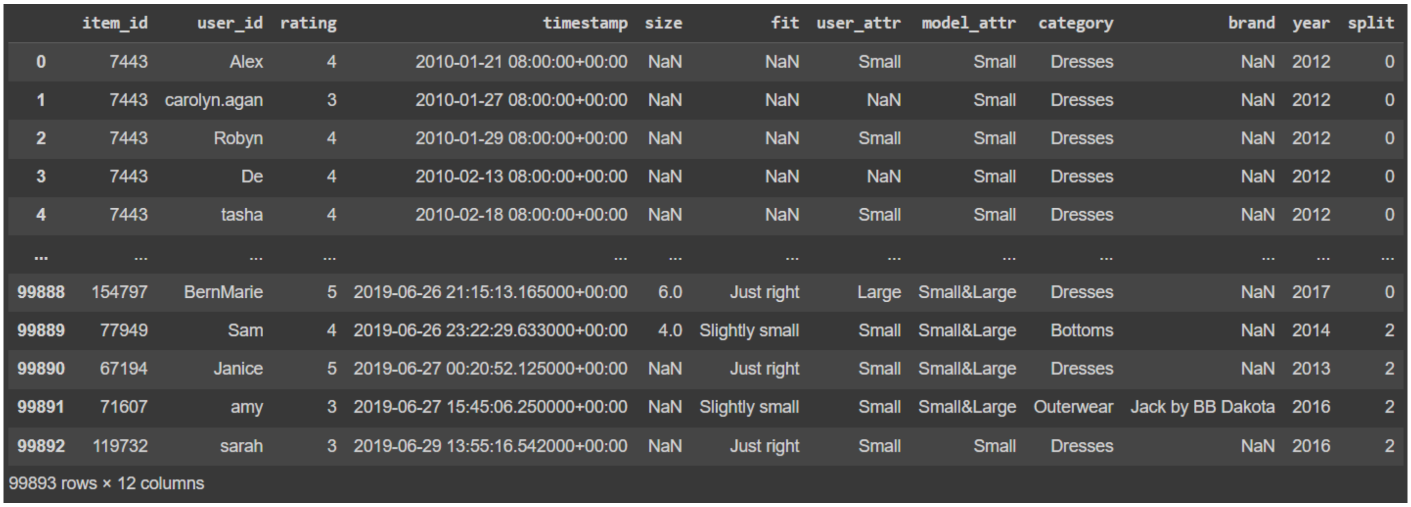
Assignment 2

First, let us see how our database looks.



	item_id	user_id	rating	timestamp	size	fit	user_attr	model_attr	category	brand	year	split
0	7443	Alex	4	2010-01-21 08:00:00+00:00	NaN	NaN	Small	Small	Dresses	NaN	2012	0
1	7443	carolyn.agan	3	2010-01-27 08:00:00+00:00	NaN	NaN	NaN	Small	Dresses	NaN	2012	0
2	7443	Robyn	4	2010-01-29 08:00:00+00:00	NaN	NaN	Small	Small	Dresses	NaN	2012	0
3	7443	De	4	2010-02-13 08:00:00+00:00	NaN	NaN	NaN	Small	Dresses	NaN	2012	0
4	7443	tasha	4	2010-02-18 08:00:00+00:00	NaN	NaN	Small	Small	Dresses	NaN	2012	0
...	...	...	...	...	...	...	...	...	...	...	...	...
99888	154797	BernMarie	5	2019-06-26 21:15:13.165000+00:00	6.0	Just right	Large	Small&Large	Dresses	NaN	2017	0
99889	77949	Sam	4	2019-06-26 23:22:29.633000+00:00	4.0	Slightly small	Small	Small&Large	Bottoms	NaN	2014	2
99890	67194	Janice	5	2019-06-27 00:20:52.125000+00:00	NaN	Just right	Small	Small&Large	Dresses	NaN	2013	2
99891	71607	amy	3	2019-06-27 15:45:06.250000+00:00	NaN	Slightly small	Small	Small&Large	Outerwear	Jack by BB Dakota	2016	2
99892	119732	sarah	3	2019-06-29 13:55:16.542000+00:00	NaN	Just right	Small	Small	Dresses	NaN	2016	2

