Fashionista

Abstract

In the following proposal, I introduce Fashionista, a personal stylist. It is capable of recommending new clothing and styles to users, based on their tastes, purchase history and demographic information, along with the collaborative opinion of other users. In addition, I briefly explain the data sources and implementation details with some preliminary results on our sample dataset.

1- Introduction and Motivation

With the huge selection of different brands and retail stores, together with the society's cultural norms and fashion trends, deciding what type of cloths to buy and wear has become a challenging concern for many users. Even if you know how to use online services of department stores (like Macy's, Nordstrom, Neiman Marcus and etc.), and if you know what styles suit you best, or what the newest trends are, and even if you know what suitable styles for different occasions are, you should spend hours and hours on your laptop or cell phone to find a fair piece of clothing with a reasonable price. Furthermore, even if you have shopped online, it may still require a couple of trips to the store for returning the items you didn't like on you. Based on Wall Street Journal, about a third of all Internet transactions are returned by shoppers! Clever tactics like Fashionista, can help both shoppers and retailers to save those extra cost and time spent. A personal stylist who can recommend new fashion trends and clothing styles based on your taste, body type and other demographic information, the one who can narrow your options to a best set of apparel, and can select the best set from thousands of styles, brands and retail stores, is a dream come true. Here, I am going to propose a framework to create that personal stylist using data science and machine learning techniques. Our automated free stylist has access to a huge collection of styles that are available online. It can find affordable and chic garment that suits you best based on your body type, color, age and taste. It takes into account your budget, and helps you save considerable amounts of time and money. Fashionista is able to identify contextual information from online catalogs and turn it into value, easily accessible to the users.

Our killer app, Fashionista, uses different sources of data. It uses catalogs of different department stores and brands along with user's demographic information, history of user's purchases, history and reviews of other users with similar demographic information and taste. Fashionista learns user's taste through time, updates its knowledge base constantly and builds a hybrid evolving recommender system. This way, the users would learn about new clothing items/fashion trends and receive recommendations based on their preferences. Furthermore, they are sure that they are being offered what society and other people with the same taste as themselves accept, and what is trending in the society.

Almost all large retail stores have APIs, which let other apps get the information of their catalogs in their inventory. Every item, in its catalog, has specific descriptions and meta-data as well as different images. With the data from major online stores and a large user base, a good data science and machine learning model can form our personal stylist and shopping recommender system. In the following, we explain the proposed model and all the data it requires to be built upon. Then, an overview of the implementation techniques will be provided.

2- Implementation

In order to implement Fashionista, we require to merge and make use of data from different sources. As already mentioned in Introduction Section, Fashionista not only recommends garments based on your taste and demographic information, it also uses other people's opinions. Besides it searches through the online inventory of all your favorite brands and department stores to find what you deserve. The three sources of information Fashionista uses are: 1) demographic information of the users including gender, age, skin color, body type and measurements provided by the users; 2) user preferences and taste through their purchase history, reviews, ratings and their social media (e.g., Pinterest, Facebook); and 3) online catalogs of department or single-brand stores. Fig. 0 shows an overall architecture of the proposed framework.

The online catalogs from various department stores and many single-brand stores are often available through their APIs. This is a huge set of data and would therefore need big data analysis methods. In the first version of Fashionista, we are considering only the meta-data and description of the items. We can later use images for our recommendation process as well.

Unfortunately, department stores do not provide access to their users' inventories, including their information and purchase histories. As a result, we need to gather these information from our users and build our knowledge-base through time. At initial phases, Fashionista relies on the demographic information of users as well as their preferences extracted from their social media, to build a content based recommendation. After gathering enough information about the users' purchase history and their reviews and ratings, the recommendation is going to be a hybrid approach of content based and collaborative filtering.

Currently, all the data for building such a model cannot be gathered or retrieved. As a result, we use an online publicly available datasets containing users' transactions and item descriptions to showcase the results and validate the hypothesis. In the main implementation, we would require to use data from the online retailers (retrieved through their APIs). We would also need to build a large user-base to be able to infer and learn from the users. Different incentives could be incorporated for the users to give correct information about themselves and their preferences, and to encourage them to review more items.

Although Fashionista is users' personal stylist, it has benefits for retailers as well. Many retailers accept referrals in exchange for a percentage of the whole transaction. Hence, this can be used as a clever tactic as a source of income for the service and incentives for the users to buy through our app, by offering cashbacks on purchases that the app recommends to them for free.

