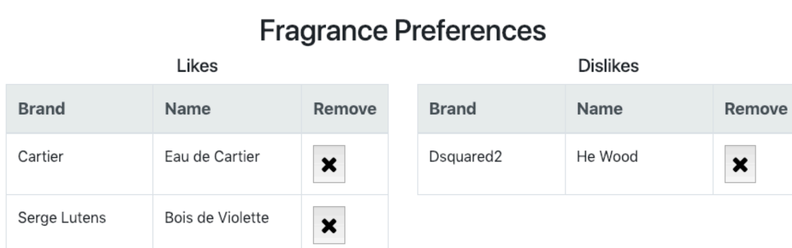


Fig. 2. Selecting Preferred Items

Fragrance	Similarity	Basenotes	Fragrantica
Cartier: Eau de Cartier Concentrée	Eau de Cartier		F
Serge Lutens: Bois de Violette	Eau de Cartier		F
Dsquared2: He Wood	Eau de Cartier		F

Fig. 3. Content-Based Recommendations



Brand	Name	Remove
Cartier	Eau de Cartier	✖
Serge Lutens	Bois de Violette	✖

Brand	Name	Remove
Dsquared2	He Wood	✖

Fig. 4. Adding More Preferences

B. Knowledge-Based Implementation

As the user becomes familiar with the fragrances, she notes a preference for violet, lavender, and cedar. These notes are added to the user's preferred notes, and knowledge-based recommendations are returned. The user continues to update her preferences with the new recommendations.

Note Preferences

Likes

Name	Remove
Violet	✖
Lavender	✖
Cedar	✖

Dislikes

Name	Remove

Recommendations

Fragrance	Similarity	Basenotes	Fragrantica
Christian Dior: Eau Noire Cologne	note preferences		
CB I Hate Perfume: M6 Do Not Ask Me Why	note preferences		
Anat Fritz: Classical	note preferences		

Fig. 5. Note Preferences and Knowledge-Based Recommendations

C. Collaborative-Filtering Implementation

The application now has six fragrance preferences from which to build a user-user collaborative filtering model.

Fragrance Preferences

Likes

Brand	Name	Remove
Cartier	Eau de Cartier	✖
Serge Lutens	Bois de Violette	✖
Christian Dior	Eau Noire Cologne	✖
Anat Fritz	Classical	✖

Dislikes

Brand	Name	Remove
Dsquared2	He Wood	✖
CB I Hate Perfume	M6 Do Not Ask Me Why	✖

Recommendations

Collaborative Filtering			
Christian Dior: Diorling	Chypre		
Krizia: Teatro Alla Scala	Undefined		
Bogner: Deep Forest	Oriental Woody		

Fig. 6. Fragrance Preferences and Collaborative Filtering Recommendations

VI. Results Evaluation

A. Knowledge- and Content-Based Models

The accuracy of the knowledge- and content-based recommendations are not evaluated. Under the assumption that the Fragrantica notes ratings represent a typical user of this application, we can assume that the cosine similarity metrics will provide accurate recommendations, based on a user's inputs.

B. User-User Collaborative Filtering Models

Several collaborative-filtering methods were created and evaluated. These methods are based on code examples found in [19]. The ratings dataset is divided into train/test sets based on a 75/25 split. Each of the filters is trained on the training dataset, and then the root mean squared error is calculated for the testing dataset.

The methods tested include three memory-based filters (Baseline, Mean Rating, and Weighted-Mean Rating), and two machine learning model-based filters (K-Nearest Neighbors and Singular Value Decomposition).

The Baseline model is created by assuming a value of 2 is returned for all user IDs and fragrances in the test dataset, while the Mean model is a simple collaborative filtering model, in which the predicted rating is the mean rating for the fragrance by all the users who have rated it. The Weighted-Mean model uses a weight coefficient to give more preference to users whose ratings are similar to the user being evaluated than to users whose ratings are dissimilar. Unfortunately, this approach produced a prediction matrix that contained several NaN ("not a number") values. This result is believed to be caused by the sparsity of the ratings matrix for many of the fragrances. While these NaN values in the prediction matrix can be converted to the baseline value of 2.0, the resulting RMSE is larger than the mean model.