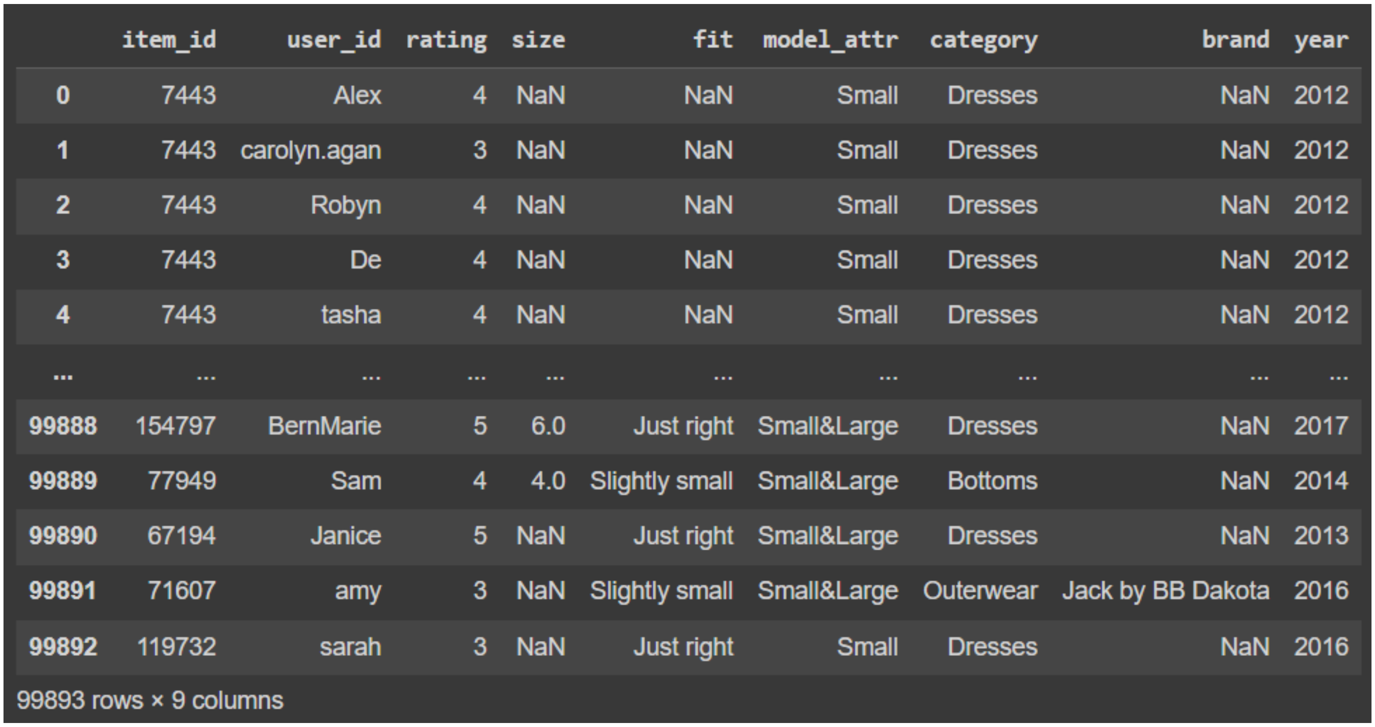
99893 rows × 12 columns

In this dataset, there are many columns.
Our goal for this assignment is to make a collaborative filtering-based recommendation system.
For that, we first need to analyze the dataset and find which information is valuable to us for achieving that.
From the columns given to us we have given the following meanings to these columns:

- **Item_id:** This is used for uniquely identifying an item.
- **User_id:** This is used for uniquely identifying a user.
- **Rating:** This is the rating a user gives to a particular item.
- **Timestamp:** This is the time at which the product has been reviewed. We will drop this column because it only stores timestamps for logging purposes and does not help us make a good recommendation system.
- **Size:** This is the size of the clothing item.
- **Fit:** This is the review of the person about how well the clothing item fits the person.
- **User_attr:** This is the size of the person that he/she actually wears. This column is not beneficial for us because it tells us about the user, which does not influence the product's popularity. Thus, we will drop this column.
- **Model_attr:** This is the size of the product that has been purchased and reviewed.
- **Category:** This is the category to which the clothing item belongs.
- **Brand:** This refers to the brand of the clothing item.
- **Year:** This is the year of manufacture of the product.
- **Split:** This identifies if the item went to the training or testing set during their analysis. We are not going to follow this split, and thus, we will drop this column.

So, we will drop three columns from our dataset.