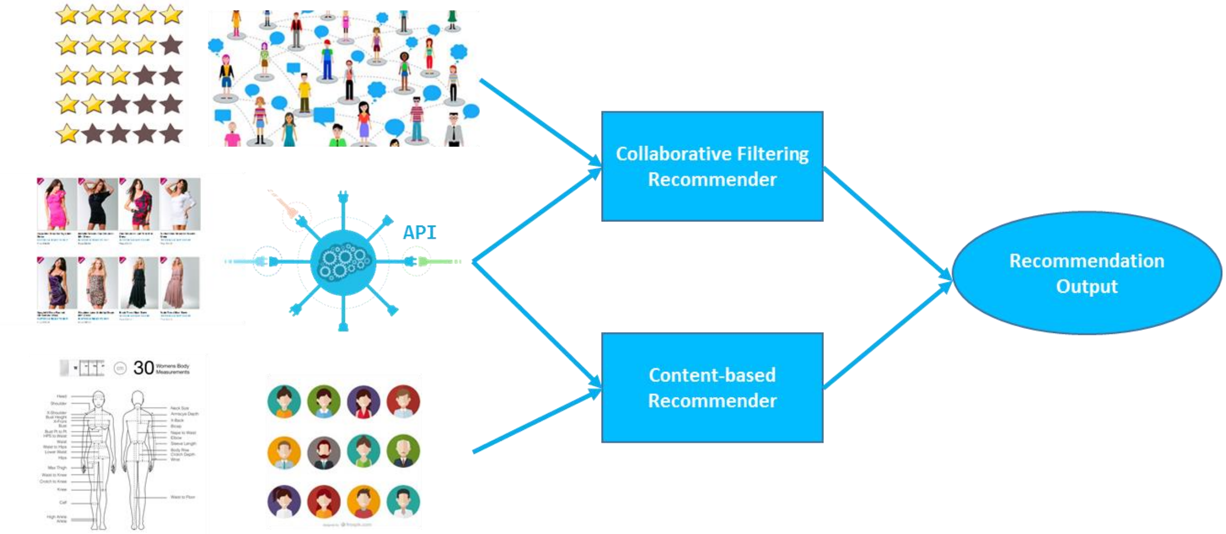

API
30 Womens Body Measurements
Collaborative Filtering Recommender
Content-based Recommender
Recommendation Output

Figure 0: The overall architecture of the proposed framework.

3- Current Implementation

Fashionista works based on a hybrid recommendation system. We use content-based filtering (for leveraging what item from which retailer is the best match to the user's taste) as well as collaborative filtering (for taking into account other users' experiences on items, along with the society's fashion trends) to find the best items for recommending to users. In the initial stages, our knowledge-base on users' ratings and tastes are limited, so the weight of the content-based filtering would be higher than the collaborative filtering, but through time the knowledge-base is getting large enough, and we can increase the weight for the collaborating filtering counterpart. In the content-based recommender, using items' descriptions and based on user profile (i.e., type of the items that user likes, the type of items suits their body type, and other demographic information), the algorithm recommends items that are the best match for the user. In the collaborative filtering subsystem, automatic predictions (filtering) about the interests of a user is made by collecting preference information from various other users (collaboration).

To show how Fashionista works, we showcase our proposal using a dataset containing user transaction data and items' descriptions. The data, Online Retail, is publicly available at UCI ML repository. The dataset contains all the transactions occurring between 01/12/2010 and 09/12/2011 for an online retail store. The dataset includes a short description for each item.

For the collaborative filtering subsystem, we implemented the approach proposed in[1]. The recommender is based on matrix factorization and alternative least square. We have shown that this recommender works well in our case of online retail data, where users have ranked what they have purchased. Receiver Operating Characteristic (ROC) curve for a sample test user is plotted in Fig. 1(a). The Area Under the Curves (AUCs) for all the test samples are depicted in Fig. 1(b). Without having any information about the description of items, the recommender system can effectively recommend items to users. The average

[1] Hu, Yifan, Yehuda Koren, and Chris Volinsky. "Collaborative filtering for implicit feedback datasets." IEEE International Conference on Data Mining, 2008.