The K-Nearest Neighbors (kNN) model is a common clustering method to group users into clusters and use only those common users when making a prediction for one of the users in that group. The implementation for this project uses a fivefold cross-validation technique and an algorithm derived from a basic nearest neighbors approach.

The final model tested, Singular Value Decomposition (SVD), is a technique developed by Simon Funk for the Netflix Problem [23]. Dimensionality-reduction techniques are used to convert an $m \times n$ sparse matrix into an $m \times d$ dense matrix, where $d << n$.

The results of the various collaborative-filtering models are summarized in TABLE VIII below. As seen in the table, the SVD model outperforms the other models. As a result, this model is used for the collaborative-filtering implementation in the web application.

TABLE VIII
COLLABORATIVE FILTERING MODEL RESULTS

Model	**Method**	**RMSE**
Baseline	all predictions = 2	0.887
Mean	prediction = average of all ratings	0.74
Weighted Mean	prediction = average with more weight given to users who have similar reviews; errors in calculation, possibly due to sparsity	0.756
kNN	k-Nearest Neighbor	0.743
SVD	Singular-Value Decomposition	0.698

VII. Conclusion

This project has demonstrated the use of multiple methods for the recommendation of fragrances, including knowledge-based, content-based, and collaborative filtering methods. Data from two prominent fragrance enthusiast websites is downloaded and then combined, in order to leverage the best features from each dataset. The combined dataset is analyzed, using the cosine similarity metric to compare fragrance note vectors with user-input note and fragrance vectors, to provide content- and knowledge-based recommendations. The SVD algorithm is applied to derive user-user collaborative filtering recommendations. All of the recommendations are delivered in real-time, through a web application, as a user enters their preferences.

VIII. References

[1] L. Turin and T. Sanchez, *Perfumes The Guide 2018*, First. Tallinn, Eesti: Perfüümista ÖÜ, 2018.

[2] "Perfumes Search." [Online]. Available: https://www.fragrantica.com/search/?q=&hPP=30&idx=fragrantica_perfumes&p=0&nR%5Bgodina%5D%5B%3C%3D%5D%5B0%5D=2018&nR%5Bgodina%5D%5B%3E%3D%5D%5B0%5D=2018&is_v=1. [Accessed: 27-Oct-2018].

[3] "Independent online guide to perfume — Basenotes.Net." [Online]. Available: http://www.basenotes.net/. [Accessed: 27-Oct-2018].

[4] "Perfumes and Colognes Magazine, Perfume Reviews and Online Community—Fragrantica.Com." [Online]. Available: https://www.fragrantica.com/. [Accessed: 27-Oct-2018].

[5] "GitHub - kellypeng/scentmate_rec: Built a fragrance recommender using multiple machine learning algorithms to help people find their perfect match. The final deliverable was a webapp." [Online]. Available: https://github.com/kellypeng/scentmate_rec. [Accessed: 26-Oct-2018].

[6] "How I Built a Recommendation System for Fragrance – Towards Data Science." [Online]. Available: https://towardsdatascience.com/building-a-recommendation-system-for-fragrance-5b00de3829da. [Accessed: 26-Oct-2018].

[7] "Fragrance Finder, perfume finder - Fragrances of the World." [Online]. Available: http://www.fragrancesoftheworld.com/FragranceFinder#txtSearch. [Accessed: 26-Oct-2018].

[8] "Fragrance Recommender." [Online]. Available: https://fragfinder.publicismedia.ai/. [Accessed: 26-Oct-2018].

[9] "Quiz – Pinrose." [Online]. Available: https://pinrose.com/pages/quiz. [Accessed: 26-Oct-2018].

[10] ScentSee.com, "ScentSee - Discover the fragrance that reveals you," *ScentSee*. [Online]. Available: http://en.scentsee.com. [Accessed: 26-Oct-2018].