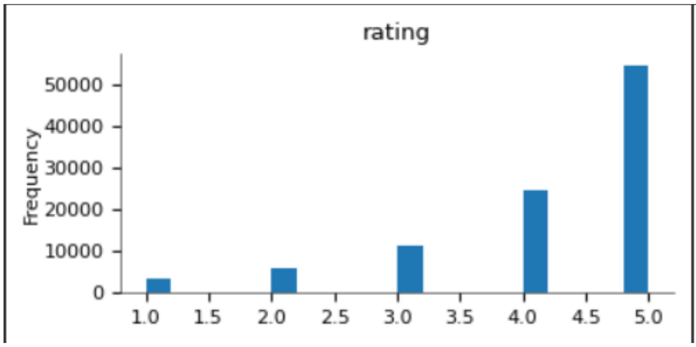
After dropping the columns, we get the following dataset.

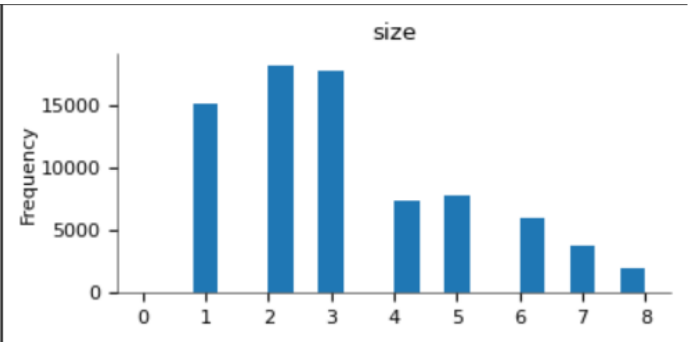


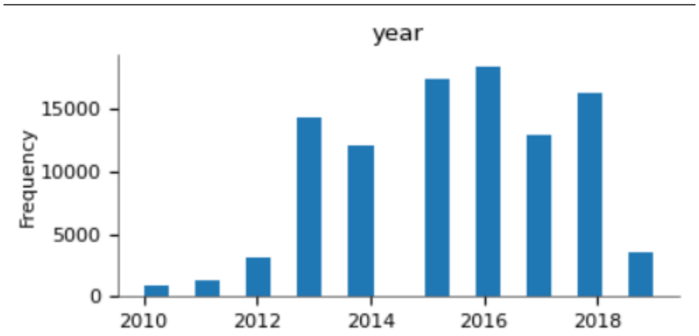
	item_id	user_id	rating	size	fit	model_attr	category	brand	year
0	7443	Alex	4	NaN	NaN	Small	Dresses	NaN	2012
1	7443	carolyn.agan	3	NaN	NaN	Small	Dresses	NaN	2012
2	7443	Robyn	4	NaN	NaN	Small	Dresses	NaN	2012
3	7443	De	4	NaN	NaN	Small	Dresses	NaN	2012
4	7443	tasha	4	NaN	NaN	Small	Dresses	NaN	2012
...	...	...	...	...	...	...	...	...	...
99888	154797	BernMarie	5	6.0	Just right	Small&Large	Dresses	NaN	2017
99889	77949	Sam	4	4.0	Slightly small	Small&Large	Bottoms	NaN	2014
99890	67194	Janice	5	NaN	Just right	Small&Large	Dresses	NaN	2013
99891	71607	amy	3	NaN	Slightly small	Small&Large	Outerwear	Jack by BB Dakota	2016
99892	119732	sarah	3	NaN	Just right	Small	Dresses	NaN	2016