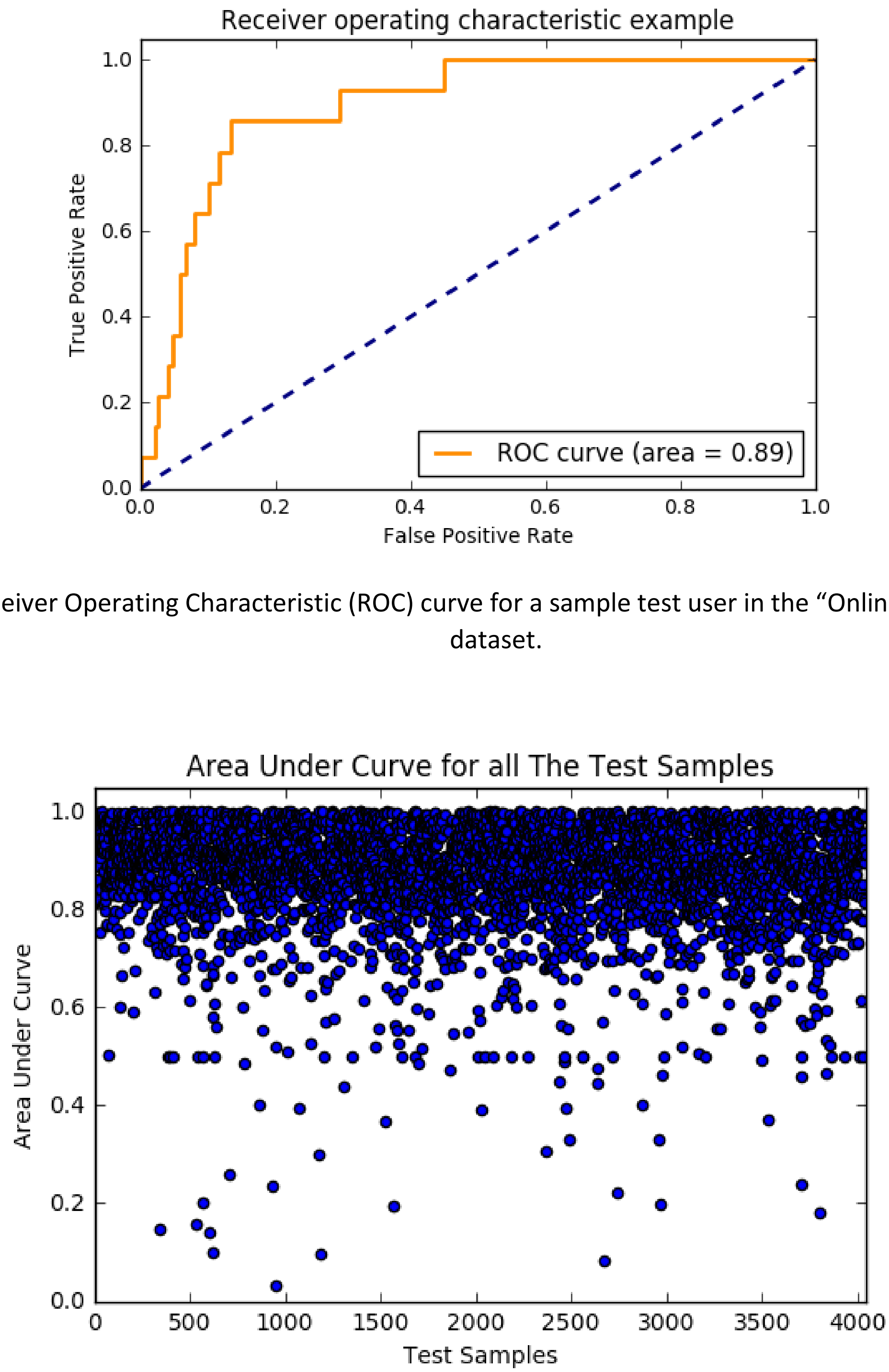
AUC for all the test samples is **0.87**, which indicates an overall efficient performance of the recommendation system. The content based filtering is based on the item descriptions and their similarities. Although in our sample dataset, we only have a short description (4-5 words) for each item, rather than full descriptions of the items available in the clothing catalogs, we can cluster the similar items. We used 1-gram Term Frequency–Inverse Document Frequency (TF-IDF)[2] to calculate the similarities of the items. We have shown the clustered similarity matrix in Fig. 2(a). To show a more detailed figure, we have plotted the similarity matrix for the first 1000 items in Fig. 2(b).

To implement the Fashionista, I am planning to use Apache Hadoop Ecosystem, since we are dealing with big data. We also need a database management system for our building the knowledge-base.

4- Conclusions

In this document, I have briefly proposed Fashionista, a personal stylist capable of gathering lots of information about users, styles, brands and catalogs to recommends garment to users based on their demographic information, taste, preferences and collaborative opinions of other users. It has the potential of becoming a killer app with a large user-base.