[11] M. Tutterow, "This repository contains supporting data and analysis for the CPSC-59000 Data Science Project for Computer Scientists, in partial fulfillment of the requirements for the Master of Science in Data .." 10-Dec-2018. [Online]. Available: https://github.com/tallrice/CPSC-590_DSProj. [Accessed: 10-Dec-2018].

[12] "Tommy 10 by Tommy Hilfiger (2006) — Basenotes.Net." [Online]. Available: http://www.basenotes.net/ID26125696.html. [Accessed: 06-Nov-2018].

[13] "Tommy 10 Tommy Hilfiger cologne - a fragrance for men 2006," *www.fragrantica.com*. [Online]. Available: https://www.fragrantica.com/perfume/Tommy-Hilfiger/Tommy-10-1393.html. [Accessed: 06-Nov-2018].

[14] "More data usually beats better algorithms," *Datawocky*. [Online]. Available: https://anand.typepad.com/datawocky/2008/03/more-data-usual.html. [Accessed: 10-Dec-2018].

[15] M. Tutterow, "This repository contains supporting data and analysis for the CPSC-59000 Data Science Project for Computer Scientists, in partial fulfillment of the requirements for the Master of Science in Data .." 10-Dec-2018. [Online]. Available: https://github.com/tallrice/CPSC-590_DSProj/blob/master/Basenotes/python/csv/basenotes.csv [Accessed: 10-Dec-2018].

[16] M. Tutterow, "This repository contains supporting data and analysis for the CPSC-59000 Data Science Project for Computer Scientists, in partial fulfillment of the requirements for the Master of Science in Data .." 10-Dec-2018. [Online]. Available: https://github.com/tallrice/CPSC-590_DSProj/blob/master/Basenotes/python/csv/fragrantica.csv [Accessed: 10-Dec-2018].

[17] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive datasets / Jure Leskovec, Standford University, Anand Rajaraman, Milliways Labs, Jeffrey David Ullman, Standford University*, Second edition. Cambridge: Cambridge University Press, 2014.

[18] *Recommender systems: The textbook*, 1st edition. New York, NY: Springer Science+Business Media, 2016.

[19] R. Banik, *Hands-On Recommendation Systems with Python*, First. Birmingham, UK: Packt Publishing Ltd., 2018.

[20] M. A. Teixeira, Ed., *Perfume engineering: Design, performance & classification*. Oxford, England ; Waltham, MA: Butterworth-Heinemann, 2013.

[21] "Eau de Cartier Cartier perfume - a fragrance for women and men 2001," *www.fragrantica.com*. [Online]. Available: https://www.fragrantica.com/perfume/Cartier/Eau-de-Cartier-309.html. [Accessed: 28-Oct-2018].

[22] "Reviews of Eau de Cartier by Cartier — Basenotes.Net." [Online]. Available: http://www.basenotes.net/fragrancereviews/fragrance/26120495. [Accessed: 28-Oct-2018].

[23] "Netflix Update: Try This at Home." [Online]. Available: https://sifter.org/~simon/journal/20061211.html. [Accessed: 10-Dec-2018].

[24] ACM Conference on Recommender Systems, SIGCHI (Group : U.S.), and Association for Computing Machinery, *RecSys'16: Proceedings of the 10th ACM Conference on Recommender Systems : September 15-19, 2016, Boston, MA, USA*. 2016.

[25] N. Hug, "Home," *Surprise*. [Online]. Available: http://surpriselib.com/. [Accessed: 16-Dec-2018].

[26] "Scikit-learn: Machine learning in Python — scikit-learn 0.20.1 documentation." [Online]. Available: https://scikit-learn.org/stable/index.html. [Accessed: 16-Dec-2018].

[27] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: an open architecture for collaborative filtering of netnews. Proceedings of the ACM Conference on Computer Supported Cooperative Work , pp. 175–186, 1994.

[28] G. Adomavicius, and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering , 17(6), pp. 734–749, 2005.

[29] A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker. Developing constraint-based recommenders. Recommender Systems Handbook , Springer, pp. 187–216, 2011.

[30] C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. Recommender Systems Handbook , pp. 107–144, 2011.

[31] M. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting Web sites. Machine learning , 27(3), pp. 313–331, 1997.