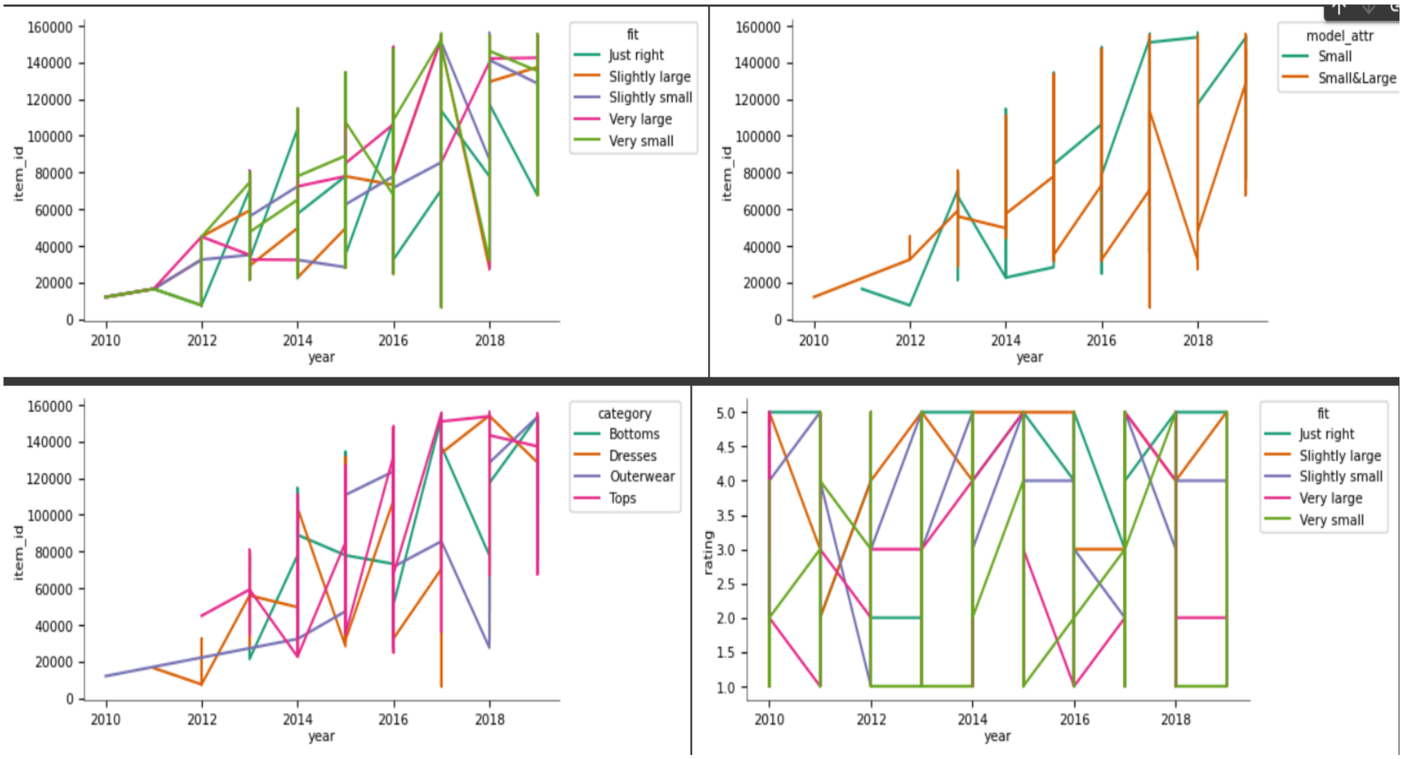
99893 rows × 9 columns

Now, let us see some visualizations of this dataset.









Now, let us discuss what we want to do in this assignment.
We want to make a recommendation system based on collaborative filtering. For this purpose, we will use item-item collaboration for recommendation rather than user-user collaboration. This is because the number of unique users(44784) is far greater than the number of unique items(1020) in our dataset. Another reason for choosing item-item-based collaborative filtering is that for most of the items, we have a frequency of their purchase, and thus, it is easy to make a system based on those purchases, and it will make a better decision, too.

Now, we will see how we have implemented this.

- First, we have given numerical values to the attribute "**fit.**" This helps us give meaning to the column, which helps us better understand the trend and dataset and make a better recommendation. We have done this in the following way:
  For **very small,** we have given a value of -2.
  For **slightly small**, we have given a value of -1.
  For **slightly large,** we have given value 1.
  For **very large,** we have given value 2.
- For the attribute **size,** we have assigned the global average value in places where the value is missing or NULL.
- For **fit**, we assumed it was a fit and gave it the value 0.
- We are encoding the columns "**brand**" and "**category**" and will use them for finding cosine similarity.

- We indirectly used **Pearson Coefficient Similarity** by calculating the normalized cosine similarity.
- If there was an item in the validation dataset, not the training dataset, we took the **global average**. (There were only four such items.)
- We took all those items rated by user x whose **normalized cosine similarity (**or Pearson similarity) to the item whose rating (which we needed to predict) was greater than 0.

In this assignment, we used a **feature vector** to calculate the similarity. A feature vector is a vector that is constructed by taking all the columns and combining them into one. For the categorical columns, we have performed encoding to convert them to numerical attributes.
We then split the dataset into training and testing. For the training dataset, we are supposed to perform k-fold cross-validation. So, we will split the training dataset into a 4:1 ratio (training: validation).
For the training dataset, we first find out the normalized cosine similarity, which in other words is called the Pearson similarity coefficient. Using this similarity measure, we have used the validation set to predict the ratings. Over four iterations (since we are using 4:1 split), we found the following values for RMSE:

- 1.1640743468815817
- 1.1650372750747184
- 1.1671211359837417
- 1.15261748827398
- 1.1715553180693117