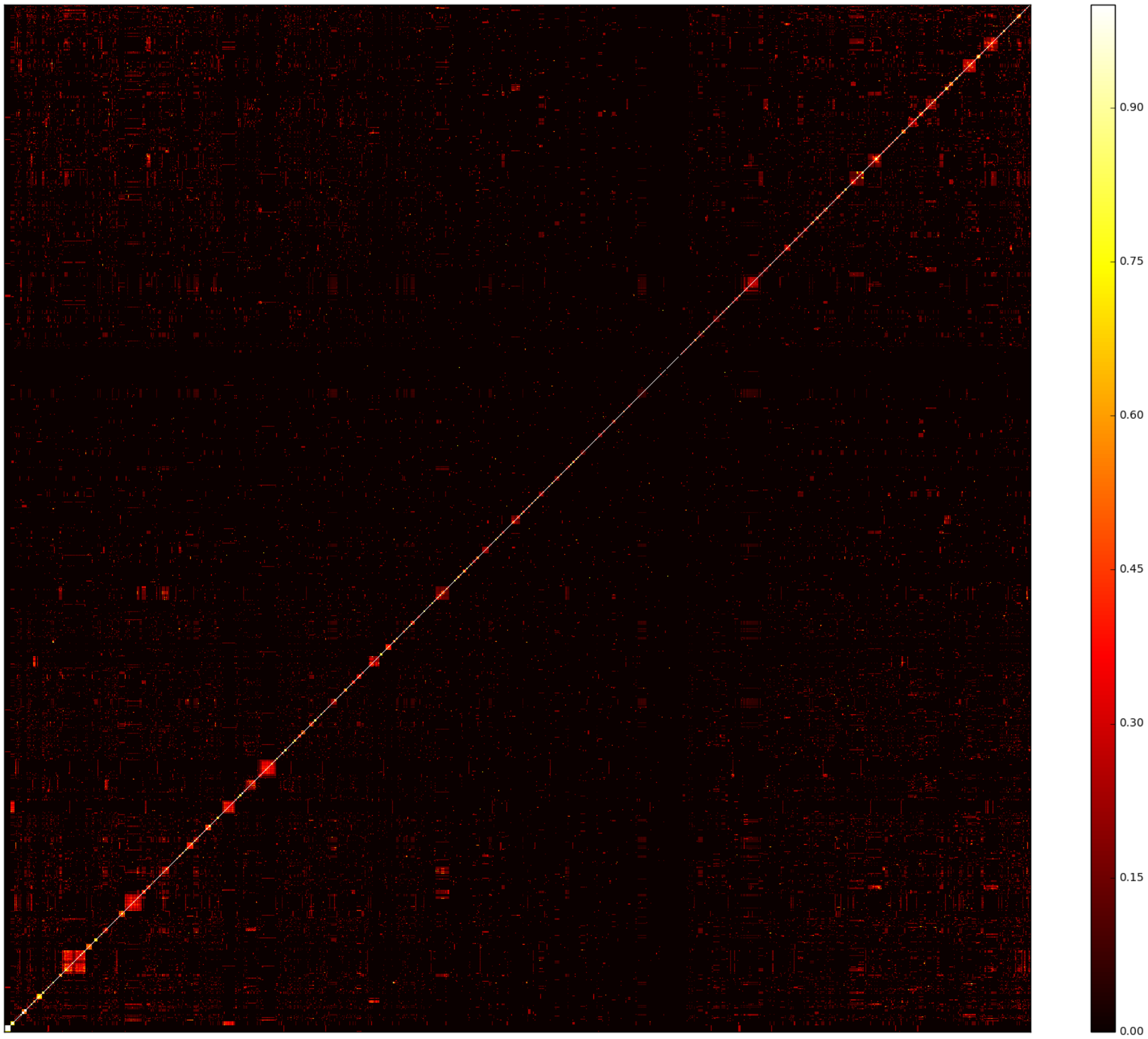
[2] Rajaraman, A.; Ullman, J. D. “Mining of Massive Datasets”, Data Mining, 2011, pp. 1–17.



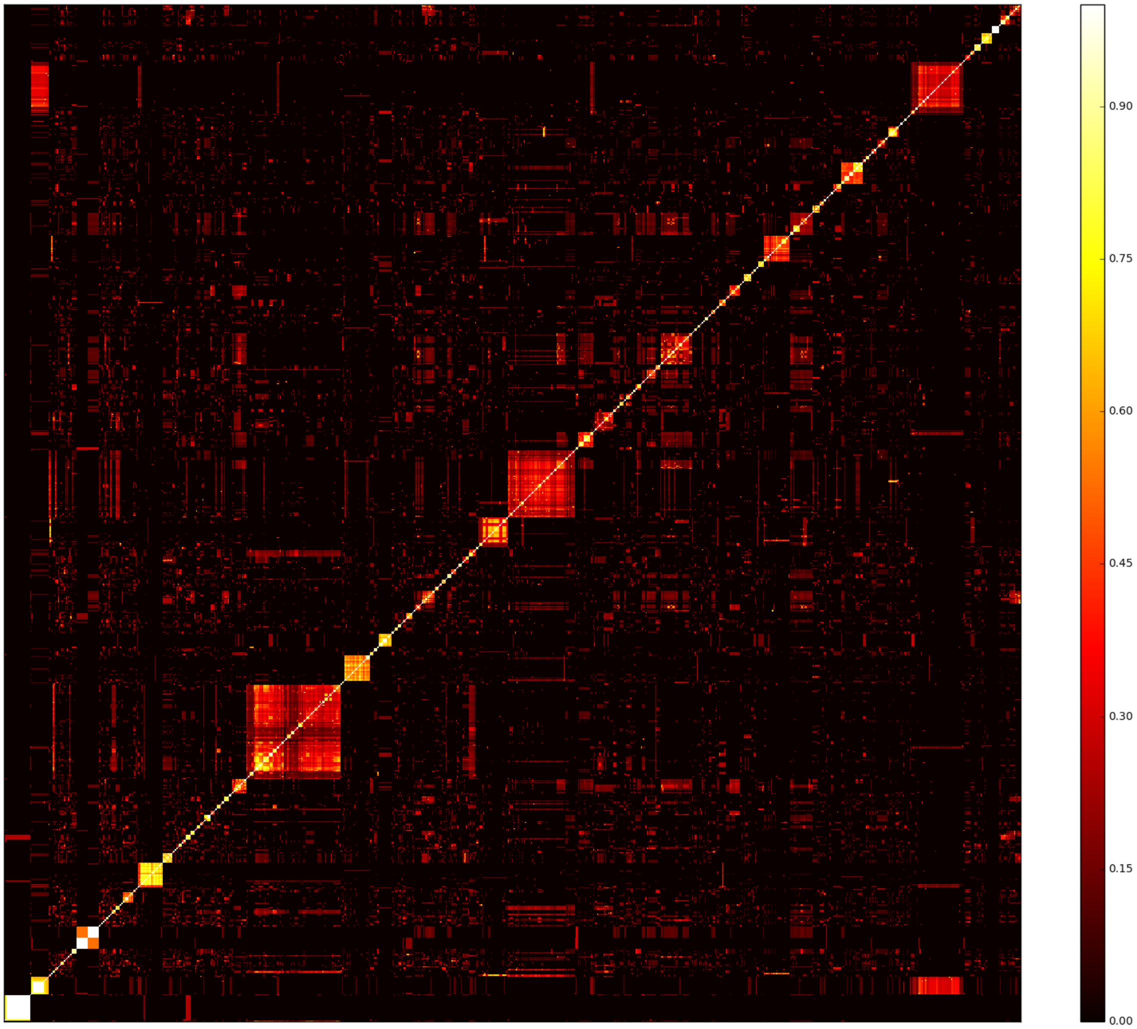
(a) Receiver Operating Characteristic (ROC) curve for a sample test user in the “Online Retail” UCI dataset.

(b) Area under ROC (AUC) for all test samples in the “Online Retail” UCI Dataset.

Figure 1. Results of the Collaborative Filtering Recommendation system on the “Online Retail” UCI dataset, with 80% of the transactions for training the model and the remaining 20% used for testing the system. (a) Shows the ROC for a sample user, and (b) plots the AUC values for all the test samples.



(a) Similarity matrix of the items in the “Online Retail” UCI dataset, using the n-gram method on the descriptions of the items. Based on this similarity, the items are clustered using a hierarchical clustering method. Then, similar items are put near each other to form small clusters.



(b) A larger view of the first 1000 items. As can be seen, the items have formed small clusters based on the similarities between their descriptions.

Figure 2. The similarity (using n-gram) and clustering (using a hierarchical clustering method) results on the "Online Retail" UCI dataset.