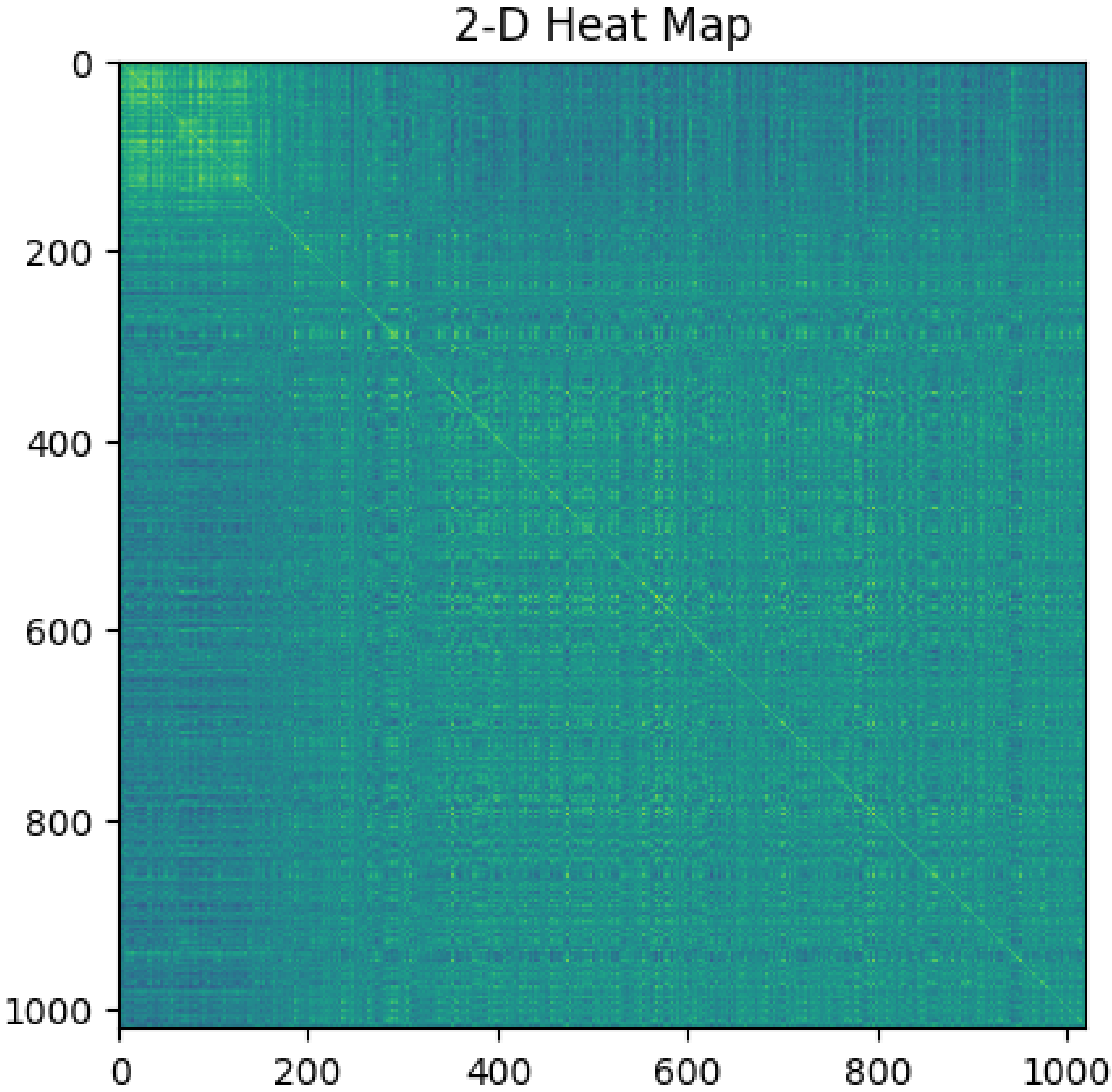
Then, we performed the testing on the test dataset. The value for Test RMSE is 1.1619350847083567.

Design Choices

In this assignment, we are using cosine similarity because:

- Cosine similarity works well with sparse data, effectively handling missing values and providing meaningful similarity scores even when data is incomplete.
- In recommendation systems, the user-item interaction matrix can be high-dimensional, especially when dealing with a large number of products and users.
- Cosine similarity produces similarity scores ranging from -1 to 1, where higher values indicate greater similarity.

We have also prepared the heatmap for the item_similarity matrix that we formed during the process, which can be seen here


2-D Heat Map